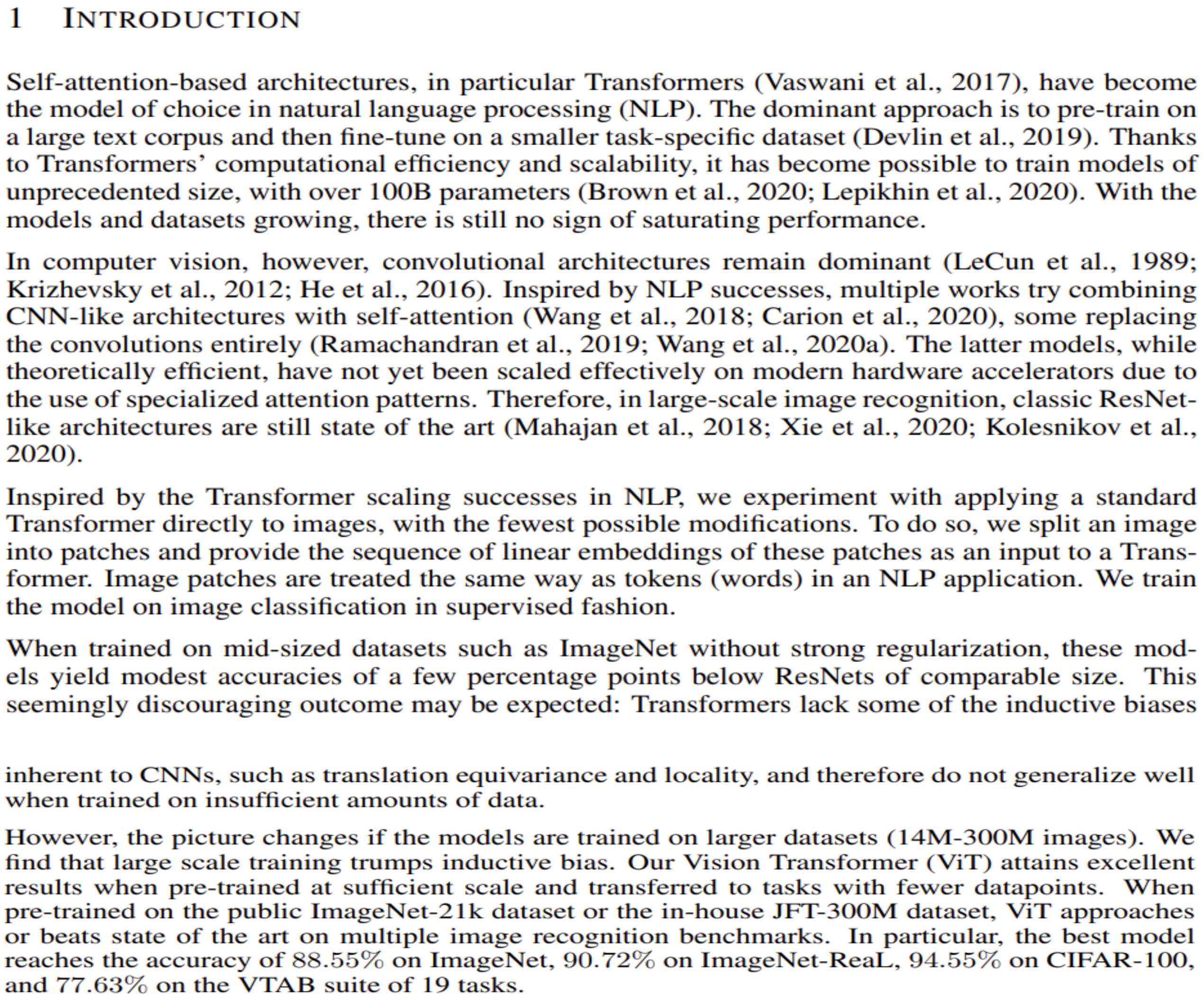
AICON Journal Club

김민수

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale



1 Introduction

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.

In computer vision, however, convolutional architectures remain dominant (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016). Inspired by NLP successes, multiple works try combining CNN-like architectures with self-attention (Wang et al., 2018; Carion et al., 2020), some replacing the convolutions entirely (Ramachandran et al., 2019; Wang et al., 2020a). The latter models, while theoretically efficient, have not yet been scaled effectively on modern hardware accelerators due to the use of specialized attention patterns. Therefore, in large-scale image recognition, classic ResNet-like architectures are still state of the art (Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020).

Inspired by the Transformer scaling successes in NLP, we experiment with applying a standard Transformer directly to images, with the fewest possible modifications. To do so, we split an image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer. Image patches are treated the same way as tokens (words) in an NLP application. We train the model on image classification in supervised fashion.

When trained on mid-sized datasets such as ImageNet without strong regularization, these models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data.

However, the picture changes if the models are trained on larger datasets (14M-300M images). We find that large scale training trumps inductive bias. Our Vision Transformer (ViT) attains excellent results when pre-trained at sufficient scale and transferred to tasks with fewer datapoints. When pre-trained on the public ImageNet-21k dataset or the in-house JFT-300M dataset, ViT approaches or beats state of the art on multiple image recognition benchmarks. In particular, the best model reaches the accuracy of 88.55% on ImageNet, 90.72% on ImageNet-ReaL, 94.55% on CIFAR-100, and 77.63% on the VTAB suite of 19 tasks.

<Introduction Summary>

Self-Attention Architecture 기반 Transformer가 NLP에서 큰 성공을 이룸.

Computer Vision에서는 CNN 기반 아키텍쳐를 기반으로 이미지를 처리하는게 지배적

Self-Attention 기반 Transformer 구조가 NLP에서 성공을 이룬 것에 영감을 받아
Vision task에 적용을 해보고자 시도하였음.

Transformer 아키텍처를 통해
실제 대규모 이미지 인식, 분류 Task에 적용하기 위하여,
Image를 Patch 형태 (NLP 단어 시퀀스 토큰 같은 형태)로 처리하여 입력으로 사용

Transformer 구조가 강력한 정규화 없이
중간 사이즈의 데이터셋(충분치 못한 데이터양)에서 훈련될 경우,

inductive bias가 부족하다.
➔ ViT는 CNN에 내재된 Inductive bias (유도 편향)가 부족함.
즉 정확한 예측을 하기 위한 추가적인 가정이 부족함.

➔ ViT는 MLP Layer에서만 Local,
Self-Attention Mechanism에서 Global한 영역 처리를 하기 때문에

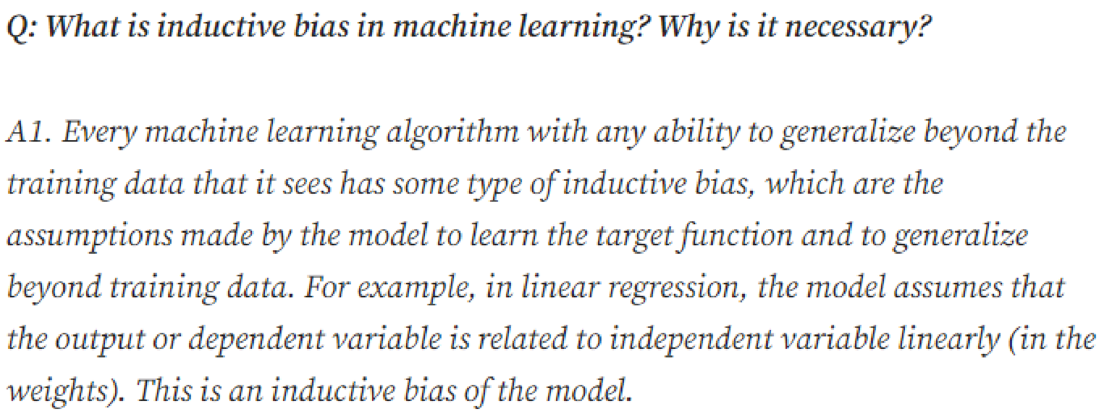
즉 학습에서 만나보지 못했던 상황에 대해 추가적인 가정이 부족함
예를 들어 아래와 같은 것
➔ Translation equivariance (입력 위치 변경->출력 위치 변경)
➔ Locality (인접한 픽셀끼리의 연관성)

그러나 **대규모 데이터셋 훈련에서는**
inductive bias가 ResNet가 비교하여 월등히 뛰어남.
-> generalization(일반화)가 잘 이루어짐

해당 ViT 연구에서 충분한 데이터 크기에서 사전 훈련된 모델을 통해

Fine-Tuning 했을 때 뛰어난 결과를 얻었음.

Inductive bias



Q: What is inductive bias in machine learning? Why is it necessary?

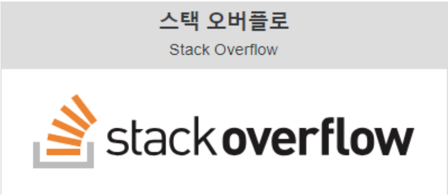
A1. Every machine learning algorithm with any ability to generalize beyond the training data that it sees has some type of inductive bias, which are the assumptions made by the model to learn the target function and to generalize beyond training data. For example, in linear regression, the model assumes that the output or dependent variable is related to independent variable linearly (in the weights). This is an inductive bias of the model.

Stackoverflow에 나온 Inductive Bias에 대한 Q&A

> 머신러닝 알고리즘은 타겟 함수를 학습하고 학습 데이터를 넘어 일반화하기 위해
모델에 의해 가정된 Inductive Bias(유도 편향)를 의도적으로 사용한다는 의미이다.

예를 들어,
Linear regression 모델은 종속 변수는 독립 변수(출력)와 선형적으로 관련이 있다는 것을 가정

출처 :



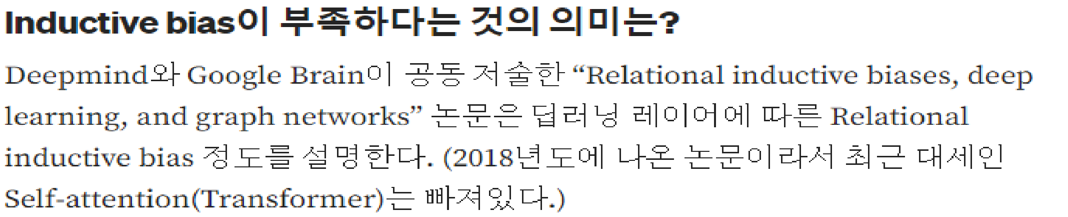
스택 오버플로

Stack Overflow

미디엄

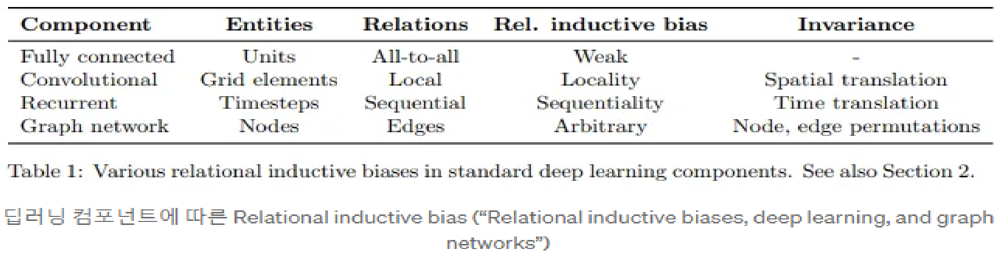
창립 2012년 8월 14일

본사 소재지 샌프란시스코



Inductive bias이 부족하다는 것의 의미는?

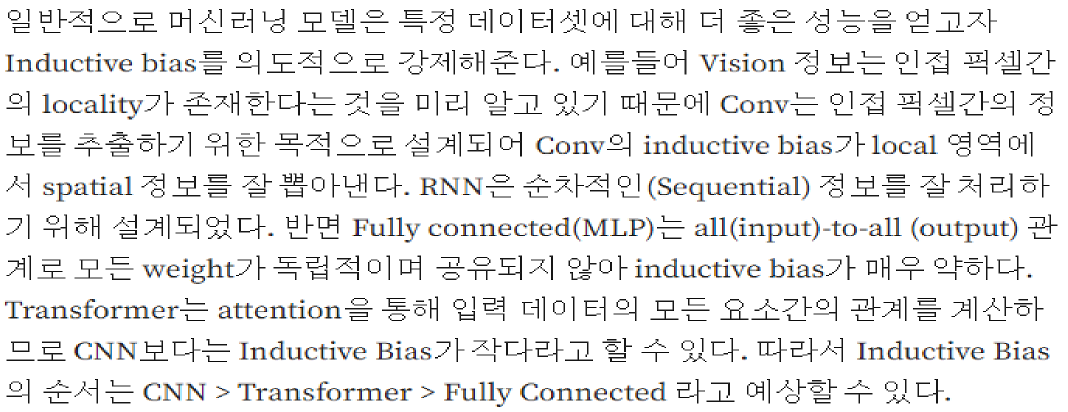
Deepmind와 Google Brain이 공동 저술한 “Relational inductive biases, deep learning, and graph networks” 논문은 딥러닝 레이어에 따른 Relational inductive bias 정도를 설명한다. (2018년도에 나온 논문이라서 최근 대세인 Self-attention(Transformer)는 빠져있다.)



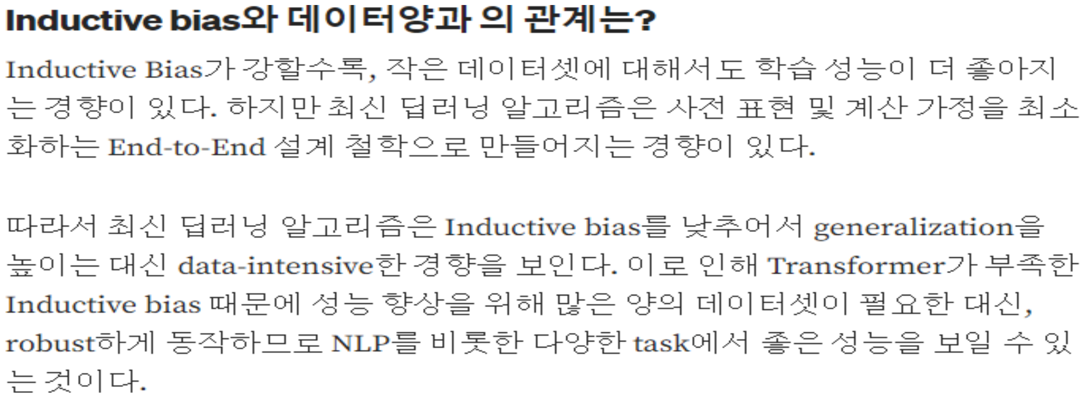
Component	Entities	Relations	Rel. inductive bias	Invariance
Fully connected	Units	All-to-all	Weak	-
Convolutional	Grid elements	Local	Locality	Spatial translation
Recurrent	Timesteps	Sequential	Sequentiality	Time translation
Graph network	Nodes	Edges	Arbitrary	Node, edge permutations

Table 1: Various relational inductive biases in standard deep learning components. See also Section 2.

딥러닝 컴포넌트에 따른 Relational inductive bias (“Relational inductive biases, deep learning, and graph networks”)



일반적으로 머신러닝 모델은 특정 데이터셋에 대해 더 좋은 성능을 얻고자 Inductive bias를 의도적으로 강제해준다. 예를들어 Vision 정보는 인접 픽셀간의 locality가 존재한다는 것을 미리 알고 있기 때문에 Conv는 인접 픽셀간의 정보를 추출하기 위한 목적으로 설계되어 Conv의 inductive bias가 local 영역에서 spatial 정보를 잘 뽑아낸다. RNN은 순차적인(Sequential) 정보를 잘 처리하기 위해 설계되었다. 반면 Fully connected(MLP)는 all(input)-to-all (output) 관계로 모든 weight가 독립적이며 공유되지 않아 inductive bias가 매우 약하다. Transformer는 attention을 통해 입력 데이터의 모든 요소간의 관계를 계산하므로 CNN보다는 Inductive Bias가 작다라고 할 수 있다. 따라서 Inductive Bias의 순서는 CNN > Transformer > Fully Connected 라고 예상할 수 있다.



Inductive bias와 데이터양과 의 관계는?

Inductive Bias가 강할수록, 작은 데이터셋에 대해서도 학습 성능이 더 좋아지는 경향이 있다. 하지만 최신 딥러닝 알고리즘은 사전 표현 및 계산 가정을 최소화하는 End-to-End 설계 철학으로 만들어지는 경향이 있다.

따라서 최신 딥러닝 알고리즘은 Inductive bias를 낮추어서 generalization을 높이는 대신 data-intensive한 경향을 보인다. 이로 인해 Transformer가 부족한 Inductive bias 때문에 성능 향상을 위해 많은 양의 데이터셋이 필요한 대신, robust하게 동작하므로 NLP를 비롯한 다양한 task에서 좋은 성능을 보일 수 있는 것이다.

Architecture

BN = Feature 단위 mean,std 계산, 정규화

LN = Data Sample 단위 mean,std 계산, 정규화
(같은 배치 동일 시점의 시퀀스)

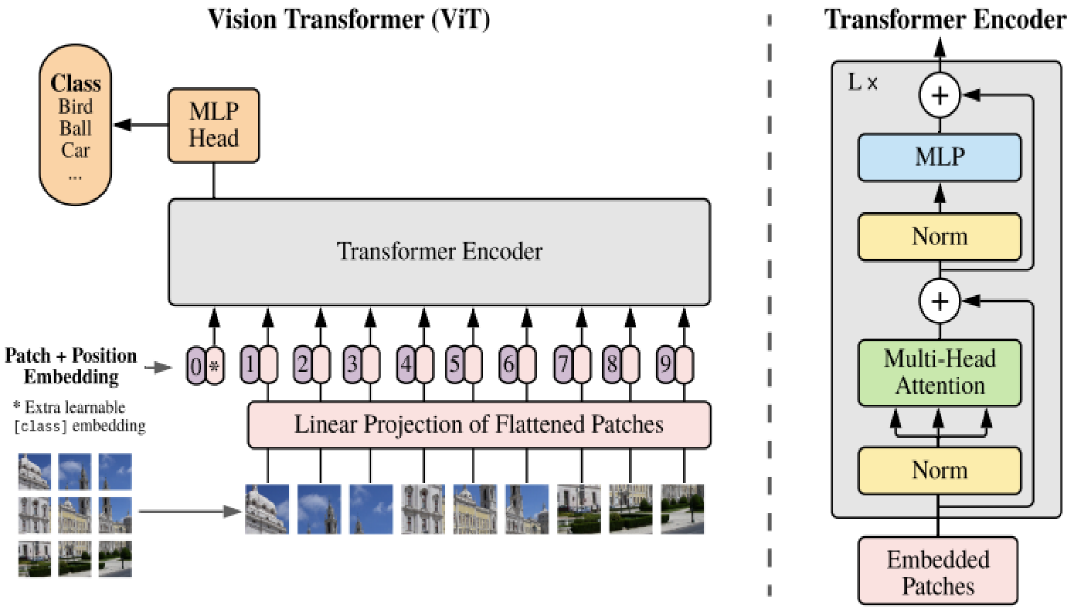


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

1. Input Image를 **Fixed-size Patch**로 분할
➔ 이미지가 겹치지 않게 Stride 설정을 Kernal_size와 동일하게

2. **Linear Projection of Flatten Patches (Patch Embedding)**
- Linear를 사용할 수 있지만 학습 퍼포먼스 향상을 위해 Conv를 권하기도 함
- Transformer Encoder Input으로 활용하기 위하여 Transpose 수행
(Batch_size, Sequence Size, Embedding Dimension)

3. CLS Token, Postional Vector가 더해져 반환된
Embedding Patch가 Transformer Encoder 통과하여 연산 수행
논문에서는 2D Positional Embedding 사용 시 성능 향상이 미미하다고 함.

4. MLP Layer
: 원래 임베딩 차원의 4배가 되도록 Linear 연산을 한번 거친 후, 원래 사이즈로 돌아오는 과정으로 구현되어 있음. (Expansion -> Reduction)

- 효과
➢ Representation 효과 증대 -> 차원을 늘려서 다양하고 복잡한 특징 파악 가능
➢ Non-Linearity (비선형성) 강화 > 어떤 Activation Function을 사용하느냐에 따라 다름

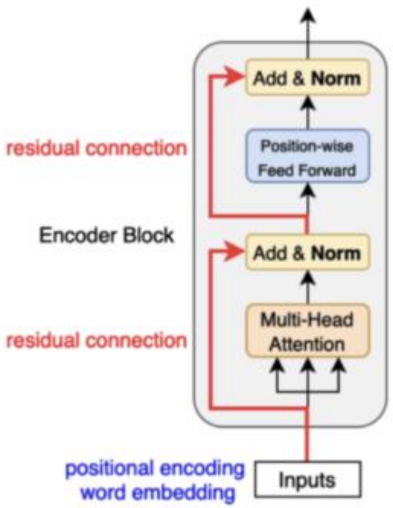
5. Pre-training & Fine Tuning 경우
Final Layer 연산 후 CLS 토큰 추출 및 **Classification linear layer** 추가
➔ 패치로 분할된 이미지 전체를 대표하는 Representation
➔ CLS Token (0번째 시퀀스)
➔ Classification Task에서 사용

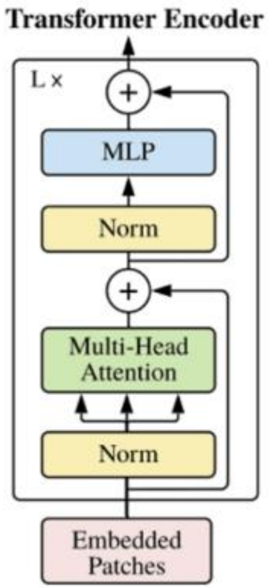
Large-Scale datasets : ImageNet-21k and JFT-300M

Method : Original Transformer

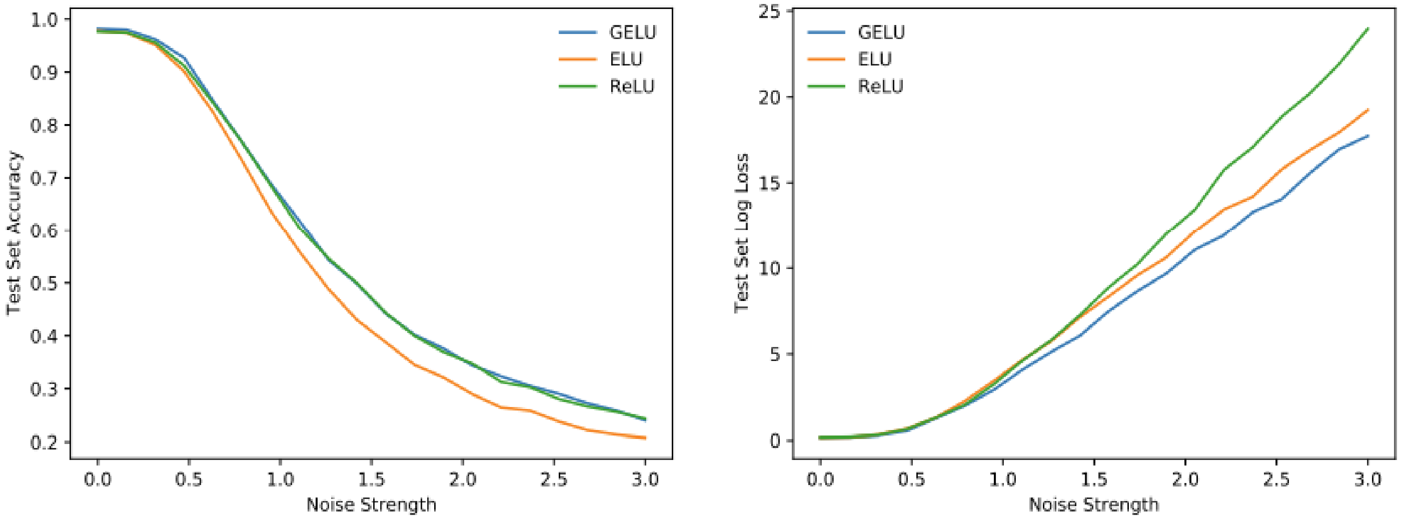
Original Transformer와의 차이점

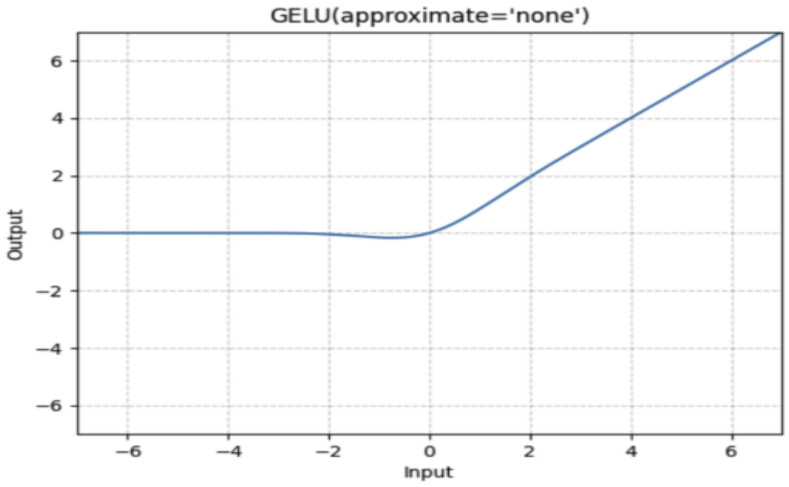
1. Attention, MLP 이전에 Layer Normalization 수행
(Transformer Encoder 변형)

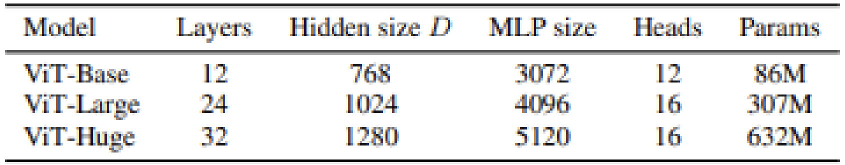




2. Relu 대신 Gaussian Error Linear Unit (**GELU**) 사용
➔NLP, Vision 다양한 Task에서 일괄적으로 좋은 성능
➔Gelu Paper : https://arxiv.org/pdf/1606.08415.pdf

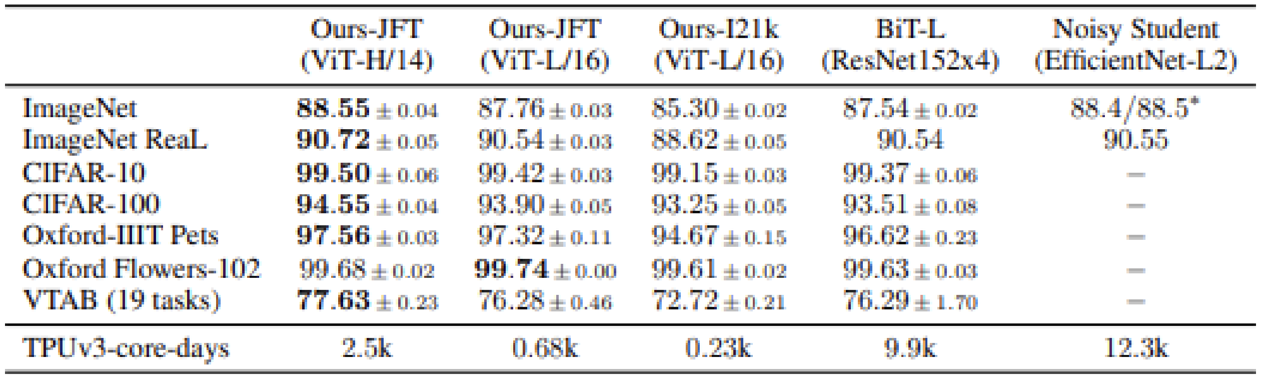






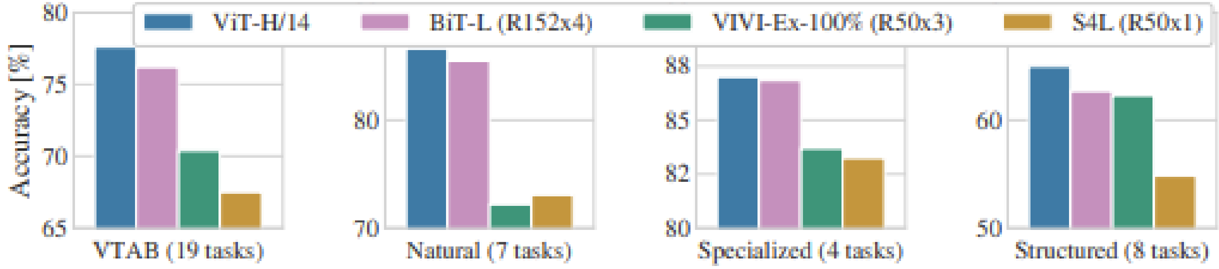
Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.



	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).



ViT Results

JFT-300M, ImageNet-21k와 같은

대규모 데이터셋으로 사전 훈련된 ViT 모델의 성능이 우수하다.

Vision Transformer Advancing Benchmarks (VTAB) 결과값에 따르면

ViT 모델이 모든 Computer Vision Task에서 성능이 좋은 것은 아니라는 것.

➔ 어떤 특정 작업에서는 CNN 기반의 모델이 여전히 경쟁력을 유지하고 있다는 점

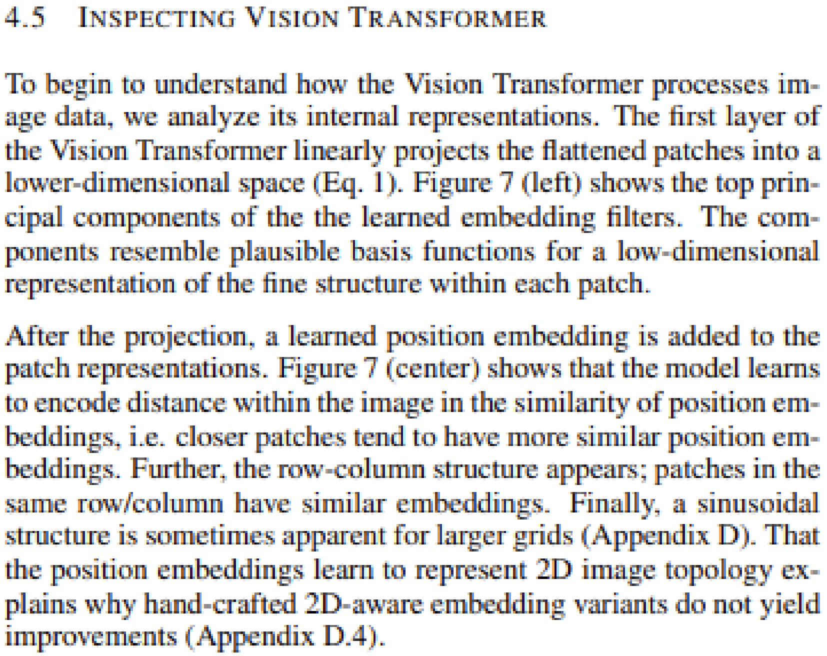
CLS Token 활용 방안

즉 ViT 모델에서 CLS Token은 최종적으로 이미지 전체를 대표하는 1차원 Representation Vector로써의 역할 수행

3D ViT 모델의 CLS 토큰 활용 방안 (3D Brain MRI Dataset)

: CNN 기반 SNUH GBM Survival Model에 3D Vision Transformer 추가 -> GBM Survival Analysis Model 확장

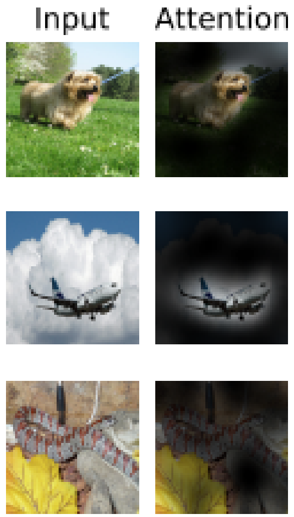
➔ CLS Token 기반 Attention Map Visualization: MRI 기반 GBM 이미지의 어떤 패치가 생존 예측에 영향을 미치는지 시각화

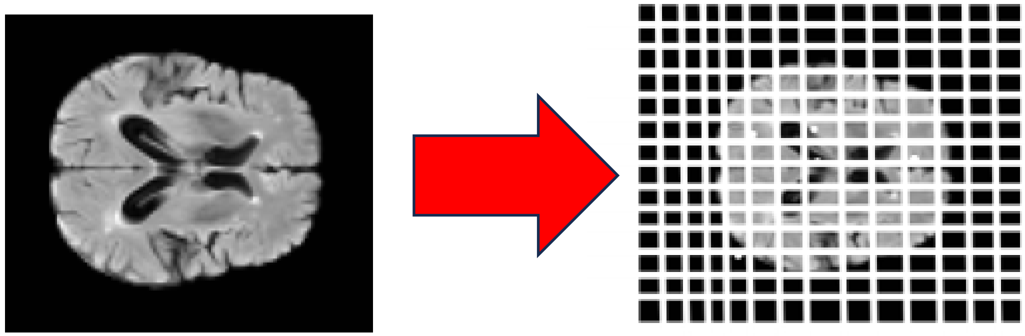


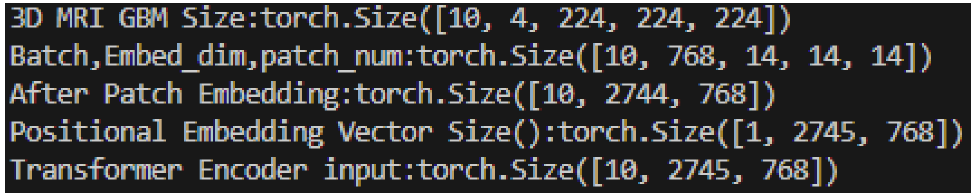
4.5 INSPECTING VISION TRANSFORMER

To begin to understand how the Vision Transformer processes image data, we analyze its internal representations. The first layer of the Vision Transformer linearly projects the flattened patches into a lower-dimensional space (Eq. 1). Figure 7 (left) shows the top principal components of the the learned embedding filters. The components resemble plausible basis functions for a low-dimensional representation of the fine structure within each patch.

After the projection, a learned position embedding is added to the patch representations. Figure 7 (center) shows that the model learns to encode distance within the image in the similarity of position embeddings, i.e. closer patches tend to have more similar position embeddings. Further, the row-column structure appears; patches in the same row/column have similar embeddings. Finally, a sinusoidal structure is sometimes apparent for larger grids (Appendix D). That the position embeddings learn to represent 2D image topology explains why hand-crafted 2D-aware embedding variants do not yield improvements (Appendix D.4).

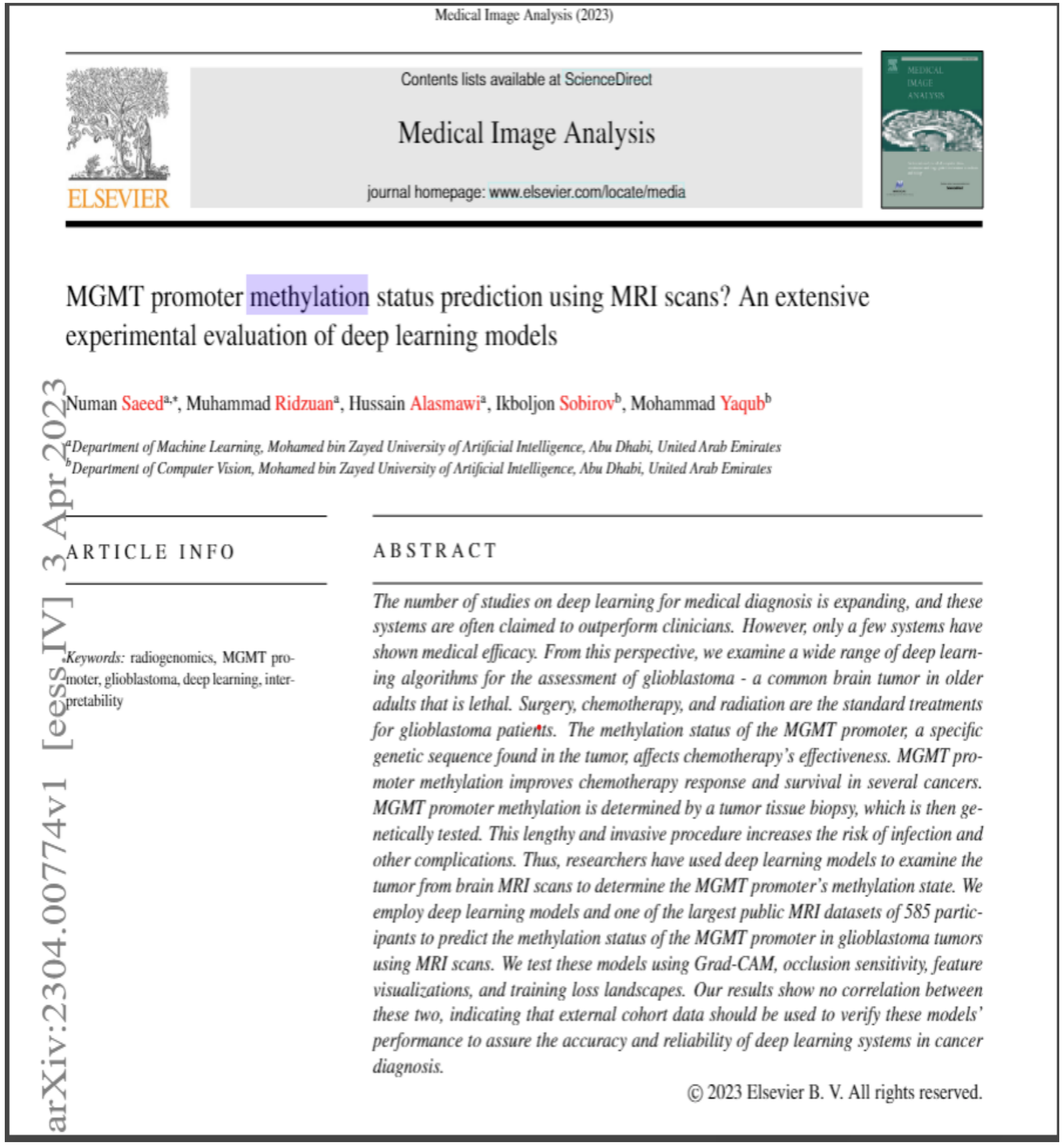






```
3D MRI GBM Size:torch.Size([10, 4, 224, 224, 224])
Batch,Embed_dim,patch_num:torch.Size([10, 768, 14, 14, 14])
After Patch Embedding:torch.Size([10, 2744, 768])
Positional Embedding Vector Size():torch.Size([1, 2745, 768])
Transformer Encoder input:torch.Size([10, 2745, 768])
```

기존 연구 : 뇌교종 환자 생존에 영향을 미치는 MGMT 유전체 상태 예측



Medical Image Analysis (2023)

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

ELSEVIER

MGMT promoter methylation status prediction using MRI scans? An extensive experimental evaluation of deep learning models

Numan Saeed[a,*], Muhammad Ridzuan[a], Hussain Alasmawi[a], Ikboljon Sobirov[b], Mohammad Yaqub[b]

[a]Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates
[b]Department of Computer Vision, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates

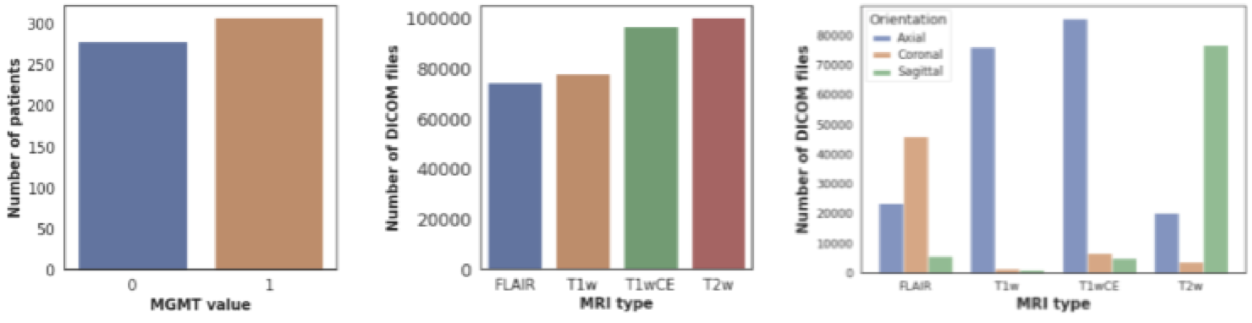
ARTICLE INFO

Keywords: radiogenomics, MGMT promoter, glioblastoma, deep learning, interpretability

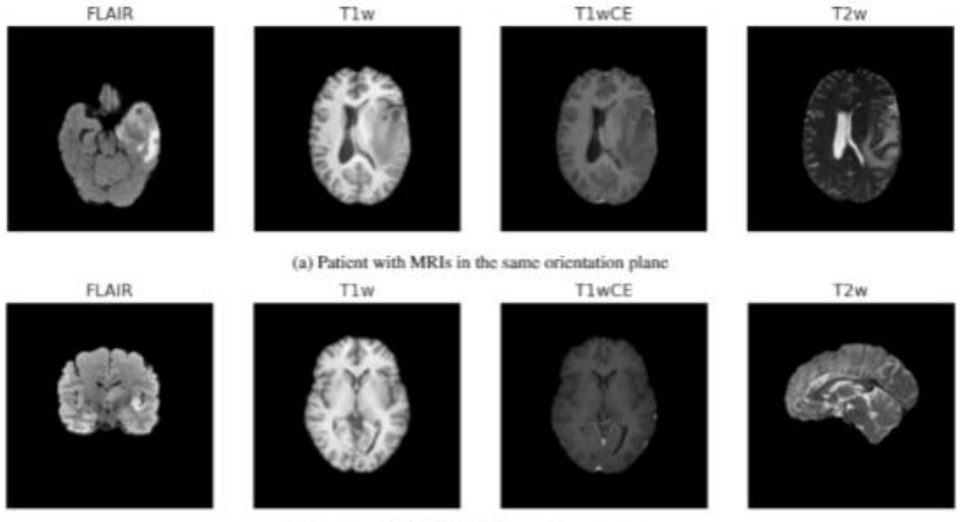
ABSTRACT

The number of studies on deep learning for medical diagnosis is expanding, and these systems are often claimed to outperform clinicians. However, only a few systems have shown medical efficacy. From this perspective, we examine a wide range of deep learning algorithms for the assessment of glioblastoma - a common brain tumor in older adults that is lethal. Surgery, chemotherapy, and radiation are the standard treatments for glioblastoma patients. The methylation status of the MGMT promoter, a specific genetic sequence found in the tumor, affects chemotherapy's effectiveness. MGMT promoter methylation improves chemotherapy response and survival in several cancers. MGMT promoter methylation is determined by a tumor tissue biopsy, which is then genetically tested. This lengthy and invasive procedure increases the risk of infection and other complications. Thus, researchers have used deep learning models to examine the tumor from brain MRI scans to determine the MGMT promoter's methylation state. We employ deep learning models and one of the largest public MRI datasets of 585 participants to predict the methylation status of the MGMT promoter in glioblastoma tumors using MRI scans. We test these models using Grad-CAM, occlusion sensitivity, feature visualizations, and training loss landscapes. Our results show no correlation between these two, indicating that external cohort data should be used to verify these models' performance to assure the accuracy and reliability of deep learning systems in cancer diagnosis.

Numan Saeed *et al.* / Medical Image Analysis (2023)



(a) (b) (c)



(a) Patient with MRIs in the same orientation plane

(b) Patient with MRIs in different orientation planes

출처 : ELSEVIER

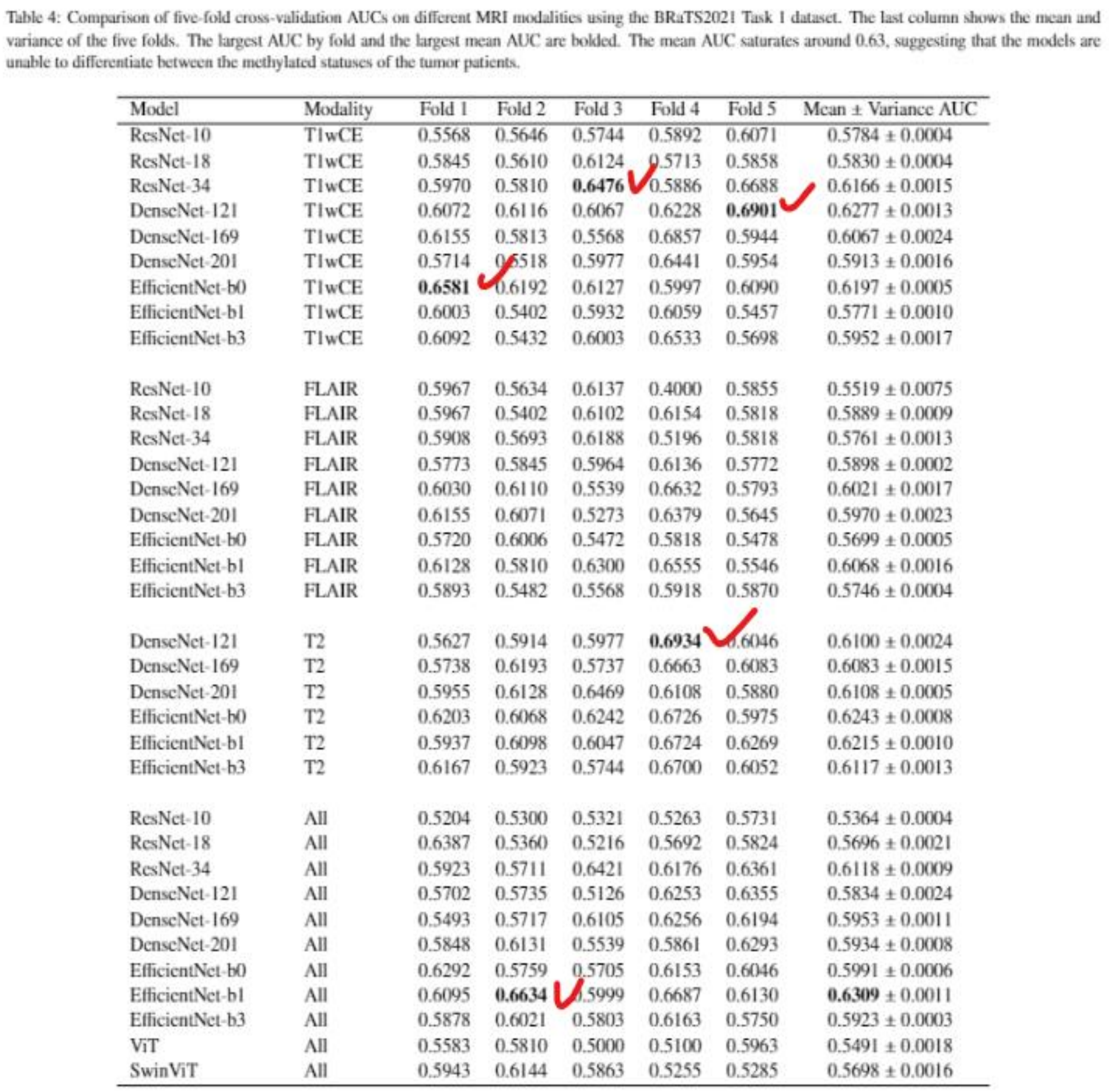


Table 4: Comparison of five-fold cross-validation AUCs on different MRI modalities using the BRaTS2021 Task 1 dataset. The last column shows the mean and variance of the five folds. The largest AUC by fold and the largest mean AUC are bolded. The mean AUC saturates around 0.63, suggesting that the models are unable to differentiate between the methylated statuses of the tumor patients.

Model	Modality	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean ± Variance AUC
ResNet-10	T1wCE	0.5568	0.5646	0.5744	0.5892	0.6071	0.5784 ± 0.0004
ResNet-18	T1wCE	0.5845	0.5610	0.6124	0.5713	0.5858	0.5830 ± 0.0004
ResNet-34	T1wCE	0.5970	0.5810	**0.6476**	0.5886	0.6688	0.6166 ± 0.0015
DenseNet-121	T1wCE	0.6072	0.6116	0.6067	0.6228	**0.6901**	0.6277 ± 0.0013
DenseNet-169	T1wCE	0.6155	0.5813	0.5568	0.6857	0.5944	0.6067 ± 0.0024
DenseNet-201	T1wCE	0.5714	0.6518	0.5977	0.6441	0.5954	0.5913 ± 0.0016
EfficientNet-b0	T1wCE	**0.6581**	0.6192	0.6127	0.5997	0.6090	0.6197 ± 0.0005
EfficientNet-b1	T1wCE	0.6003	0.5402	0.5932	0.6059	0.5457	0.5771 ± 0.0010
EfficientNet-b3	T1wCE	0.6092	0.5432	0.6003	0.6533	0.5698	0.5952 ± 0.0017
ResNet-10	FLAIR	0.5967	0.5634	0.6137	0.4000	0.5855	0.5519 ± 0.0075
ResNet-18	FLAIR	0.5967	0.5402	0.6102	0.6154	0.5818	0.5889 ± 0.0009
ResNet-34	FLAIR	0.5908	0.5693	0.6188	0.5196	0.5818	0.5761 ± 0.0013
DenseNet-121	FLAIR	0.5773	0.5845	0.5964	0.6136	0.5772	0.5898 ± 0.0002
DenseNet-169	FLAIR	0.6030	0.6110	0.5539	0.6632	0.5793	0.6021 ± 0.0017
DenseNet-201	FLAIR	0.6155	0.6071	0.5273	0.6379	0.5645	0.5970 ± 0.0023
EfficientNet-b0	FLAIR	0.5720	0.6006	0.5472	0.5818	0.5478	0.5699 ± 0.0005
EfficientNet-b1	FLAIR	0.6128	0.5810	0.6300	0.6555	0.5546	0.6068 ± 0.0016
EfficientNet-b3	FLAIR	0.5893	0.5482	0.5568	0.5918	0.5870	0.5746 ± 0.0004
DenseNet-121	T2	0.5627	0.5914	0.5977	**0.6934**	0.6046	0.6100 ± 0.0024
DenseNet-169	T2	0.5738	0.6193	0.5737	0.6663	0.6083	0.6083 ± 0.0015
DenseNet-201	T2	0.5955	0.6128	0.6469	0.6108	0.5880	0.6108 ± 0.0005
EfficientNet-b0	T2	0.6203	0.6068	0.6242	0.6726	0.5975	0.6243 ± 0.0008
EfficientNet-b1	T2	0.5937	0.6098	0.6047	0.6724	0.6269	0.6215 ± 0.0010
EfficientNet-b3	T2	0.6167	0.5923	0.5744	0.6700	0.6052	0.6117 ± 0.0013
ResNet-10	All	0.5204	0.5300	0.5321	0.5263	0.5731	0.5364 ± 0.0004
ResNet-18	All	0.6387	0.5360	0.5216	0.5692	0.5824	0.5696 ± 0.0021
ResNet-34	All	0.5923	0.5711	0.6421	0.6176	0.6361	0.6118 ± 0.0009
DenseNet-121	All	0.5702	0.5735	0.5126	0.6253	0.6355	0.5834 ± 0.0024
DenseNet-169	All	0.5493	0.5717	0.6105	0.6256	0.6194	0.5953 ± 0.0011
DenseNet-201	All	0.5848	0.6131	0.5539	0.5861	0.6293	0.5934 ± 0.0008
EfficientNet-b0	All	0.6292	0.5759	0.5705	0.6153	0.6046	0.5991 ± 0.0006
EfficientNet-b1	All	0.6095	**0.6634**	0.5999	0.6687	0.6130	**0.6309** ± 0.0011
EfficientNet-b3	All	0.5878	0.6021	0.5803	0.6163	0.5750	0.5923 ± 0.0003
ViT	All	0.5583	0.5810	0.5000	0.5100	0.5963	0.5491 ± 0.0018
SwinViT	All	0.5943	0.6144	0.5863	0.5255	0.5285	0.5698 ± 0.0016

특징

1. MRI Data 기반 MGMT Value + MRI Type + Deep Learning 활용
2. Transformer, CNN 기반 모델 AUC 성능이 낮음

   ViT : Mean ± Variance AUC = [0.5473, 0.5509]

   ResNet-18 : Mean ± Variance AUC = [0.5826, 0.5834]

   DenseNet-121 : Mean ± Variance AUC = [0.6264, 0.6290]

후속 연구로써의 차별성

1. MGMT 예측 -> 다양한 유전체 정보(PTEN, TERTp) 제공