JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 2, pp. 485-494, Feb. 2024

Check for updates

슈퍼토큰 클러스터링 기반 ViT 합성곱 임베딩 모듈 경량화

이 진[1] · 이 주 환[2] · 김 진 영[3*]
[1]전남대학교 ICT융합시스템공학과 석사과정 [2]전남대학교 ICT융합시스템공학과 박사과정
[3*]전남대학교 ICT융합시스템공학과 교수

Lightweight ViT Convolutional Embedding Module Based on Super-Token Clustering

Jin Lee[1] · Ju-Hwan Lee[2] · Jin-Young Kim[3*]

[1]Master's Course, Department of ICT Convergence System, Chonnam National University, Gwangju 61186, Korea
[2]Ph.D's Course, Department of ICT Convergence System, Chonnam National University Gwangju 61186, Korea
[3*]Professor, Department of ICT Convergence System, Chonnam National University, Gwangju 61186, Korea

[요 약]

Transformer의 자연어처리 분야의 성공 이후 컴퓨터 비전 분야에서 ViT(Vision Transformer)가 등장하였다. ViT는 Transformer와는 다르게 이미지의 특성을 고려하지 않게 이미지를 임베딩 한다. 즉 ViT의 패치 임베딩 모듈은 픽셀간의 지역적인 관계성을 고려하지 않는다. 이를 개선하기 위해 EarlyConv에서 합성곱 임베딩이 제안되었다. 하지만 패치 임베딩에 비해 3배의 학습 파라미터를 사용하여 ViT의 과도한 학습 파라미터를 사용하는 문제를 악화시켰다. 이에 본 논문은 합성곱 임베딩 경량화를 목적으로 ViT의 연산량 감소 기법 중 하나인 슈퍼토큰 클러스터링을 결합한 슈퍼토큰 합성곱 임베딩 모듈을 제안한다. 이는 합성곱 임베딩 레이어를 사용하고 슈퍼토큰 클러스터링 알고리즘을 통해 연관성 있는 토큰을 그룹화하여 합성곱 임베딩에 비해 10배 적은 학습 파라미터로 성능을 향상한다. ImageNet100 이미지 분류 실험을 통해 제안한 방법을 검증한바, 기존 ViT 패치 임베딩 모듈보다 5.08%, 합성곱 임베딩 모듈보다 2.38% 높은 이미지 분류성능을 보인다.

[Abstract]

After the success of Transformer success in natural language programming (NLP), Vision Transformer (ViT) has emerged in the field of computer vision. Unlike NLP Transformers, ViT does not consider image characteristics during embedding. Consequently, the ViT's patch embedding module neglects spatial relationships between pixels. To overcome this, EarlyConv introduced convolutional embeddings, but it worsened the problem of ViT's excessive parameter usage—using three times more parameters than patch embedding. This paper introduces super-token convolutional embedding combined with super-token clustering to reduce ViT's computational complexity, aiming for lightweight convolutional embedding. By utilizing convolutional embedding layers and grouping tokens through super-token clustering, the proposed method enhances performance using 10 times fewer learning parameters than convolutional embedding and ViT's patch embedding, achieving 2.38% and 5.08% higher performance, respectively, in ImageNet100 classification.

색인어 : 인공지능, 컴퓨터 비전, 비전 트랜스포머, 합성곱 임베딩, 토큰 클러스터링
Keyword : Artificial Intelligence, Computer Vision, Vision Transformer, Convolution Embedding, Token Clustering

http://dx.doi.org/10.9728/dcs.2024.25.2.485

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 21 December 2023; **Revised** 16 January 2024
Accepted 23 January 2024

***Corresponding Author; Jin-Young Kim**

Tel: +82-62-530-1757
E-mail: beyondi@jnu.ac.kr

Ⅰ. 서 론

어텐션(Attention)에 기반한 트랜스포머(transformer)[1] 모델은 자연어처리 분야에서 큰 성과를 보였다. 따라서 트랜스포머 모델을 컴퓨터 비전 분야에 적용하였는데, 그 시초가 비전 트랜스포머 ViT(Vision Transformer)이다[2]. ViT의 성능 향상을 위하여 다양한 변형된 트랜스포머들이 제안되었는데, Swin Transformer[3], MobileViT[4], Deformable Attention Transformer[5]등이 있다. 트랜스포머는 이렇듯 컴퓨터 비전 분야에서 합성곱망(Covolutional Neural Network)에 비하여 우수한 성능을 보이지만 이미지를 임베딩하는 과정에서 문제점이 존재한다.

자연어처리의 Transformer는 데이터를 시퀀스로 만들고, 딥러닝 모델의 입력으로 사용하기 위한 토큰화와 단어 임베딩의 전처리 과정이 필요하다. 반면 컴퓨터 비전의 ViT는 이 두 가지를 통합한 패치 임베딩을 사용하는데, 이는 자연어 임베딩에 비하여 매우 간소하다. 패치 임베딩은 데이터 증강 이후 이미지를 일정한 크기로 분리하여 토큰화하고, 한 개의 전역 연결 레이어를 통해 토큰을 임베딩하는 과정을 말한다.

하지만 ViT의 패치 임베딩 방식은 본래 이미지 데이터가 갖는 의미론적인 특성에 부적합하다. 이미지는 객체, 배경 요소가 연속적으로 이루어져 있고, 이러한 요소들은 최소 단위의 집합인 픽셀로 구성된다. 이미지에 존재하는 객체는 일정 영역의 픽셀들이 유의미한 관계성을 가지고 모인 집합으로 볼 수 있다. 하지만 ViT는 이미지의 픽셀 간 지역적 관계를 고려하지 않고 패치 임베딩을 수행한다.

이러한 점에 착안하여 합성곱 임베딩이 제안되었다. 합성곱 임베딩이란 합성곱 신경망이 이미지에서 지역적인 픽셀의 연관성을 학습하는 방법을 차용하여 ViT에 적용한 임베딩 방법이다. EarlyConv[6]에 따르면 합성곱 임베딩 방법의 적용으로 기존 ViT에 비해 ImageNet-1k 데이터셋[7] 이미지 분류 정확도가 약 2% 향상된다. 하지만 합성곱 임베딩 방법은 기존 ViT 임베딩 모듈에 비해 약 3배의 학습 파라미터를 필요로 하는 큰 단점이 존재한다.

이 문제의 개선을 목적으로 본 연구는 합성곱 임베딩 모듈을 경량화하고자 하였다. 합성곱 임베딩은 이미지를 합성곱 연산으로 이미지를 저차원의 특징맵으로 만든다. 이 과정에서 많은 수의 채널을 사용하여 학습 파라미터를 과도하게 사용한다. 합성곱의 채널 수를 감소시키고, 학습 파라미터를 적게 사용하는 다운샘플링 방식을 사용하면 모듈의 경량화 목적을 달성할 수 있다. 하지만 과도한 학습 파라미터의 감량은 자칫 성능의 감소로 이어질 수 있어 합리적으로 경량화하려는 접근이 필요하다. 이러한 점을 고려하여 학습 파라미터를 적게 사용하는 다운샘플링 방법으로 토큰 샘플링을 주목하였다.

토큰 샘플링 연구는 ViT의 Attention 연산의 연산복잡도를 낮추기 위한 목적으로 등장하였다. ViT의 핵심적인 역할을 하는 Attention 연산은 입력 시퀀스 n, 즉 패치 임베딩 된 토큰에 $O(n^2)$이다. 이에 따른 연산 복잡도 증가는 대용량 데이터셋 학습을 요구하는 ViT를 학습하는데 긴 학습 시간이 불가피해진다. 또한 고화질의 이미지를 처리하기 어려운 문제가 발생한다.

이러한 연산복잡도를 낮추기 위한 방법 중, 토큰 샘플링 방법이 최근에 주목받았다. 토큰 샘플링은 학습 파라미터를 사용하지 않으면서 중요한 토큰을 선택하는 샘플링 알고리즘에 대한 연구이다. 이는 유의미한 토큰만을 선택하여 Attention 연산하기 때문에 기존 대비 파라미터를 대폭 낮출 수 있는 장점이 있다.

토큰 샘플링 연구는 일부 토큰을 제거하는 '토큰 제거'와 토큰을 그룹화하는 '토큰 그룹화' 방법으로 나뉜다. '토큰 제거'는 클래스 토큰을 기반으로 유의미한 토큰만을 활용하기 위하여 Top-K 알고리즘[8], 이분 매칭 알고리즘(Bipartite Algorithm)[9]을 접목한다. 반면 '토큰 그룹화'는 토큰 간의 유사도 및 거리를 연산하기 위하여 클러스터링 알고리즘을 접목한다. 토큰 샘플링 비교 연구[10]에서의 성능 비교를 통해 토큰 그룹화 방법이 낮은 성능을 보였다. 이후 유사한 픽셀을 지역적으로 클러스터링하는 슈퍼픽셀 알고리즘을 기반으로 SViT[11]는 토큰을 클러스터링하는 슈퍼토큰 클러스터링 알고리즘이 제안하였다. 이러한 슈퍼토큰 클러스터링 기반의 토큰 샘플링은 토큰 제거 방법에 비해 우수한 이미지 분류 정확도를 보여주었다.

토큰 클러스터링 알고리즘은 학습 파라미터를 사용하지 않으면서 중요한 토큰을 선택하는 다운샘플링 연산이다. 이러한 점에서 토큰 클러스터링은 합성곱을 대신하여 학습 파라미터 사용 없이 합리적으로 다운샘플링이 가능하다. 이에 따라 경량화된 합성곱 임베딩 모듈을 제안한다. 제안하는 방법은 합성곱 임베딩 과정 높은 성능을 보인 슈퍼토큰 클러스터링을 활용하여 합성곱 레이어의 일부를 대신하여 다운샘플링하는 방법이다. 이는 입력 이미지를 적은 수의 채널을 사용하는 합성곱 연산하여 특징맵을 얻고, 슈퍼토큰 클러스터링으로 다운샘플링하는 과정으로 이루어진다. 제안하는 임베딩 모듈은 ViT에서 이미지의 지역적인 관계성을 잘 학습하기 위한 합성곱 임베딩 모듈의 경량화와 함께 높은 성능을 유지한다.

본 논문에서 기여하는 연구 내용은 다음과 같다.

1) 슈퍼토큰 클러스터링을 합성곱 임베딩에 통합한 경량 임베딩 모듈을 제안하고, ViT의 패치 임베딩 모듈에 비해 학습 파라미터를 10배 감소시켰다.
2) 벤치마크 데이터셋 ImageNet100[12]의 이미지 분류에 대하여, 기존 ViT 패치 임베딩 모듈 대비 5.08%, 합성곱 임베딩 모듈 대비 2.38% 성능 향상을 하였다.

논문의 전체 구성은 다음과 같다. 2장에서는 본 연구와 관련된 연구를 기술하고, 3장에서는 제안하는 토큰 클러스터링 경량 임베딩 모듈을 논한다. 4장에서는 임베딩 모듈의 실험 결과 및 분석에 관하여 기술하고 5장에서는 결론을 기술한다.

II. 관련 연구

2-1 Vision Transformer의 합성곱 임베딩

ViT 등장 이전은 합성곱 신경망(Convolutional Neural Network)이 컴퓨터비전 분야에서 지배적이었다. 합성곱 신경망은 가중합 연산을 하는 윈도우를 슬라이딩 하는 방식으로 학습한다. 이러한 슬라이딩 윈도우 내 가중합 연산을 통해 픽셀간의 지역적인 관계성을 모델링 할 수 있다.

반면 ViT는 Attention 기반 이미지의 전체 맥락을 학습한다. 이는 패치 임베딩의 과정에서 입력 이미지를 시퀀스로 변환한다. 임베딩의 과정에서 2D 이미지는 1D 벡터로 변환되는데, 이는 원본 이미지가 가지고 있는 픽셀 간의 관계성을 유지하기 어렵게 만든다. 이러한 이유로 합성곱 신경망에 비해 낮은 지역성을 가지기 때문에 귀납적 편향이 약하게 나타나는 특성을 갖는다. 따라서 사전학습 없이 합성곱 신경망과 동일한 성능을 나타내기 위해서는 많은 학습 데이터와 강한 데이터 증강, 긴 학습시간을 요구한다.

이러한 문제를 개선하고자 하는 목적으로 ViT에 합성곱 신경망이 가지는 귀납적 편향을 주입하려는 목적의 다양한 연구가 진행되었다. 이러한 연구는 합성곱 신경망의 계층적인 구조와 합성곱 레이어를 주로 활용한다. 이에 관련된 주요 연구는 기존의 ViT보다 귀납적 편향이 강화된 CVT[13] 모델이 대표적이다. CvT는 ViT에 합성곱 레이어의 추가를 통해 지역적인 정보를 효과적으로 추출한다. 이는 합성곱 신경망과 유사하게 모델 아키텍처를 3가지 단계로 구분한다. 각 단계는 합성곱 임베딩으로 지역적인 정보를 추출하고, 이후 Attention 연산을 통해 전역적인 정보를 추출한다. 이러한 합성곱 기반 토큰 임베딩의 등장 이후 Swin Transformer, Focal Transformer[14], CMT[15] 등의 ViT 기반 모델은 계층적인 아키텍처에 이를 활용한다.

이와 다르게 EarlyConv는 계층적인 아키텍처를 사용하지 않고, 기존의 패치 임베딩 모듈만을 합성곱 임베딩으로 변경하였다. 해당 모듈은 ViT에서 ImageNet-1k 데이터셋 이미지 분류 성능을 기존 ViT 대비 약 1~2% 향상시켰다.

제안된 합성곱 임베딩 방법들은 ViT모델에 귀납적 편향을 강하게 주입하여 이미지의 의미론적인 문맥과 관계성을 학습도록 유도하지만, 기존 ViT보다 더 많은 학습 파라미터를 요구한다. 이는 고차원의 원본 이미지를 많은 채널수를 가진 합성곱으로 연산하기 위해서는 필수 불가결한 문제이다.

2-2 Vision Transformer 토큰 샘플링

ViT에서 사용하는 Attention 연산의 계산 복잡도는 입력 시퀀스 n에 대하여 $O(n^2)$ 이다. 이때 ViT의 입력 시퀀스는 패치 임베딩 된 토큰이다. 이는 입력 이미지의 크기가 커질수록 연산량이 제곱으로 증가함을 의미한다. 이러한 계산 복잡도는 대용량 데이터셋 학습 시 많은 양의 학습 시간을 요구하며, 고화질의 이미지를 처리하기 어려운 문제를 만든다.

이에 따른 계산 복잡도를 줄이고자 하는 연구로 주목받는 방법은 토큰 샘플링 연구이다. 토큰 샘플링은 유의미한 토큰만을 선택하는 방법으로, Attention 연산의 계산 복잡도를 줄이는데 효과적이다. 일반적으로 이미지 데이터는 연속적이며 중복성이 많은 특징을 보인다[16]. 이것은 이미지 내의 객체 및 배경과 같은 추상적 요소를 판별 시 모든 픽셀이 필요한 것이 아님을 의미한다. 이에 따라 유의미한 토큰만을 선택하는 것은 이미지의 특성에 맞춘 합리적인 모델 학습 방법이라고 할 수 있다. 토큰 샘플링의 대표적인 연구[8],[9],[16]는 유사성이 낮은 일부 토큰만을 사용하는데 이는 기존 Attention 연산 속도를 빠르게 향상 시킨다.

토큰 샘플링은 토큰 선택 방법에 따라 토큰 제거 방법과 토큰 그룹화 방법으로 분류한다. 연산량 감소 측면에서 초기 연구로 토큰 제거 방법이 가장 먼저 제안되었다. 클래스 토큰을 기준으로 유의미한 토큰만을 선택하는 EViT[8]와 경량화 네트워크를 통해 토큰을 선택하는 AdaViT[16], 이분 매칭 알고리즘[9]을 토큰 제거에 도입하는게 대표적인 방법들이다.

반면, 토큰 그룹화 방법은 토큰을 제거하지 않고 의미론적인 토큰끼리 그룹화하는 방법을 사용한다. K-Medoids 알고리즘을 기반으로 토큰 그룹화하는 방법, DPC-KNN 알고리즘 기반의 방법이 대표적이다. 이는 이미지의 중요한 맥락이나 의미론적인 요소를 잃어버릴 위험 줄일 수 있으나, 토큰 제거 방법의 성능을 능가하지 못한다.

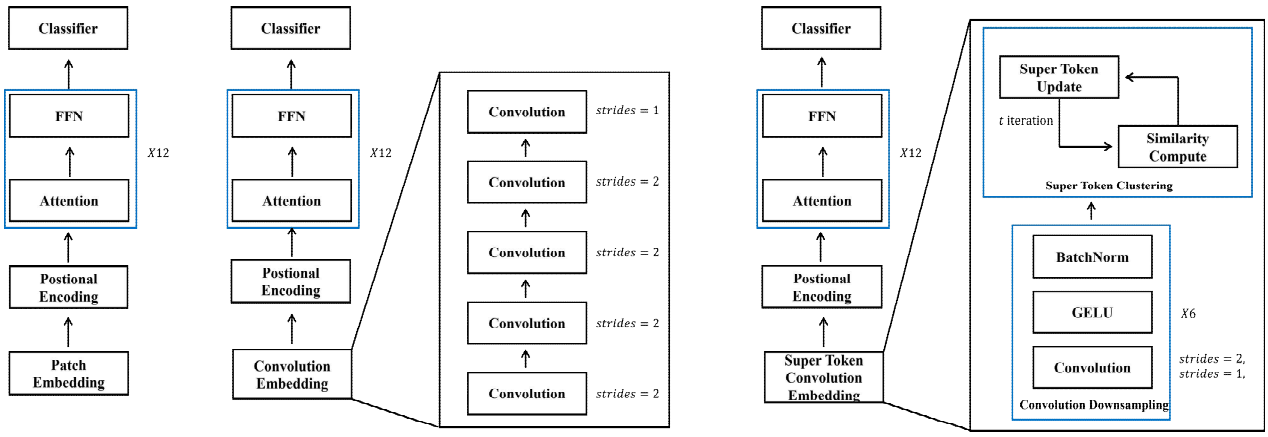
이후 슈퍼픽셀 알고리즘 기반의 토큰 그룹화 방법이 주목받는다. 일반적으로 슈퍼픽셀 알고리즘은 이미지에서 유사한 특성을 가진 작은 영역(슈퍼픽셀)으로 분할하는 최적의 방법으로 알려져 있다. 이는 색상, 강도, 텍스처 등과 같은 특성을 기반으로 결정한다.

토큰 샘플링 연구는 유사한 픽셀 그룹화의 장점을 토큰에 적용하였다. 즉 토큰을 그룹화하는 슈퍼토큰 클러스터링을 사용한다. 관련 연구인 계층적인 이미지 분할에 슈퍼픽셀의 개념을 활용한 HCFormer[17]가 있다. 그리고 기존의 클러스터링 알고리즘과 다르게 슈퍼픽셀 알고리즘을 사용한 SViT[11]가 대표적이다. SViT는 토큰을 Attention 연산 직전에만 클러스터링하고 Attention 이후 이를 원본 토큰을 슈퍼픽셀로 재구성하는 방식이다. 이는 2,500백만 개(25M)의 학습 파라미터로 ImageNet-1k 이미지 분류 정확도 83.6%의 높은 성능을 보였다.

III. 슈퍼토큰 클러스터링 경량 합성곱 임베딩 모듈

3-1 ViT 패치 임베딩 및 합성곱 임베딩

기존 ViT의 패치 임베딩은 x_{img}를 입력 텐서를 재배열하고 전역 연결 레이어 1개를 사용하여 임베딩 차원을 맞춘다.



(a) ViT (b) 합성곱 임베딩 ViT (c) 슈퍼토큰 합성곱 임베딩 ViT

그림 1. 임베딩 모듈 ViT 구조 비교
Fig. 1. Vit embedding module structure comparsion

입력 텐서의 재배열은 2D의 이미지를 1D의 시퀀스로 변경하는데, 픽셀간 위치관계를 포함한 지역적 특징이 손실된다.

EarlyConv의 ViT 합성곱 임베딩 방법은 가로 H, 세로 W, 채널 C의 x_{img}를 패치 크기 P에 맞추어 합성곱 레이어로 $x_{embed} \in R^{\frac{H}{P} \times \frac{W}{P} \times C}$로 다운샘플링하고 텐서 재배열을 통해 임베딩을 완성한다. 합성곱 임베딩의 합성곱 연산은 2D 커널 크기 안의 픽셀들을 가중합하여 하나의 값으로 만든다. 이러한 연산 방식은 이미지를 다운샘플링하면서 원본 이미지가 가지고 있는 픽셀간의 지역적인 정보가 유지되게 한다. 하지만 합성곱 임베딩은 합성곱을 사용하여 추가적인 학습 파라미터를 사용한다. 학습 파라미터를 비교하였을 때, ViT의 패치 임베딩 모듈에 비해 3배 정도 많은 파라미터를 사용한다.

3-2 슈퍼토큰 합성곱 임베딩 모듈

슈퍼픽셀 알고리즘은 픽셀을 그룹화하는 알고리즘으로 그룹화를 위해 그래프 기반 방법과 클러스터링 방법으로 나뉜다. 이러한 슈퍼픽셀의 개념에 착안하여 SViT는 슈퍼픽셀 알고리즘을 기반으로 하는 토큰 클러스터링을 제안하여 Attention 연산량을 줄이고자 하였다. 이러한 목표에 맞추어 SViT는 네트워크 내의 모든 Attention 연산 전에 토큰을 슈퍼픽셀 클러스터링한다. Attention 연산 토큰의 개수는 클러스터링 된 토큰의 개수로 감소한다. 어텐션 연산 이후 SViT는 원래 토큰의 개수만큼 복원하고 전체 토큰을 순방향 신경망으로 전달한다. 본 논문은 토큰 클러스터링을 SViT와 다른 목적인 합성곱 임베딩의 경량화 목적으로 사용하고자 한다. SViT와 다르게 토큰의 복원 과정은 사용하지 않고, 클러스터링만 하면 학습 파라미터를 사용하지 않으면서 유사한 토큰간의 그룹화를 통해 입력 데이터가 다운샘플링 된다. 이러한 특성을 활용하여 합성곱 레이어의 일부를 슈퍼토큰 클러스터링으로 대체하면 임베딩 모듈의 학습 파라미터는 감소하지만, 우수한 다운샘플링 성능으로 오히려 성능이 개선된다.

제안하는 경량 슈퍼토큰 합성곱 임베딩 모듈의 구조는 그림 1의 (c)와 같다. 네트워크의 임베딩 모듈을 제외한 나머지 부분은 (a)와 같은 ViT의 구조를 그대로 사용하였다. 합성곱 임베딩 모듈은 그림(b)와 같이 이미지를 다운샘플링하는 모든 과정을 학습 파라미터를 사용하는 합성곱 레이어를 사용하였다. 이와 달리 제안하는 경량 임베딩 모듈에서는 다운샘플링 과정의 일부를 학습 파라미터를 사용하지 않는 슈퍼 토큰 클러스터링을 통하여 연산한다.

본 논문에서 제안하는 슈퍼 토큰 클러스터링 경량 임베딩 모듈은 크게 두 가지 연산 과정을 통해 최종적으로 입력 이미지를 임베딩한다. 먼저 입력 이미지를 합성곱 레이어로 다운샘플링한다. 그리고 토큰 클러스터링 과정에서 합성곱 다운샘플링이 완료된 결과의 토큰을 내적합으로 유사도를 계산한다. 이후 이미지의 특정 토큰을 대표하는 클러스터링 토큰에 입력한다. 클러스터링 과정은 시각적으로 그림2로 표현하였다. 두 가지의 연산 과정에 대한 구체적인 과정은 다음과 같다.

1) 합성곱 다운샘플링

임베딩 모듈은 가로 H, 세로 W, 3채널의 데이터 증강이 적용된 이미지 $x_{in} \in R^{(H \times W \times 3)}$를 입력으로 사용한다. 이를 합성곱 레이어를 통해 다운샘플링한다. 이는 표 5와 같이, 합성곱 필터의 stride를 2와 1을 교차하여 사용하고 합성곱을 총 6회 반복한다. stride 1을 기반한 합성곱 레이어를 교차한 이유는 이미지의 지역적인 정보 추출하기 위함이다. 수식 1에 따르면, 합성곱 다운샘플링의 출력값 X_{out}이 출력된다.

$$X_{out} = BatchNorm(GELU(Conv(x_{in}))) \quad (1)$$

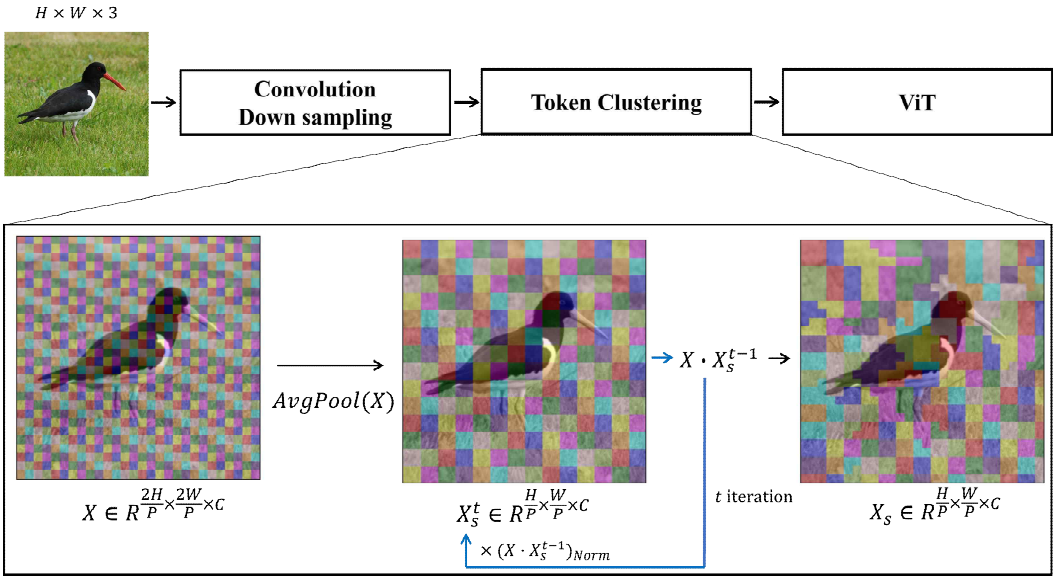


그림 2. 슈퍼토큰 클러스터링 과정
Fig. 2. Superpixel token clustering process

이후 입력 이미지 x_{in}에 대해 합성곱 레이어를 통과하고, GELU[18] 활성화 함수를 사용한다. 이를 통해 비선형 특징을 추출하고 배치정규화(BatchNormalization)[19]하여 X_{out}가 얻어진다. 반복된 합성곱 레이어로부터 얻어진 X_{out}은 슈퍼토큰 클러스터링을 위한 입력으로 사용된다.

2) 슈퍼토큰 클러스터링 유사도 연산

합성곱 다운샘플링의 출력값 X_{out}을 기반으로 슈퍼토큰 클러스터링을 수행한다. 슈퍼토큰 클러스터링 모듈은 합성곱 다운샘플링 된 가로 $\hat{H}$, 세로, $\hat{W}$, 채널 수가 C인 특징맵 $X_{out} \in R^{(\hat{H} \times \hat{W} \times C)}$을 입력으로 사용하여 클러스터링 데이터 X_s의 초기값을 연산한다. 이때 t는 클러스터링을 반복하는 횟수를 의미한다. 이후 수식 2와 3에 따라서 클러스터링 반복 횟수 t를 증가시킨다. 초기 클러스터링 데이터 X_s^0는 다운샘플링 된 출력값 X_{out}을 $P \times P$ 평균 풀링하여 얻은 $X_s^0 \in R^{(\frac{H}{P} \times \frac{W}{P} \times C)}$ 이다. 그리고 수식 4와 같이, 유사도 S^t는 $X(t=1$일 때, $X = X_{out})$와 X_s^{t-1}의 토큰 간의 유사도 계산을 위해 내적합 연산을 한다. 연산된 유사도는 소프트맥스 함수로 확률화 한다. 최종적으로 확률화 된 유사도 정보를 정규화한 유사도 정보 S^t는 식4와 같이 가중합되어 X_s의 값에 재입력된다.

$$X_s^0 = AvgPool(X_{out}) \tag{2}$$

$$S^{t-1} = Softmax(X \cdot (X_s^{t-1})^T) \tag{3}$$

$$X_s^t = (S_{Norm}^{t-1})^T \cdot X \tag{4}$$

최종적으로 합성곱 연산된 특징맵 X의 28×28개의 토큰은 2×2개의 유사한 토큰끼리 클러스터링 된다. 이는 그림 2의 최종 출력 이미지와 같이 유사한 토큰끼리 클러스터링 되는 것을 알 수 있다. 그리고 이것은 전역 연결 레이어 1개를 통해 차원을 맞추어 최종적인 임베딩 출력값을 얻는다.

제안하는 경량 모듈에서 사용하는 합성곱 레이어는 EarlyConv에서 제안하는 합성곱 레이어 보다 채널수가 3배 적은 합성곱을 사용하였지만, 지역적인 정보를 추출하는 합성곱 레이어와 학습 파라미터를 사용하지 않은 슈퍼토큰 클러스터링으로 표 1처럼 합성곱 임베딩 모듈에 10배 적은 학습 파라미터를 사용한다. 또한 ViT에서 제안한 패치 임베딩에 비해서도 3배 적은 학습 파라미터를 사용한다.

표 1. 임베딩 모듈 별 학습 파라미터 수 비교
Table 1. Comparsion of learnable parameters of embedding modules

Embedding	Learnable Parameter(M)	Dimension
Patch Embedding	0.30M	384
Convolution Embedding	1.00M	384
Proposed Embedding	0.09M	384

표 2. 실험 하이퍼파라미터 및 값
Table 2. Experiment hyperparameter and value

Experiment Configuration	Value
Optimizer	AdamW
Base Learing Rate	1e-3
Weight Decay	0.05
Batch Size	128
Mixup	1.0
Cutmix	0.8

Ⅳ. 실험 및 결과 분석

4-1 실험 설정

1) 실험 설정 및 데이터셋

제안하는 슈퍼토큰 합성곱 임베딩 모듈에 대한 정량적 성능 평가를 위해 ViT 논문에서 제안한 패치 임베딩 방법, EarlyConv에서 제안한 합성곱 임베딩 방법과 함께 이미지 분류 성능을 비교하였다. 실험을 위해 GPU는 NVIDIA Titan-V 두 개를 병렬구조로 사용하였다.

모든 실험에는 ViT-S(Small) 모델을 사용하였고, 각 임베딩에 따라 생성되는 토큰의 개수는 ViT의 기존 패치 임베딩 과정과 동일하게 196개로 설정하였다.

제안하는 모듈은 28×28 이미지 크기까지 합성곱 다운샘플링하여 클러스터링을 하였다. 초기 클러스터링 데이터를 생성하기 위한 평균 풀링은 2×2로 설정하였다. 그리고 클러스터 과정에서 유사도 갱신 횟수는 $t = 3$으로 설정하였다.

표 3. 패치 임베딩 모듈 구조
Table 3. Patch embedding module structure

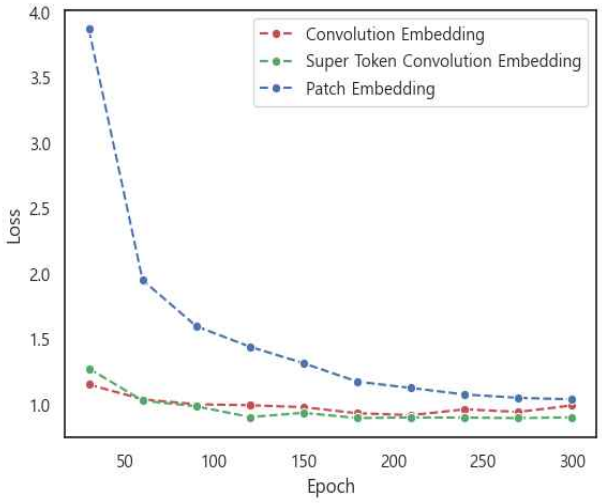
Layer	Output	Channels
Reshape	(196,768)	768
LayerNorm	(196,768)	768
FC	(196,768, 384)	384
LayerNorm	(196,384)	384

표 4. 합성곱 임베딩 모듈 구조
Table 4. Convolutional embedding module structure

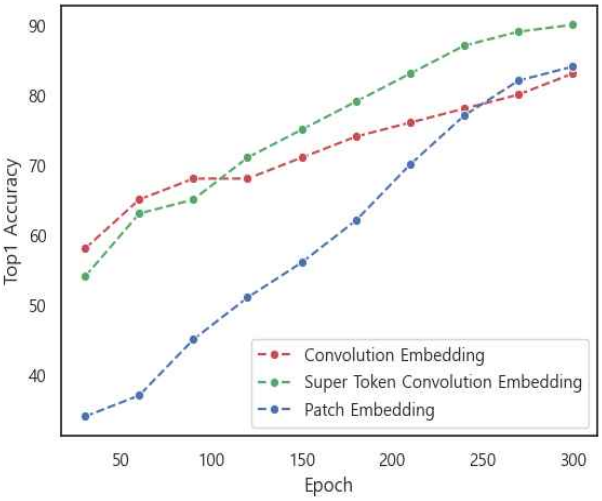
Layer	Output	Channels	Kernel size	Strides
Convolution	(112,112,96)	96	3x3	2
Convolution	(56,56,192)	192	3x3	2
Convolution	(28,28,384)	384	3x3	2
Convolution	(14,14,384)	384	3x3	2
Convolution	(14,14,384)	384	3x3	1
Reshape	(196,384)	384	–	–

표 5. 슈퍼토큰 합성곱 임베딩 모듈 구조
Table 5. SuperToken convolution embedding module structure

Layer	Output	Channels	Kernel size	Strides
Convolution Module1	(112,112,16)	16	3x3	2
Convolution Module 2	(112,112,32)	32	3x3	1
Convolution Module 3	(56,56,32)	32	3x3	2
Convolution Module 4	(56,56,64)	64	3x3	1
Convolution Module 5	(28,28,64)	64	3x3	2
Convolution Module 6	(28,28,64)	64	3x3	1
Super Token Clustering	(14,14,64)	64	–	–
Reshape	(196, 64)	64	–	–
FC	(196,384)	384	–	–



(a) 각 모듈별 Loss 그래프



(b) 각 모듈별 Top1 Accuracy 그래프

그림 3. 임베딩 모듈별 모델 학습 결과
Fig. 3. Model training results by embedding module

각 임베딩 모듈을 학습 시 설정한 하이퍼파라미터는 표 2와 같이 동일하게 적용하였다. 데이터 증강 기법은 DeiT[20]의 설정과 동일하다. 실험 및 평가를 위해 ImageNet-1k 데이터셋을 랜덤하게 100개의 클래스만을 샘플링한 ImageNet100 데이터셋을 사용하였다. 총 100개 클래스의 학습 데이터셋 130,000장, 테스트 데이터셋 5,000장으로 구성되어 있다. 본 실험은 사전학습 없이 스크래치부터 학습하였으며, 평가는 정확도 지표를 사용하였으며, 수식 5와 같이 계산한다.

$$\text{정확도} = \frac{\text{올바른 예측의 수}}{\text{전체 테스트 클래스의 수}} \times 100\% \quad (5)$$

2) 임베딩 모듈 구성

각 임베딩 모듈의 실험을 위한 구성은 다음과 같다. 첫 번째 ViT의 임베딩 모듈의 구성은 표 3과 같다. 텐서의 재배열

과 레이어 정규화(Layer Normalization)[21] 이후 전역 연결 레이어를 통하여 임베딩 차원을 맞추는 단순한 구조이다.

두 번째, 합성곱 임베딩은 ViT의 임베딩 과정을 합성곱 연산과 텐서 재배열만 사용하였다. 구체적인 합성곱 임베딩의 모듈 구성은 표 4와 같다.

마지막으로, 본 논문에서 제안하는 슈퍼토큰 합성곱 임베딩 모듈은 표 5와 같다. 이는 합성곱 임베딩 모듈과 채널 수 및 합성곱 이후 토큰 클러스터링을 수행하는 차이가 있다. 기존 합성곱 임베딩 모듈과 동일하게 합성곱 세부 모듈은 수식 1과 같이 합성곱, GELU, 배치정규화 순서로 구성되어 있다.

4-2 실험 결과 분석

1) ImageNet100 실험 결과 분석

임베딩 모듈에 따른 이미지 분류 성능 결과는 표 6에 따르면, 제안하는 슈퍼토큰 합성곱 임베딩 방법은 83.52%로 기존 패치 임베딩 방법에 비해 5.08% 성능이 향상되었다. 또한 합성곱 임베딩 방법과 비교하여 2.38% 성능이 향상하였다.

그림 3의 (a)와 (b)에 따르면, 총 300번의 epoch에 따라 제안하는 방법이 다른 모듈에 비해 학습 수렴속도가 가장 빠르고 안정적임을 알 수 있다. 학습 초기에는 합성곱 임베딩보다는 낮은 정확도를 보였으나, 학습이 진행됨에 따라 더 빠르게 개선되어 나중에는 가장 높은 성능을 달성한다.

하지만 각 임베딩 모듈의 최적의 수렴 구간이 다르기 때문에 보다 정확한 비교가 필요하다. 이에 따라 패치 임베딩, 합성곱 임베딩은 추가적인 학습이 필요하다. 수렴 속도를 고려하여 총 500 epoch 횟수에 따라 두 개의 모듈을 재학습하였다. 기존 300 epoch 횟수의 학습 결과 대비 패치 임베딩은 0.76%, 합성곱 임베딩은 1.58% 성능이 향상한다. 실험 결과에 따르면 두 모듈은 최적의 수렴을 달성했지만, 제안하는 방법에 비교하여 성능이 여전히 낮은 것을 관찰할 수 있다.

결론적으로 제안하는 슈퍼토큰 합성곱 임베딩 모듈은 ViT 모델의 학습 과정의 효율성 향상에 도움 될 뿐 아니라, 가장 적은 파라미터를 사용하면서 높은 분류 성능을 도출하였다.

2) 합성곱 세부 모듈 계층화 및 성능 비교

제안하는 방법은 합성곱 세부 모듈의 계층 구조에 따라 출력 특징맵의 정보량 차이가 발생하고, 그로 인한 슈퍼토큰 클러스터링에 직접적인 영향을 미친다. 저수준에서 고수준에 이르기까지 복잡한 특징맵을 형성하기 위해서 합성곱 세부 모듈의 최적 계층화가 필요하다.

합성곱은 이미지의 공간적 해상도를 점진적으로 감소시키지만, 이는 동시에 중요한 특징을 더욱 효율적으로 강조하는 결과를 만든다. 이에 따라 합성곱 세부 모듈의 계층화 구조에 따른 이미지 분류 성능을 실험적으로 확인하였다. 본 실험은 공통적으로 300 epoch만을 실험하였다.

표 6. 임베딩 모듈 별 ImageNet100의 Top-1 분류 성능 비교
Table 6. Compare ImageNet100's Top-1 classification performance by embedding module

Embedding	Learnable Parameter(M)	Top-1 Accuracy(%)	Epochs
Patch Embedding	18M	78.44(+0.76)	300(+200)
Convolution Embedding	20M	81.14(+1.58)	300(+200)
Token Clustering Embedding	17M	83.52	300

표 7. 합성곱 세부 모듈의 계층화에 따른 성능 비교
Table 7. Comparing the performance of convolutional detail modules as they are tiered

Modules	Output	Learnable Parameter(M)	Top-1 Accuracy(%)
w/o Convolution Module	(224, 224)	17.7M	48.26
Convolution Module1	(224, 224)	17.8M	80.20
Convolution Module 1 ~ 2	(112, 112)	17.8M	80.68
Convolution Module 1 ~ 4	(56, 56)	17.8M	82.18
Convolution Module 1 ~ 6	(28, 28)	17.9M	83.52
Convolution Module 1 ~ 6 + Additional Convolution	(14, 14)	17.9M	81.50

표 8. 유사도 정보 갱신에 따른 분류 성능 및 Throughput 결과
Table 8. Classification performance and throughput results based on the number of times similarity information updated

t	Throughput(Image/Sec)	Top-1 Accuracy(%)
1	766	82.96
2	749	83.02
3	737	83.52
4	718	83.30

표 7에 따르면, 합성곱 세부 모듈의 계층화는 세부 모듈의 구성을 점진적으로 늘릴수록 분류의 성능이 높아지는 경향이 있다. 이는 제안하는 합성곱 세부 모듈이 클러스터링에 매우 효과적으로 동작함을 입증한다. 가장 높은 성능을 달성하는 계층화는 83.52%인 모듈 1~6까지 계층화한 구조이다. 모듈 1~6에서 추가 레이어(합성곱 연산)를 통해 출력 특징맵을 14×14을 사용하면 성능이 2.02% 하락한다. 이는 반복되는 다운샘플링으로 인해 원본 이미지의 정보 손실량이 기준치를 초과하여 유의미한 클러스터링이 원활하게 동작하지 않음을 시사한다.

3) 합성곱 세부 모듈 제거

제안하는 슈퍼토큰 합성곱 임베딩 모듈 설계 시, 임베딩 모듈의 경량화를 위한 합성곱 세부 모듈 전체를 제거하고 슈퍼토큰 클러스터링만을 사용하는 실험을 하였다.

표 7의 '모듈 제거'는 합성곱 세부 모듈을 사용하지 않으며 오직 슈퍼토큰 클러스터링만을 사용한 실험이다. 그 결과 정확도는 48.26%를 보여준다. 이는 동일한 이미지 크기 대비 31.94% 성능이 하락하였다.

일반적으로 ViT는 합성곱 신경망에 비해 귀납적 편향 약하기 때문에 Mixup[22], Cutmix[23]와 같은 정규화(Regularization)가 학습에 필요하다. 하지만 이러한 정규화는 원본 이미지의 혼합된 레이블을 생성하기 때문에 이에 직접적인 슈퍼토큰 클러스터링을 적용한다면 의미론적인 토큰의 그룹화가 어려워진다. 이것은 모델이 이미지의 중요한 관계를 포착하기에 정보 손실이 높아져 있기 때문에 객체, 텍스쳐, 모양과 같은 특징을 학습하기 어렵다. 그렇기 때문에 제안하는 모듈은 합성곱 레이어를 통해서 원본 이미지를 저차원의 특정 공간에 매핑하여 보편화된 특징을 갖도록 유도한다. 그리고 강한 정규화로 인한 혼합된 레이블의 학습 악영향을 최소화하면서 유의미한 특징을 더욱 잘 학습할 수 있게 돕는다. 본 실험을 통하여 합성곱 세부 모듈은 슈퍼토큰 클러스터링을 위하여 반드시 필요한 모듈임을 입증하였다.

4) 토큰 클러스터링의 유사도 정보 갱신

토큰 클러스터링에서 입력 데이터 X_{in}의 토큰과 클러스터링 된 데이터 X_s의 토큰의 유사도를 연산하는 과정은 반복적으로 수행한다. 실험적으로 유사도 정보 갱신 횟수가 많을수록 성능은 향상이 향상되지만, 갱신하는 정보의 횟수가 많아지며 학습 속도가 느려지는 문제가 발생한다. 따라서 최적의 반복 횟수를 찾기 위하여 반복 횟수에 따른 이미지 분류 성능 및 이미지의 초당 처리속도인 Throughput을 계산한다. 실험에서 $t=1$인 경우는 X_s의 초기값이 갱신되지 않은 상태를 의미한다. 모든 실험은 Batch를 128로 고정하여 학습하였으며, Throughput은 RTX-3060ti에서 계산하였다.

실험 결과인 표 8에 따르면, 갱신 횟수가 증가함에 따라 Throughput이 초당 766개 이미지에서 718개로 점진적으로 감소한다. 이는 갱신 횟수가 반복 진행됨에 따라 클러스터링의 복잡성이 증가 및 연산 비효율성으로 인해 처리 속도가 느려진다. 분류 성능의 결과로 초기값을 사용하였을 때 82.96%로 시작하여 점진적으로 증가하여, $t=2$일 때 83.52%로 증가한다. 이러한 경향성은 유사도 갱신 과정의 반복을 거치면서 모델이 이미지 분류 능력을 개선하고 있음을 시사한다. 하지만 $t=4$ 일 때, 성능이 소폭 하락하는 것으로 보아 유사도 갱신 횟수의 지나친 반복은 모델의 이미지 분류 정확도에 악영향을 미치는 것을 알 수 있다.

결과적으로 유사도 정보 갱신의 반복은 모델을 학습하기 위한 최적화 과정에 도움 되지만, Throughput과 정확도에 Trade-off 관계가 있음을 확인하였다. 실험적으로 제안하는 슈퍼토큰 합성곱 임베딩 모듈은 $t=3$으로 설정하는 것이 최적이다.

V. 결 론

ViT 연구에서 제안한 패치 임베딩은 이미지 내의 픽셀의 지역적인 관계성을 고려하지 않고 이미지를 임베딩 한다. 이러한 점을 개선하고자 합성곱 임베딩이 제안되었다. 합성곱 임베딩은 ViT의 임베딩 문제를 개선하였다. 하지만 ViT의 패치 임베딩에 비해 3배 많은 학습 파라미터를 사용하는 단점이 존재한다.

이러한 점을 개선하고자 본 논문은 경량화된 합성곱 임베딩 모듈을 제안하였다. 제안하는 모듈인 슈퍼토큰 합성곱 임베딩 모듈은 합성곱의 채널을 3배 적게 사용함과 더불어 합성곱 레이어 일부를 슈퍼토큰 클러스터링으로 대체한다. 이를 통해 학습 파라미터를 10배 줄여 임베딩 모듈을 경량화하였다.

ImageNet100 데이터셋 이미지 분류 정확도 비교를 통해, 제안하는 슈퍼토큰 합성곱 임베딩 모듈의 성능을 검증하였다. 실험 결과로 ViT의 기존 패치 임베딩 및 합성곱 임베딩 방식에 비해 우수한 성능 향상을 확인하였다. 그리고 슈퍼토큰 합성곱 임베딩 모듈이 갖는 다양한 하이퍼파라미터에 대해서 이를 최적화하는 실험을 수행하였다. 결론적으로 제안하는 방법은 기존 임베딩 모듈에 비해 훨씬 경량화된 모델 학습 파라미터를 사용하면서도 이미지 분류 정확도 성능을 상당히 개선하였다.

본 연구는 슈퍼토큰 합성곱 임베딩 모듈의 성능 평가를 ViT 모델을 기준으로 평가하였다. 향후 연구에서는 제안한 임베딩 모듈의 성능을 ViT의 구조를 변형한 다양한 네트워크에서 성능을 검증할 것이다.

감사의 글

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 지역지능화혁신인재양성사업임(IITP-2024-00156287)

참고문헌

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, ... and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach: CA, pp. 5998-6008, December 2017. https://doi.org/10.4855

0/arXiv.1706.03762

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, ... and N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” arXiv:2010.11929v2, June 2021. https://doi.org/10.48550/arXiv.2010.11929

[3] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, ... and B. Guo, “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows,” in *Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, Canada, pp. 9992-10002, October 2021. https://doi.org/10.1109/ICCV48922.2021.00986

[4] S. Mehta and M. Rastegari, “MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer,” arXiv:2110.02178v2, March 2022. https://doi.org/10.48550/arXiv.2110.02178

[5] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, “Vision Transformer with Deformable Attention,” in *Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans: LA, pp. 4784-4793, June 2022. https://doi.org/10.1109/CVPR52688.2022.00475

[6] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollar, and R. Girshick, “Early Convolutions Help Transformers See Better,” in *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online, pp. 30392-30400, December 2021. https://doi.org/10.48550/arXiv.2106.14881

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, “ImageNet: A Large-Scale Hierarchical Image Database,” in *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami: FL, pp. 248-255, June 2009. https://doi.org/10.1109/CVPR.2009.5206848

[8] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, “Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations,” arXiv:2202.07800v2, April 2022. https://doi.org/10.48550/arXiv.2202.07800

[9] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, “Token Merging: Your Vit but Faster,” arXiv:2210.09461v3, May 2023. https://doi.org/10.48550/arXiv.2210.09461

[10] J. B. Haurum, S. Escalera, G. W. Taylor, and T. B. Moeslund, “Which Tokens to Use? Investigating Token Reduction in Vision Transformers,” in *Proceedings of 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Paris, France, pp. 773-783, October 2023. https://doi.org/10.1109/ICCVW60793.2023.00085

[11] H. Huang, X. Zhou, J. Cao, R. He, and T. Tan, “Vision Transformer with Super Token Sampling,” arXiv:2211.11167v2, January 2024. https://doi.org/10.48550/arXiv.2211.11167

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012)*, Lake Tahoe: NV, pp. 1097-1105, December 2012.

[13] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, “CvT: Introducing Convolutions to Vision Transformers,” in *Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, Canada, pp. 22-31, October 2021. https://doi.org/10.1109/ICCV48922.2021.00009

[14] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, “Focal Self-Attention for Local-Global Interactions in Vision Transformers,” arXiv:2107.00641v1, July 2021. https://doi.org/10.48550/arXiv.2107.00641

[15] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, “CMT: Convolutional Neural Networks Meet Vision Transformers,” in *Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans: LA, pp. 12165-12175, June 2022. https://doi.org/10.1109/CVPR52688.2022.01186

[16] M. Fayyaz, S. A. Koohpayegani, F. R. Jafari, S. Sengupta, H. R. Vaezi Joze, E. Sommerlade, ... and J. Gall, “Adaptive Token Sampling for Efficient Vision Transformers,” in *Proceedings of the 17th European Conference on Computer Vision (ECCV 2022)*, Tel Aviv, Israel, pp. 396-414, October 2022. https://doi.org/10.1007/978-3-031-20083-0_24

[17] T. Suzuki, “Clustering as Attention: Unified Image Segmentation with Hierarchical Clustering,” arXiv:2205.09949v1, May 2022. https://doi.org/10.48550/arXiv.2205.09949

[18] D. Hendrycks and K. Gimpel, “Gaussian Error Linear Units (GELUs),” arXiv:1606.08415v1, June 2016. https://doi.org/10.48550/arXiv.1606.08415

[19] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, pp. 448-456, July 2015. https://doi.org/10.48550/arXiv.1502.03167

[20] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training Data-Efficient Image Transformers

& Distillation through Attention," in *Proceedings of the 38th International Conference on Machine Learning*, Online, pp. 10347-10357, July 2021. https://doi.org/10.48550/arXiv.2012.12877

[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," arXiv:1607.06450v1, July 2016. https://doi.org/10.48550/arXiv.1607.06450

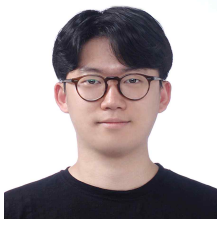
[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond Empirical Risk Minimization," arXiv: 1710.09412v1, October 2017. https://doi.org/10.48550/arXiv.1710.09412

[23] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features," in *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, pp. 6022-6031, October-November 2019. https://doi.org/10.1109/ICCV.2019.00612

이진(Jin Lee)

2022년 : 전남대학교 전자컴퓨터공학부 (공학학사)

2022년～현 재: 전남대학교 ICT융합시스템공학과 석사과정
※관심분야 : 인공지능, 비전 트랜스포머, 지도학습



이주환(Ju-Hwan Lee)

2019년 : 전남대학교 지구환경과학부 (이학학사)

2019년～현 재: 전남대학교 ICT융합시스템공학과 석박통합과정
※관심분야 : 인공지능, 설명가능한 인공지능, 지식 증류

김진영(Jin-Young Kim)

1986년 : 서울대학교 전자공학과 (공학학사)
1988년 : 서울대학교 전자공학과 대학원 (공학석사)
1994년 : 서울대학교 전자공학과 대학원 (공학박사-전자공학)

1995년～현 재: 전남대학교 ICT융합시스템공학과 교수
※관심분야 : 디지털 신호처리, 영상 처리, 음성 신호처리, 머신러닝, 딥러닝