[2026 winter seminar]

CVSP Basic

Pusan National University
Computer Vision & Signal Processing Lab

Presented By
윤상연

CVSP Lab
Computer Vision & Signal Processing

Outline

1) Dosovitskiy et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. *(ICLR)*
2) Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261* (2018).
3) Raghu et al. (2021). Do vision transformers see like convolutional neural networks?. Advances in *Neural Information Processing Systems (NeurIPS)*, 34, 12116-12128.

- Basic Course (CS231n, Stand Alone Deep Learning)
  - Basic Architectures (K-NN / Linear / MLP / CNN)
  - Loss / Regularization / Optimization

- Vision Transformer (ViT)
  - ViT - Architecture & Experiment [1)]
  - ViT vs. CNN: focusing on Inductive Bias [2, 3)]

CVSP Lab
Computer Vision & Signal Processing

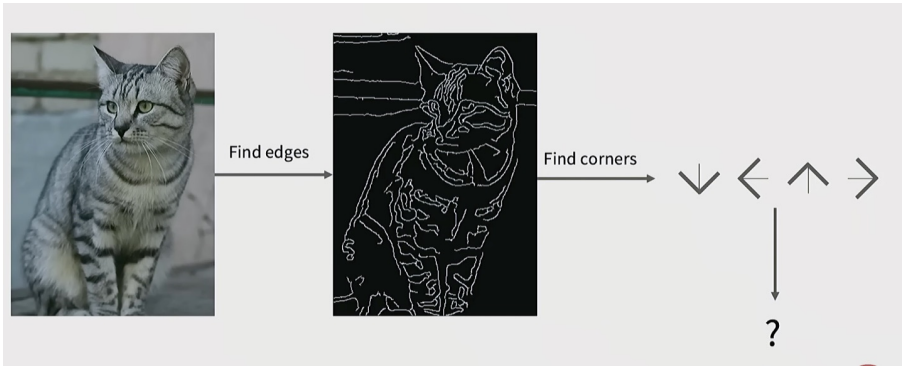
Basic Course
(CS231n, Stand Alone Deep Learning)

CVSP Lab
Computer Vision & Signal Processing

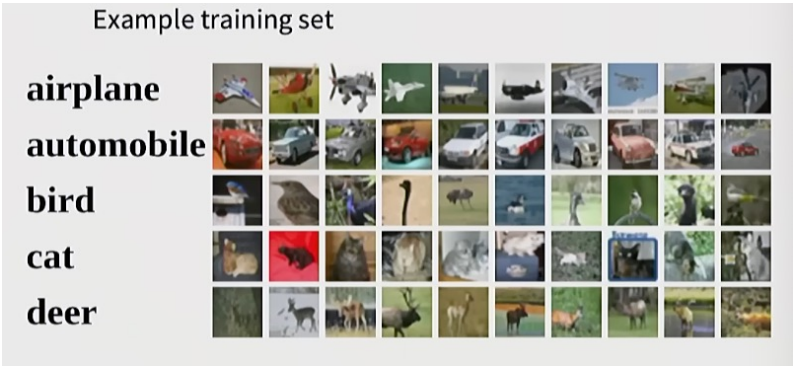
Basic Course – Introduction

- 특정 Task를 해결하기 위한 접근법

Algorithmic Approach



Data-Driven Approach



CVSP Lab
Computer Vision & Signal Processing

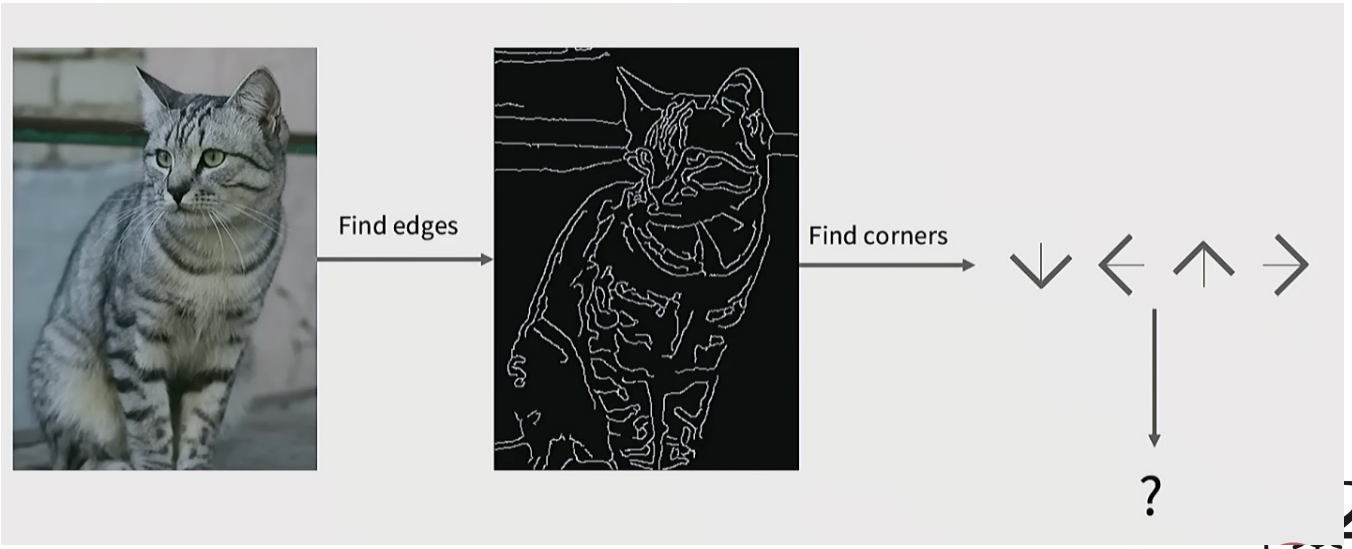
Basic Course – Introduction

- 특정 Task를 해결하기 위한 접근법

1. Algorithmic Approach

- 사람의 직관 / 수학적 모델을 이용하여 무엇이 feature인지 정의
  - Canny Edge Detector, Sobel Filter, Laplacian Filter …
- 정의된 규칙에 따라 추출된 feature를 이용하여 Task 해결

- 장점
  - Feature의 의미를 해석 가능
  - 연산 자체가 가볍고 빠름



CVSP Lab
Computer Vision & Signal Processing

Basic Course – Introduction

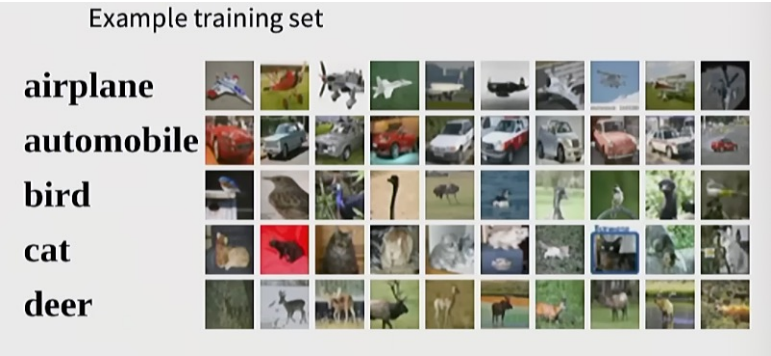
- 특정 Task를 해결하기 위한 접근법

1. Algorithmic Approach

- 단점
  - 복잡한 패턴 및 추상적 개념은 규칙으로 정의 힘듦
  - 데이터셋 내 각 데이터당 파라미터를 조정해줘야 함

2. Data-Driven Approach

- 사람이 정한 규칙이 아닌, 대량의 데이터를 통해 특징을 추출하는 방법과 판단하는 기준을 학습
- 장점
  - 데이터가 충분할 시, 사람이 정의한 규칙보다 정교하고 robust한 특징을 찾아냄
  - 각 데이터당 파라미터 조정할 필요 없음



Basic Course – K-Nearest Neighbor

- K-NN 알고리즘
  1. Train data와 Test data 간 거리를 계산 후 가장 가까운 K개의 이웃 선택
  2. K개의 이웃 중 가장 많이 등장하는 class를 test data의 class로 결정

- Distance Metric (pixel-wise)

  L1 distance

  - 식: $d_1(I_1, I_2) = \sum_P |I_1^P - I_2^P|$

    ∴ 모든 feature(pixel) 간 거리를 같은 비율로 반영

  L2 distance

  - 식: $d_2(I_1, I_2) = \sqrt{\sum_P (I_1^P - I_2^P)^2}$

    ∴ feature(pixel) 간 거리가 멀수록 더욱 많이 반영

CVSP Lab
Computer Vision & Signal Processing

Basic Course – Linear Model

- K-NN

  단점

  - 픽셀 단위의 조금의 차이만으로도 큰 오차 발생
  - 모든 데이터셋을 테스트 때 사용

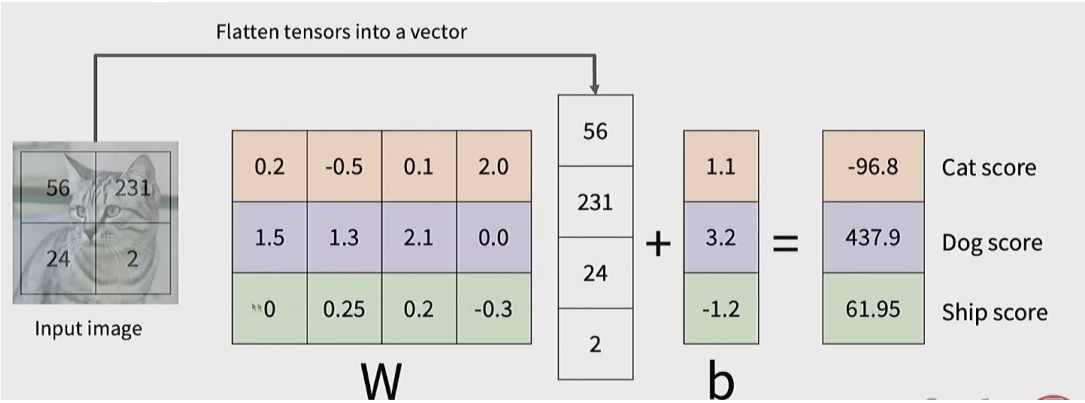
    ⁘ 메모리 효율이 떨어짐, 계산량이 훈련 시보다 더 큼

- Linear Model

  가중치 파라미터를 도입

  - **적절한 가중치 행렬 W**만 있다면 class에 대한 점수를 매길 수 있음

    ⁘ K-NN의 Test시 모든 데이터를 보유하고 있어야 하는 문제점 해결



CVSP Lab
Computer Vision & Signal Processing

Basic Course – Linear Model

- Loss Function

  적절한 가중치 행렬 W를 찾기 위한 **기준**

  - 특정 weight matrix와 데이터셋이 존재할 때, task를 해결하기 위한 최적의 W를 찾기 위함

  e.g. Classification: (Softmax) Cross-Entropy Loss

  - $L_i = -log(\frac{e^{sy_i}}{\sum_j e^{sj}})$

- Gradient Descent

  적절한 가중치 행렬 W를 찾기 위한 **방법**

  - Loss Function을 최소화하는 방향으로 θ 업데이트
  - $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$

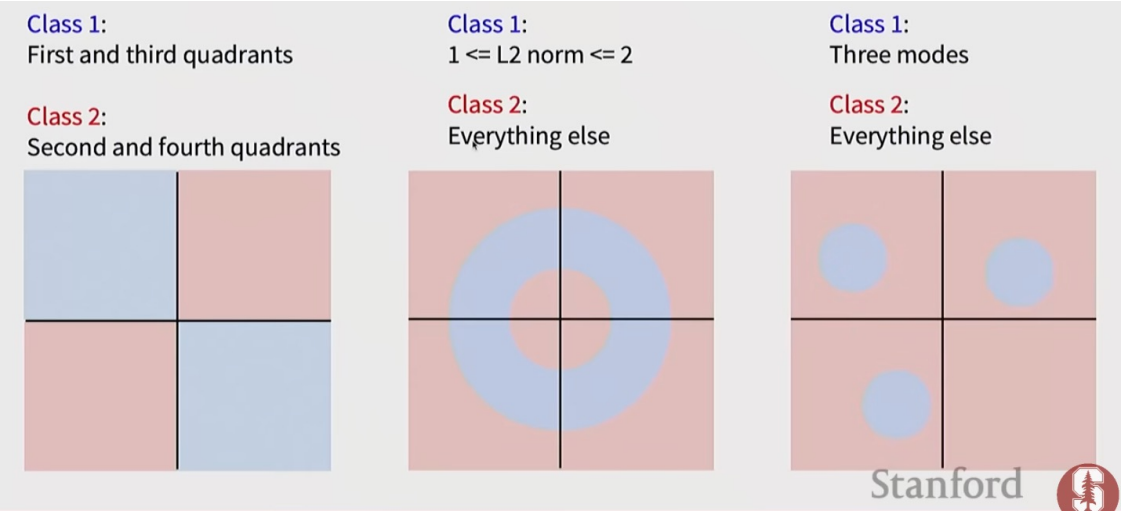
CVSP Lab
Computer Vision & Signal Processing

Basic Course – MLP

- Linear Model

  단점

  - 오직 선형 결정 경계(Decision Boundary)만 학습 가능
    → 선형 분리 불가능 데이터에는 성능 하락



- Multi Layer Perceptron (MLP)

  활성화 함수(비선형)의 도입

  - 복잡한 결정 경계 학습 가능

Universal Approximation Theorem
→ 충분한 뉴런을 가진 단일 은닉층 MLP는 어떤 연속 함수도 근사 가능

CVSP Lab
Computer Vision & Signal Processing

Basic Course – MLP

- 구조

Input / Hidden / Output 으로 이루어진 계층 구조

- 입력층을 제외한 모든 뉴런에 다음과 같은 연산이 수행됨

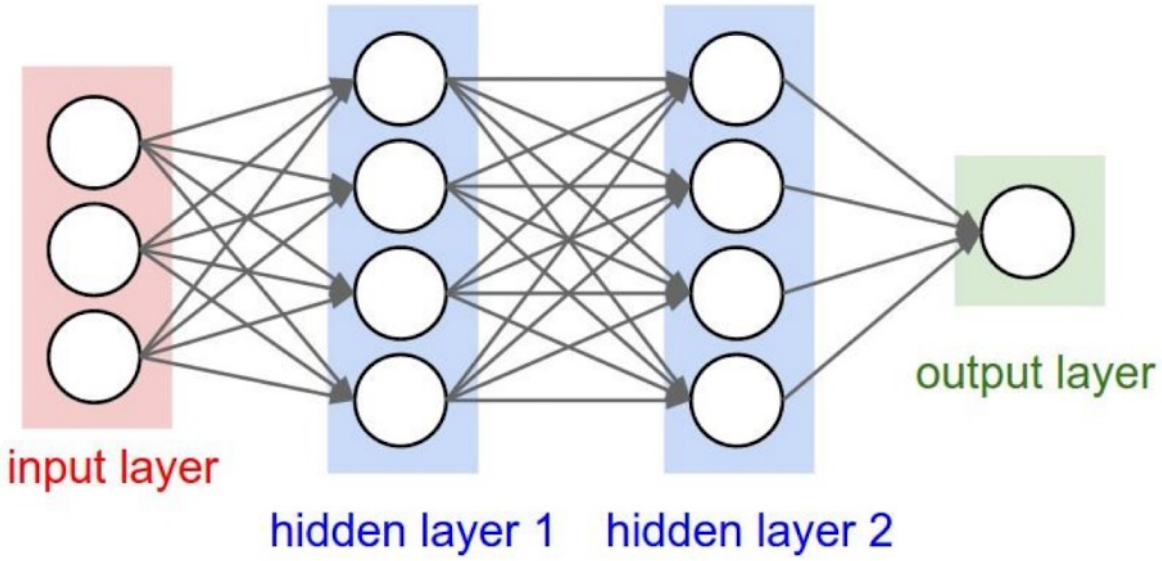
$z = \sum_{\{i=1\}}^{n} w_i x_i + b = W^T X + b,\ \boldsymbol{a} = \boldsymbol{\sigma}(\boldsymbol{z})$

✓X: 입력 행렬

✓W: 가중치 행렬

✓b: bias

✓σ: 활성화 함수



CVSP Lab
Computer Vision & Signal Processing

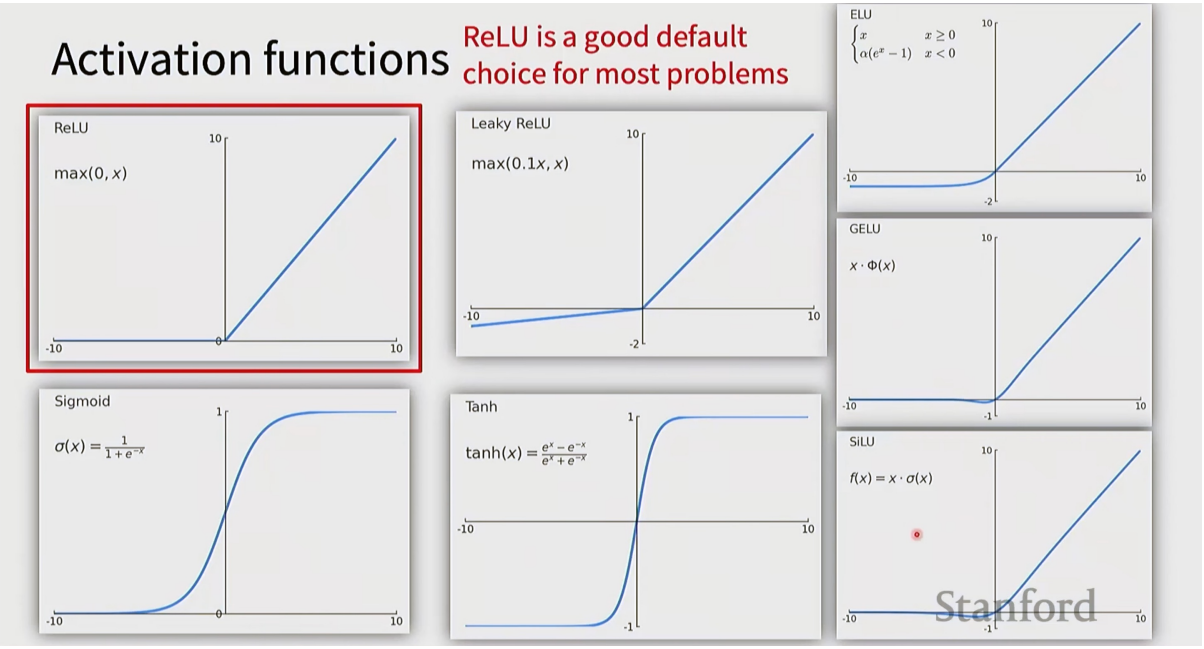
Basic Course – MLP

- **Activation Function($\sigma(x)$, 비선형 함수)** 도입

  비선형성을 추가 → 여러 Linear Layer를 쌓기 가능

  - 계층적인 구조 형성 → 각 층의 특징들을 조합하여 **새로운 특징 표현 가능**

  Sigmoid나 Tanh의 Gradient Vanishing 문제로 인해 주로 ReLU 계열 사용



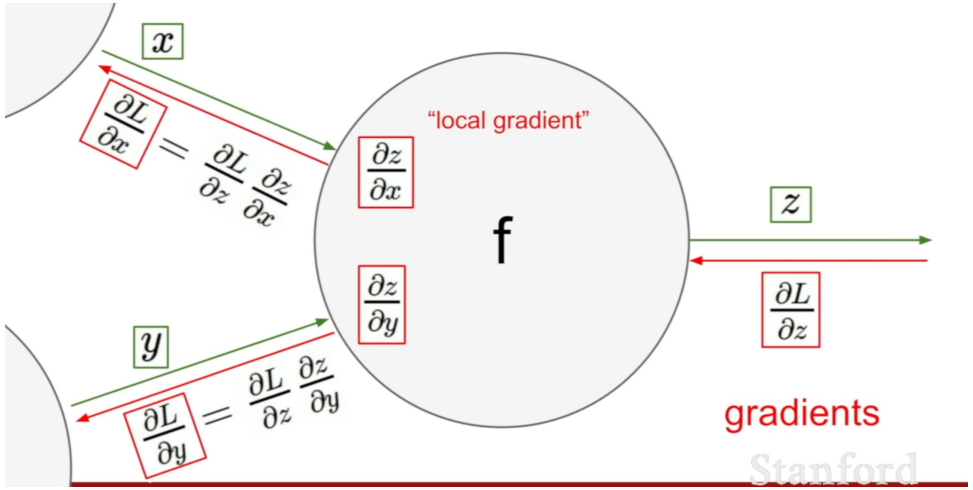
CVSP Lab
Computer Vision & Signal Processing

Basic Course – MLP

- Backpropagation

MLP

- 층이 깊어짐 → 사용하는 파라미터가 증가
  - 사용하는 모든 파라미터에 대해 각각 Gradient를 계산 → 계산 복잡도 폭발
    - ✓실질적으로 학습 불가능
- **Chain rule을 이용한 Backpropagation 수행**
  - 특정 파라미터의 Gradient 계산을 이전에 계산했던 Gradient를 포함한 식으로 표현
    - ✓계산 복잡도 대폭 낮춤 → 실질적 학습 가능



CVSP Lab
Computer Vision & Signal Processing

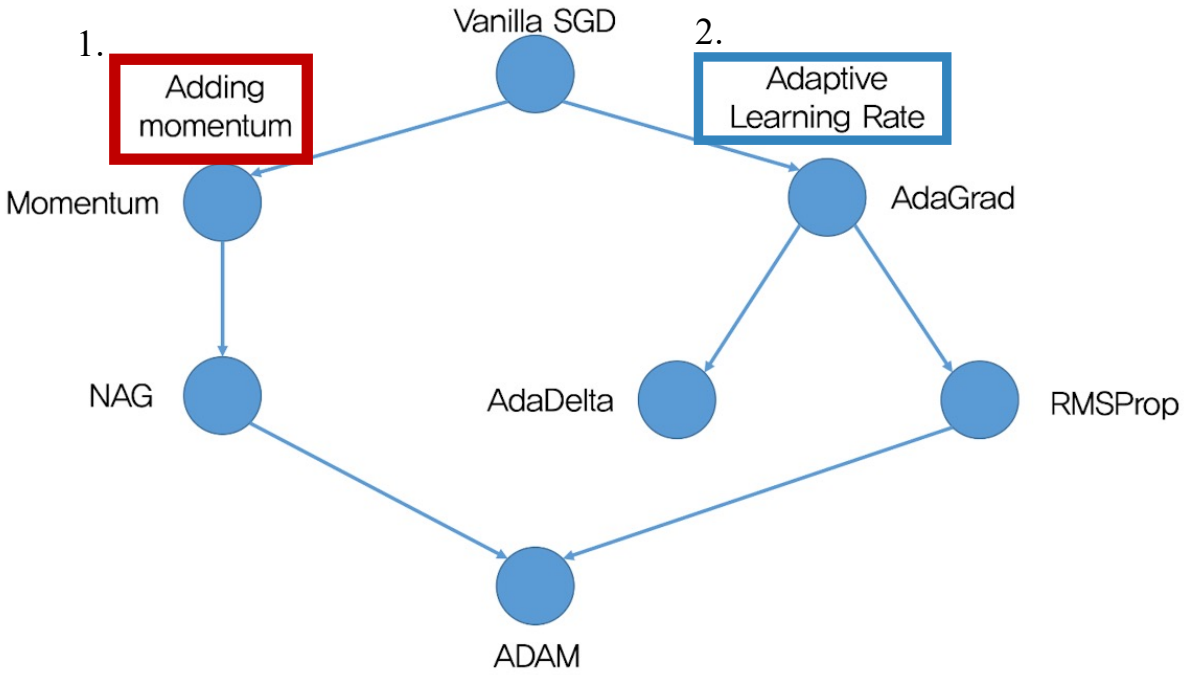
Basic Course – MLP

*: Appendix에서 추가 설명

- Optimization*

  가중치를 업데이트하는 방법

Overview



CVSP Lab
Computer Vision & Signal Processing

Basic Course – MLP

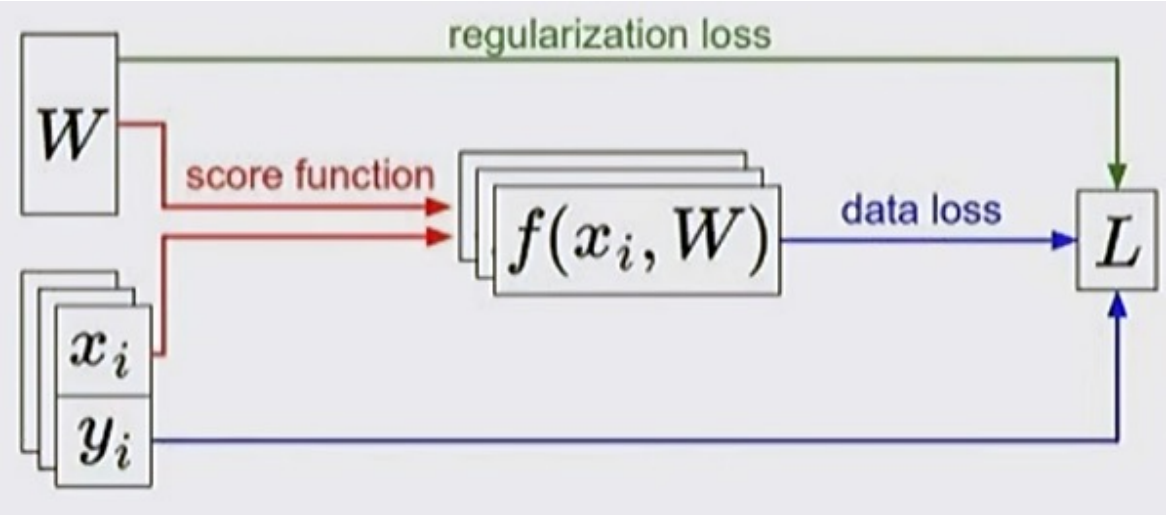
- Regularization

  Loss로만 가중치 업데이트 → 학습 데이터에 대해 Overfitting

  이를 방지하기 위해 Regularization loss term 추가

  - 모델이 일반화가 되도록 유도
  - Data Loss term과 trade-off 관계

$$L(W) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i)}_{\text{Data Loss}} + \underbrace{\lambda R(W)}_{\text{Regularization Loss}}$$



CVSP Lab
Computer Vision & Signal Processing

Basic Course – CNN

- Multi-Layer Perceptron (MLP)

  단점

  - 모델 내 모든 연결 지점마다 고유한 가중치를 부여해야 함
    - 파라미터 수 폭발
  - Input을 넣기 위해 flatten 작업 수행
    - 이미지의 공간적인 구조를 무시

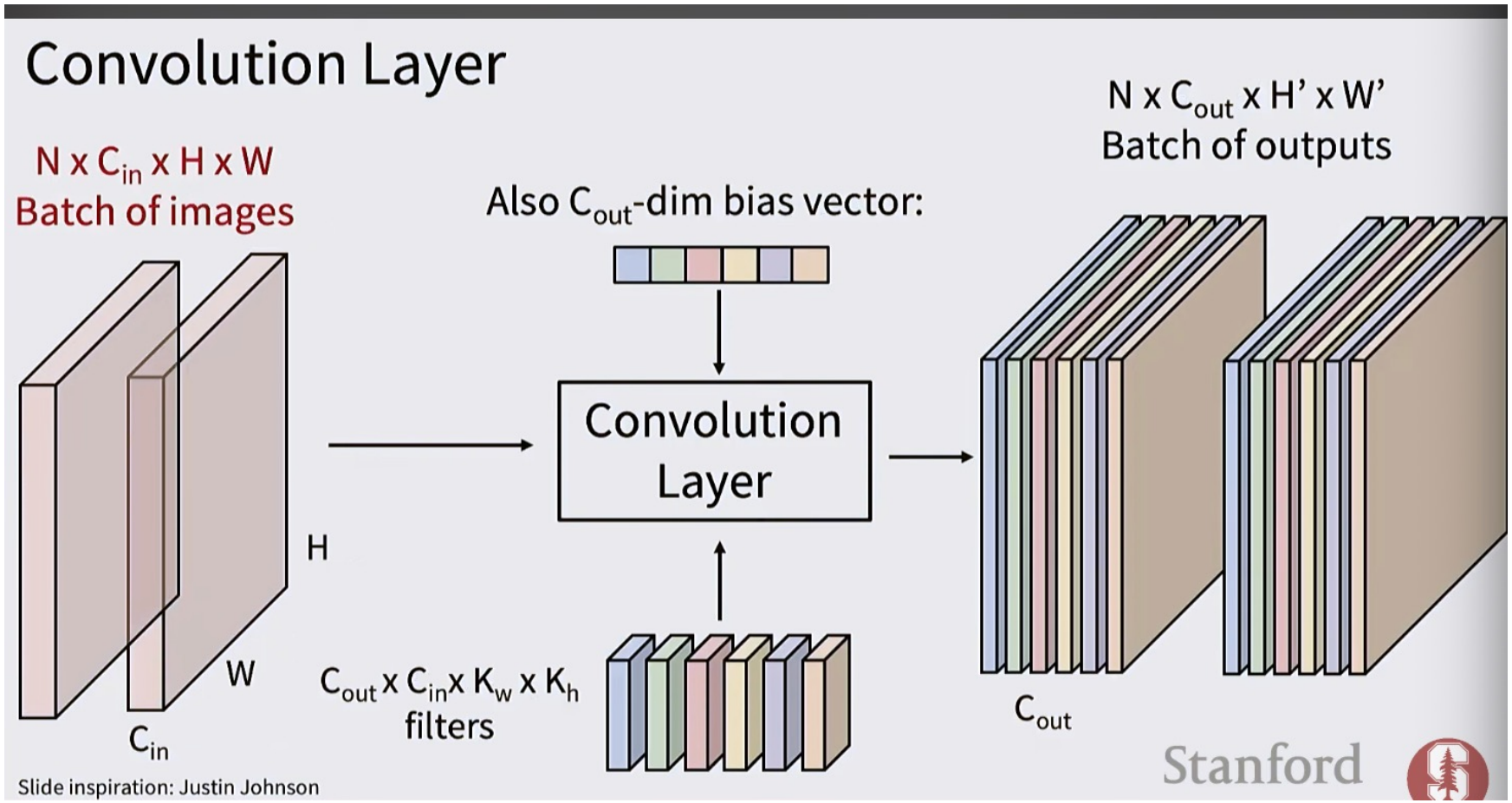
- Convolution Neural Network (CNN)

  MLP의 단점을 해결

  - 각 layer 내에서 **Filter**들을 이용하여 학습
    - Filter라는 파라미터를 공유 → MLP와 비교했을 때, 상대적으로 **파라미터 수 감소**
  - 입력: **이미지** 형태
    - 모델의 입력으로 **공간적인 구조가 유지된 이미지** 입력
      - ✓Filter를 이용하여 Patch 단위로 공간적인 구조를 학습

Basic Course – CNN

- 구조



CVSP Lab
Computer Vision & Signal Processing

Basic Course – CNN

- 한 필터에서의 계산 과정

  연산 @ : $\sum_{c,i,j} P[c,i,j] \cdot F[c,i,j] + b$

  - b : bias (필터 당 동일한 값)
  - 필터를 움직여가며 지속적인 연산 수행 → 하나의 feature map 생성
  - 이후 Backpropagation으로 Filter와 bias가 업데이트 → **이미지 내의 특정 패턴을 학습**

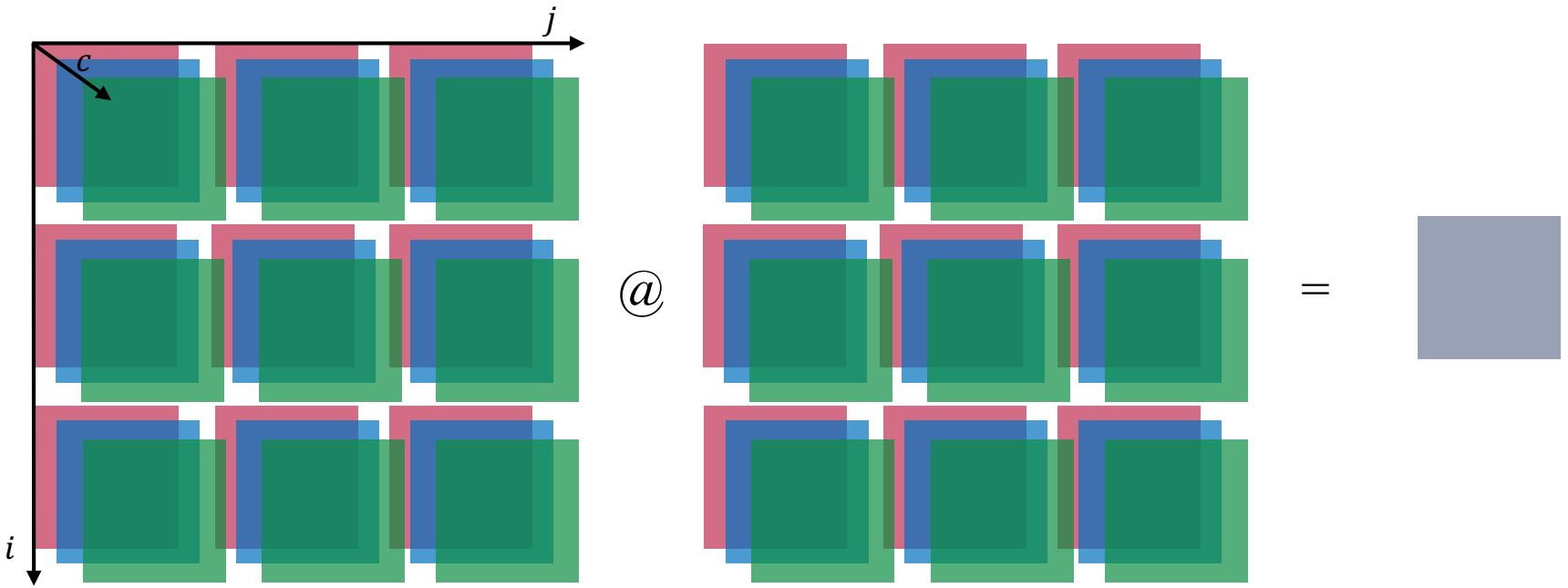


Image Patch
(P)

Filter
(F)

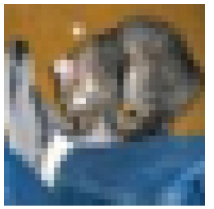
CVSP Lab
Computer Vision & Signal Processing

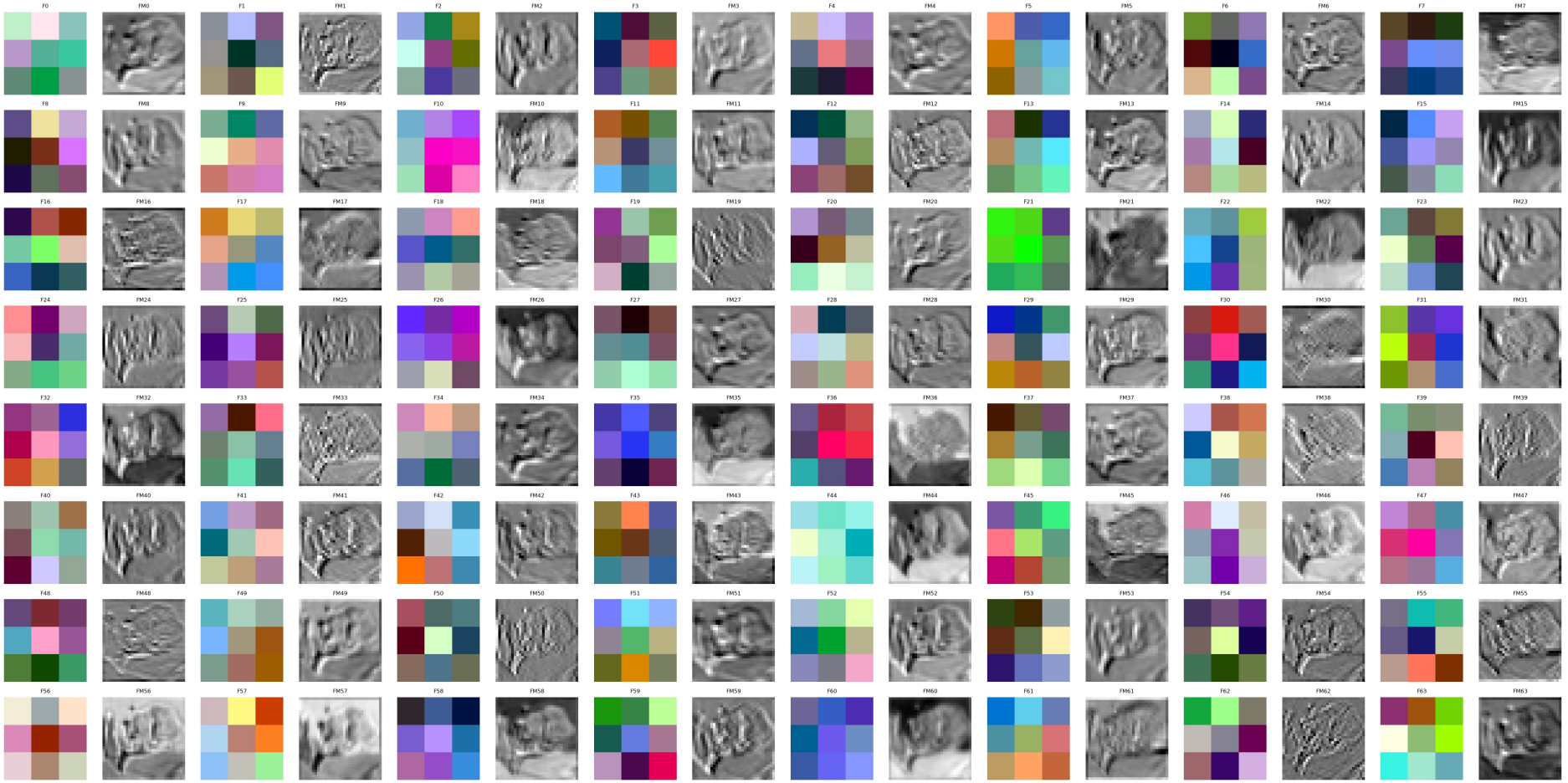
Basic Course – CNN

- Example: VGG16 – first layer (filter shape: 3x3x3 num: 64)

  Filter가 이미지 내 각기 다른 패턴들을 학습

Input (CIFA10):





CVSP Lab
Computer Vision & Signal Processing

Basic Course – CNN

- Receptive field

  특정 layer의 픽셀 하나에 영향을 주는 Input의 픽셀 범위

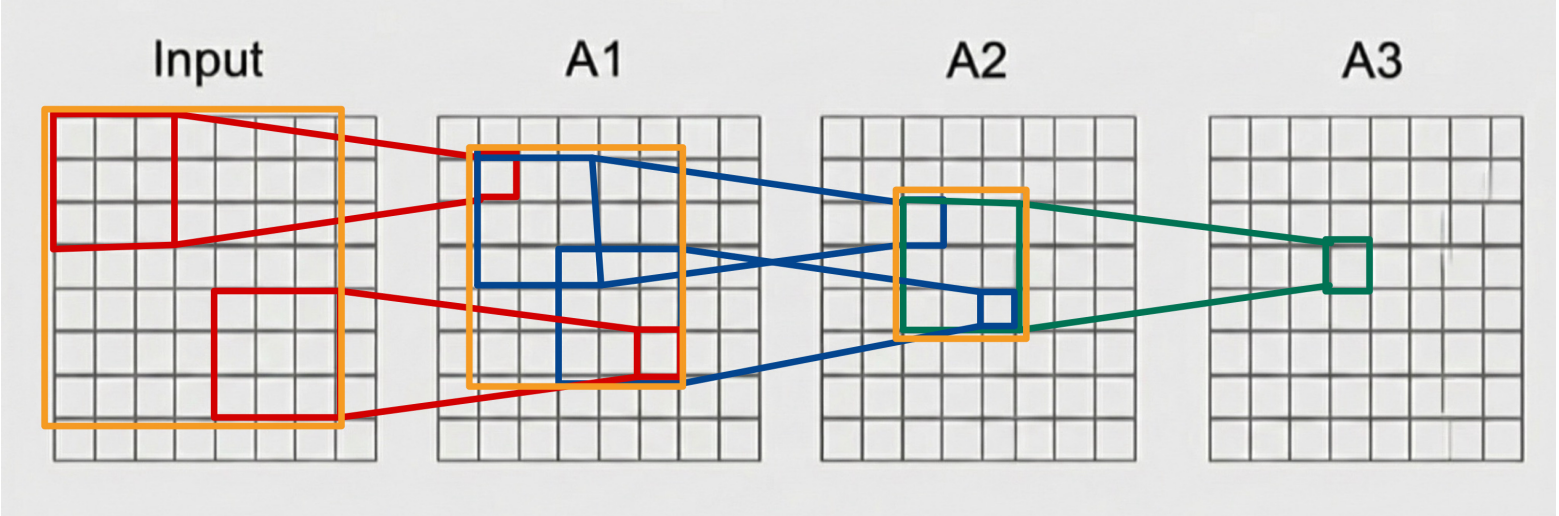
  - 깊은 Layer일수록, Receptive Field의 범위가 넓음

    ⁘ 얕은 층의 필터들 → Input의 local한 특징 학습

    ⁘ 깊은 층의 필터들 → Input의 Global한 특징 학습

  Receptive field를 넓히는 방법

  - Layer 적층 / 큰 Filter 사용 / Stride 키우기 / Pooling



CVSP Lab
Computer Vision & Signal Processing

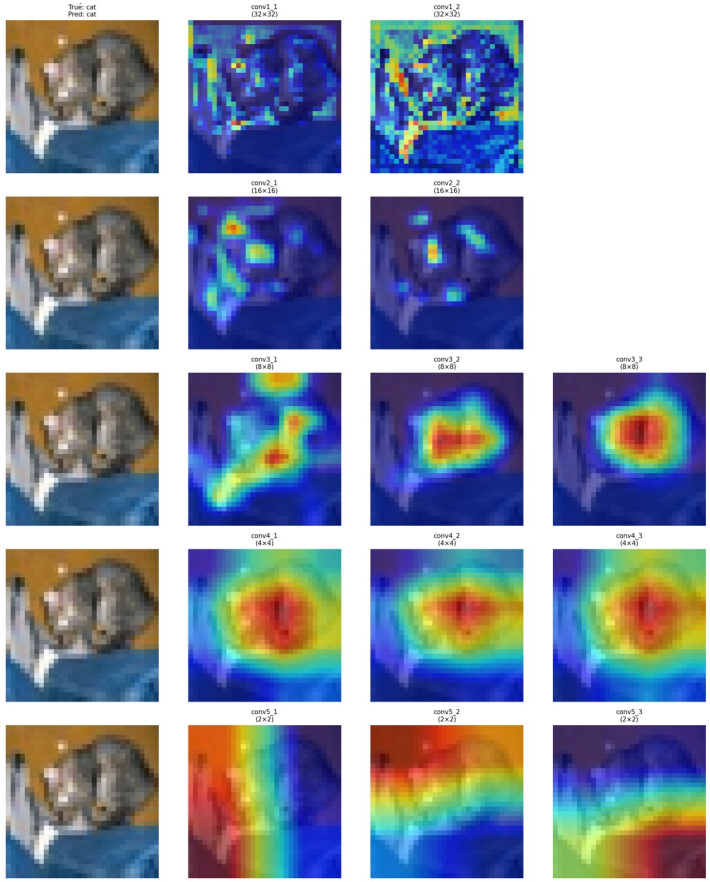
Basic Course – CNN

4) Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.
*: Appendix에서 추가 설명

- Example: VGG16의 각 conv layer에 Grad-CAM[4], * 적용

  Receptive field의 범위가 layer의 깊이에 따라 점점 늘어남을 확인

  - 얕은 layer: local 특징 (edge)
  - 깊은 layer: global 특징 (object)



Shallow

Deep

CVSP Lab
Computer Vision & Signal Processing

Basic Course – CNN

- Architecture: AlexNet & VGGNet

  AlexNet의 비효율적인 filter 사용을 VGGNet에서 개선

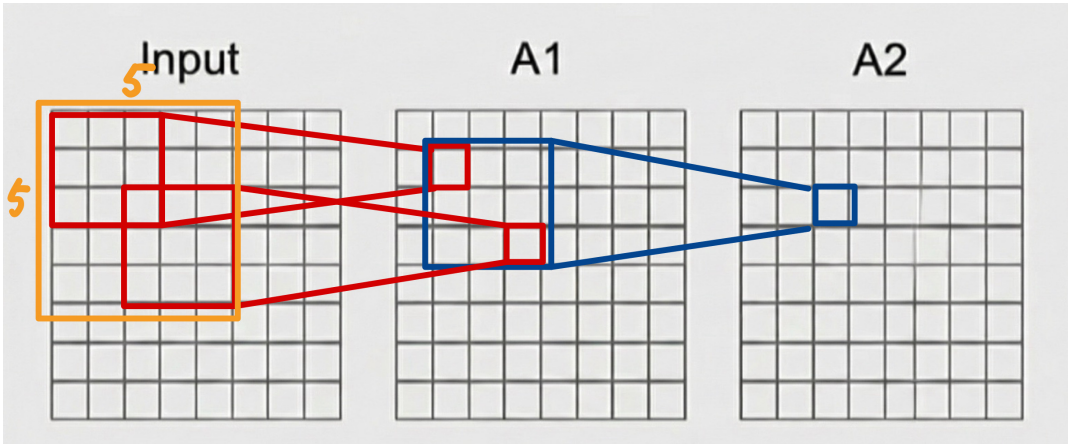
  - Ex. 5x5 filter x1 → 3x3 filter x2 : receptive field 유지, 파라미터 수 감소

    Params: 5x5 = 25 > 3x3x2 = 18

  층이 깊어지면 Gradient vanishing 문제 발생



AlexNet & VGGNet Architecture



Receptive Field: 5x5 1개 vs. 3x3 2개

CVSP Lab
Computer Vision & Signal Processing

Basic Course – CNN

5) Li, Hao, et al. "Visualizing the loss landscape of neural nets." *Advances in neural information processing systems* 31 (2018).

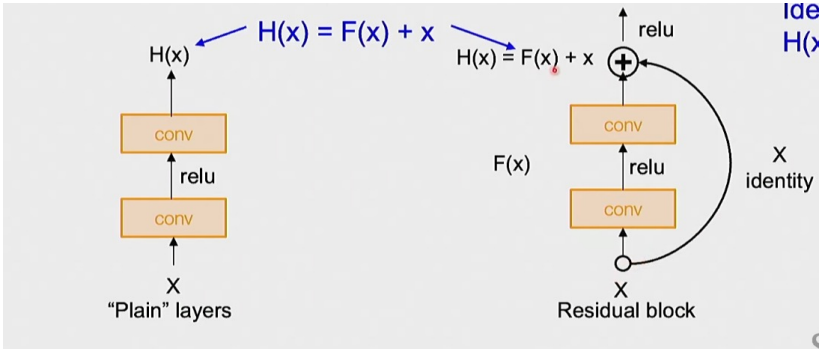
- Architecture: ResNet

기존 CNN 모델들

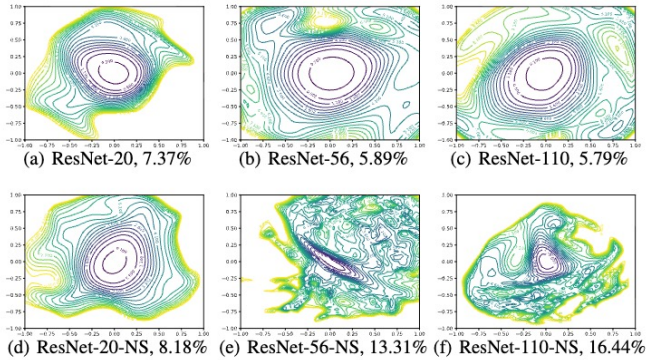
- 층이 깊어질수록 2가지 단점 발생
  - Loss Landscape 복잡화 → optimization 어려움
  - Gradient Vanishing 발생

ResNet

- Residual connection(Skip connection)을 추가함으로써 모델의 목표를 변경
  - scratch에서부터 목적 값을 예측 → Input에서 목적 값과의 변화를 예측
    - ✓장점 1) Loss landscape를 간단하게 만듦 [5)]
    - ✓장점 2) 미분 시, Gradient가 최소 1이므로 Gradient Vanishing을 완화



Plain layers vs. Residual Block



Skip connection: O

Skip connection: X

Figure 5: 2D visualization of the loss surface of ResNet and ResNet-noshort with different depth.

CVSP Lab
Computer Vision & Signal Processing

Vision Transformer (ViT)

CVSP Lab
Computer Vision & Signal Processing

ViT – Introduction

- 배경

  2017년, Transformer 발표 → NLP 분야에서 큰 성공

  Transformer를 **Vision Task에도 접목**하고자 하는 시도 존재

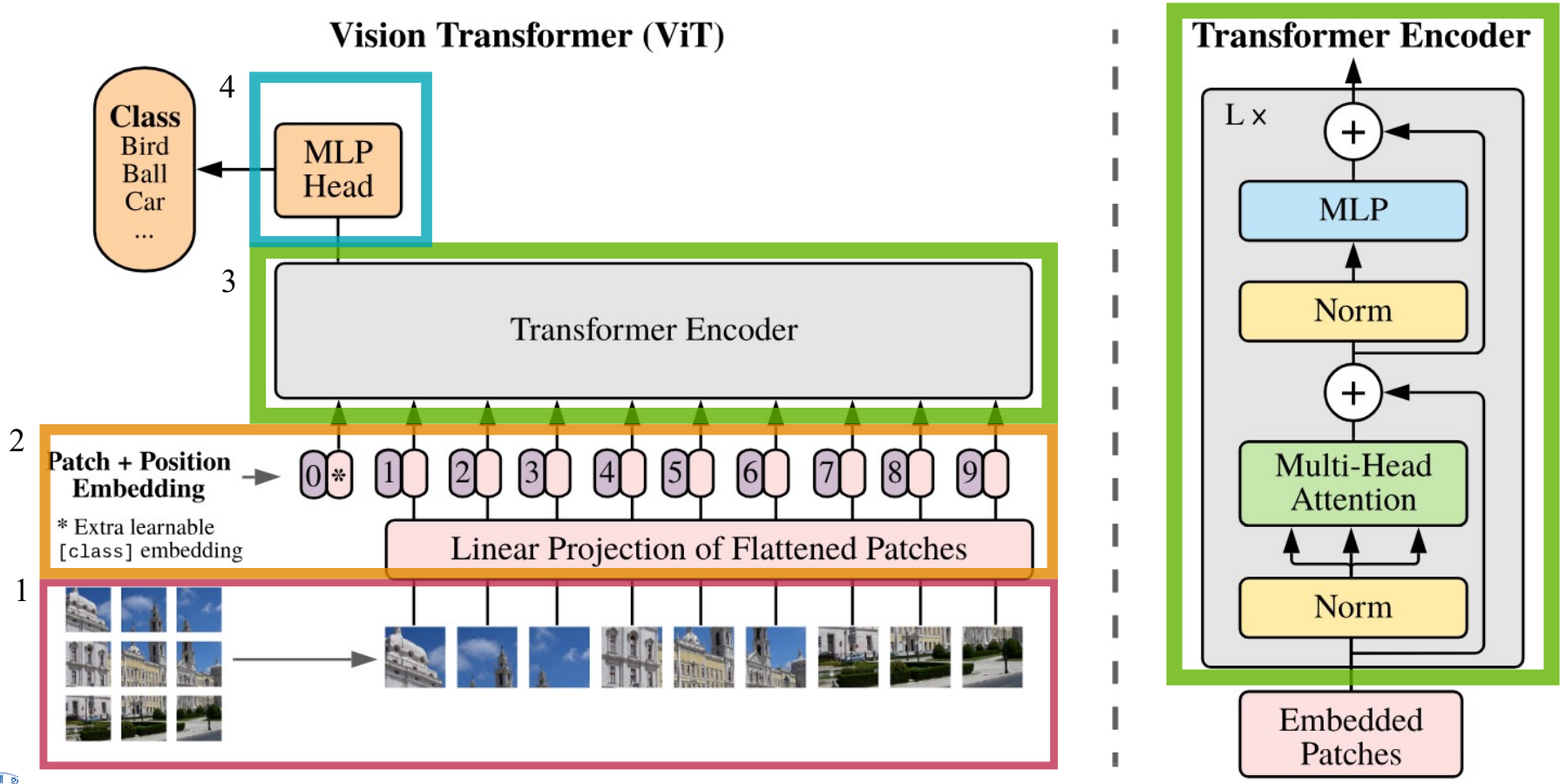
  ViT는 Transformer 입력 형태로 이미지를 변형 → Transformer 구조 사용 가능

  - Patch 단위 Attention
  - **기존 Attention 메커니즘 유지** → NLP 분야에서 확인한 장점들을 그대로 취함

CVSP Lab
Computer Vision & Signal Processing

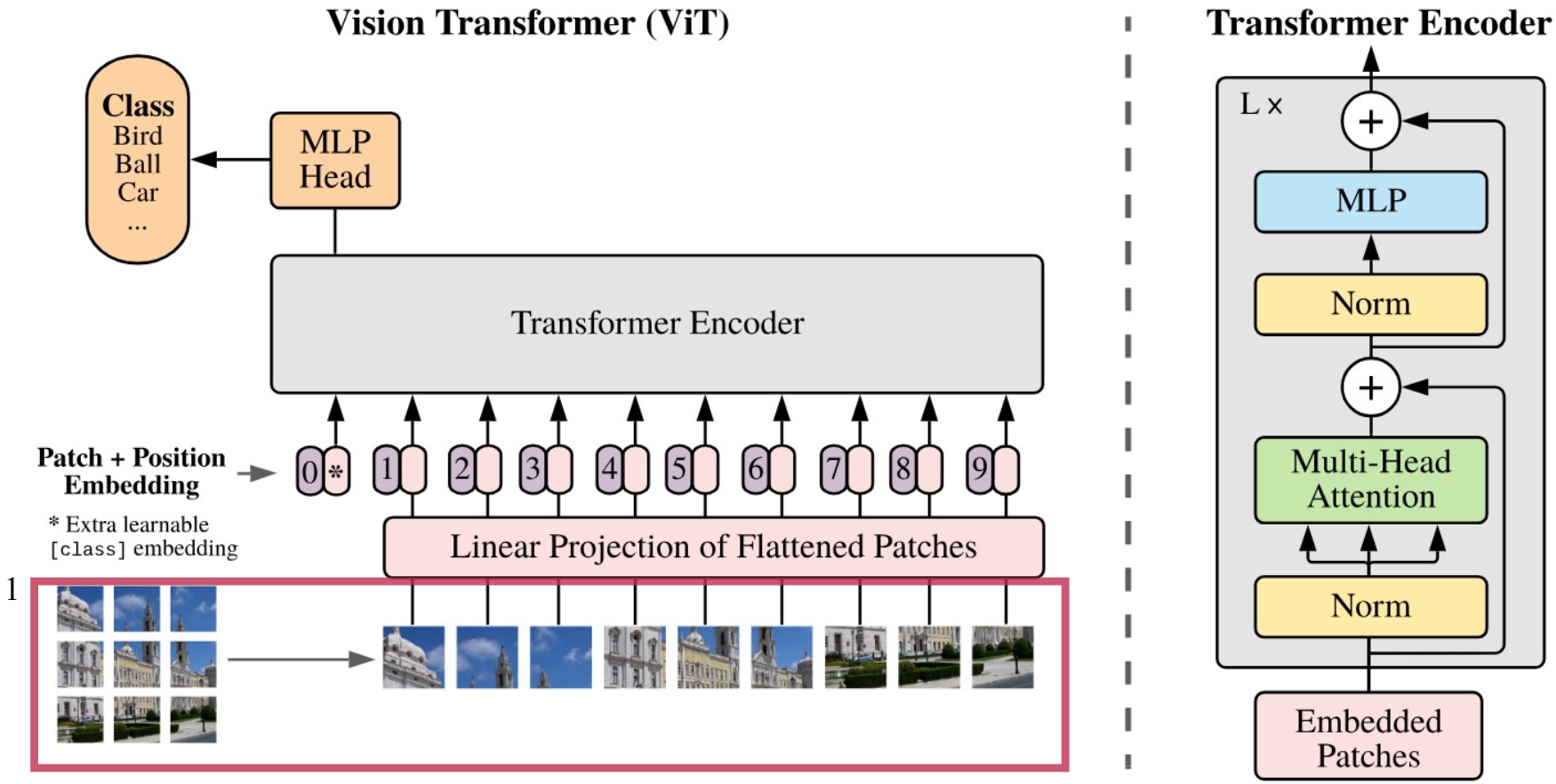
ViT – Architecture & Experiment

- 구조



ViT – Architecture & Experiment

- 1. Divide into patch → Flatten



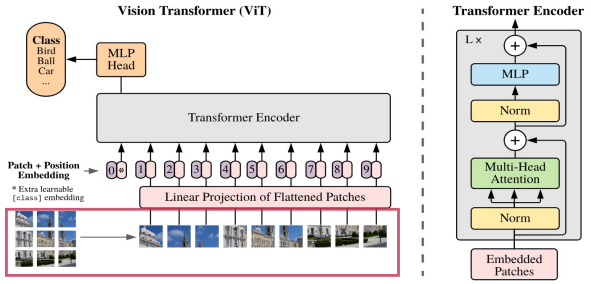
CVSP Lab
Computer Vision & Signal Processing

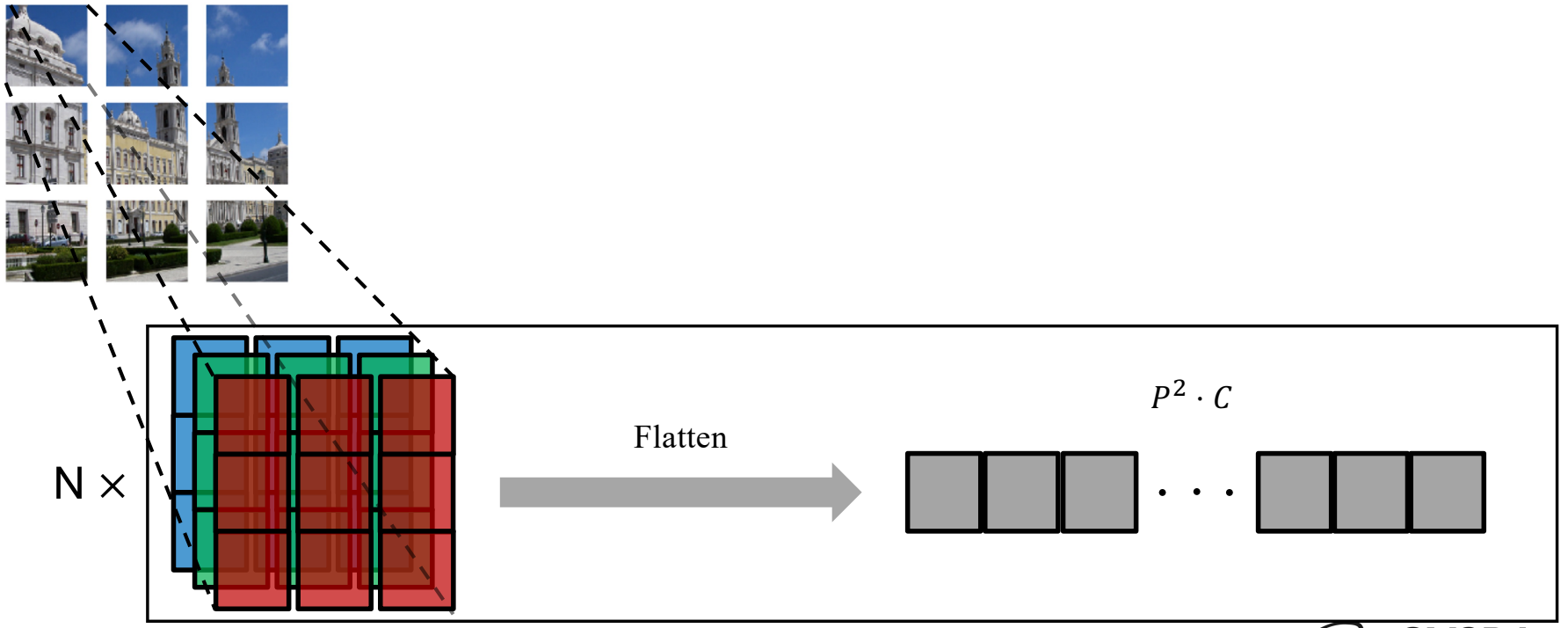
ViT – Architecture & Experiment

- 1. Divide into patch → Flatten

$$X \in R^{H \times W \times C} \rightarrow X \in R^{N \times (P^2 \cdot C)}$$

  - (H, W): 원본 이미지의 해상도 / (P, P): 이미지 Patch 해상도
  - C: Channel 수
  - N: Patch 수 (HW/P^2)

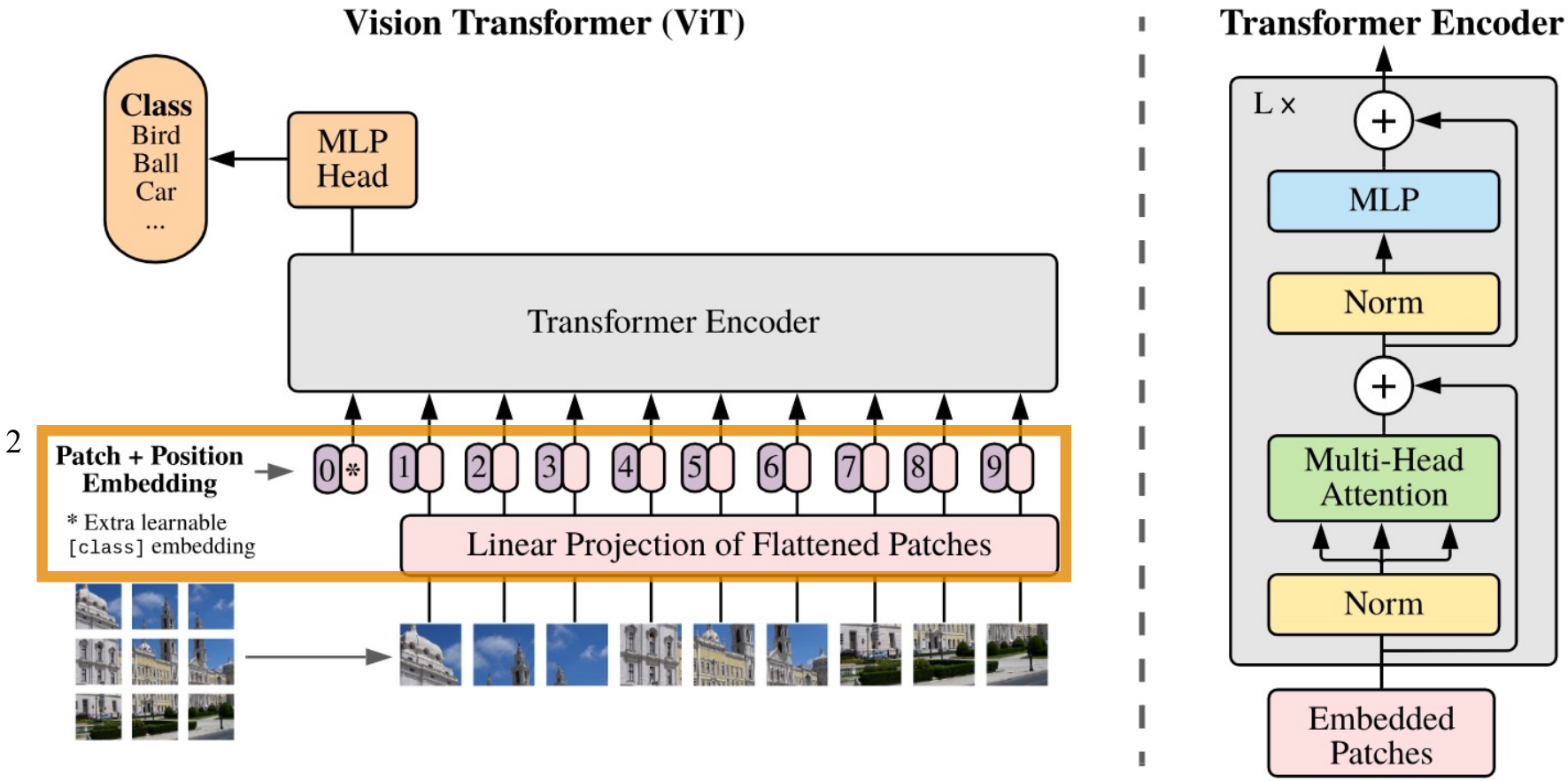




CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

- 2. Patch Embedding & Position Embedding



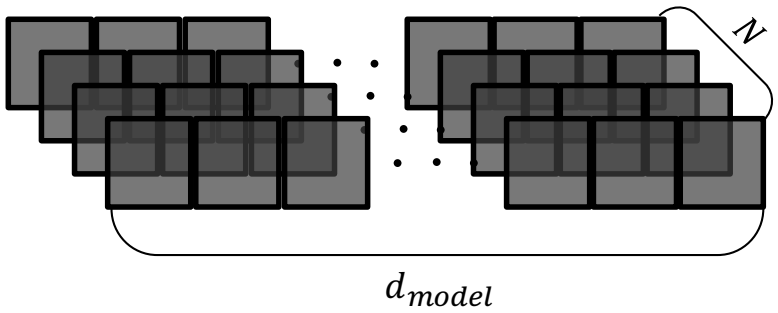
CVSP Lab
Computer Vision & Signal Processing

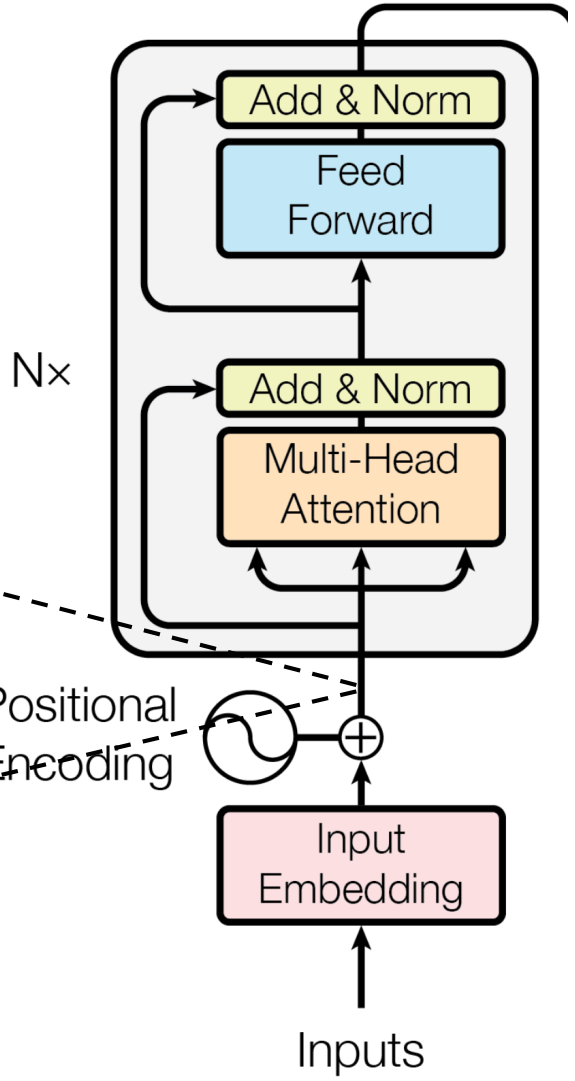
ViT – Architecture & Experiment

- Background: Transformer's Input shape

  $N \times d_{model}$

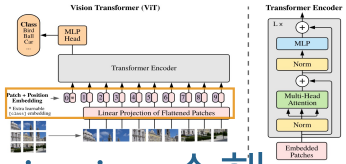
  - N : Token의 수
    - Token : 특정 기준으로 입력을 분할한 결과
  - d_{model}: 각 토큰의 임베딩 벡터 차원





CVSP Lab
Computer Vision & Signal Processing

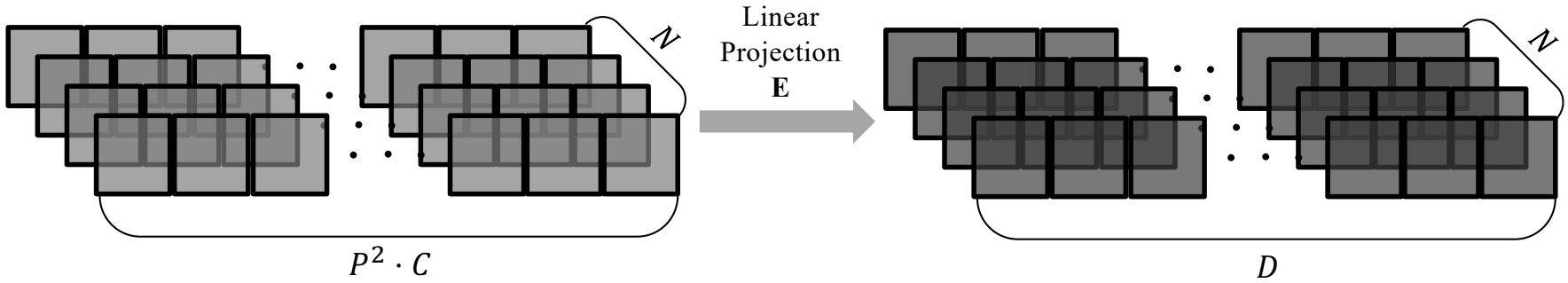
ViT – Architecture & Experiment



- 2.1 Patch Embedding

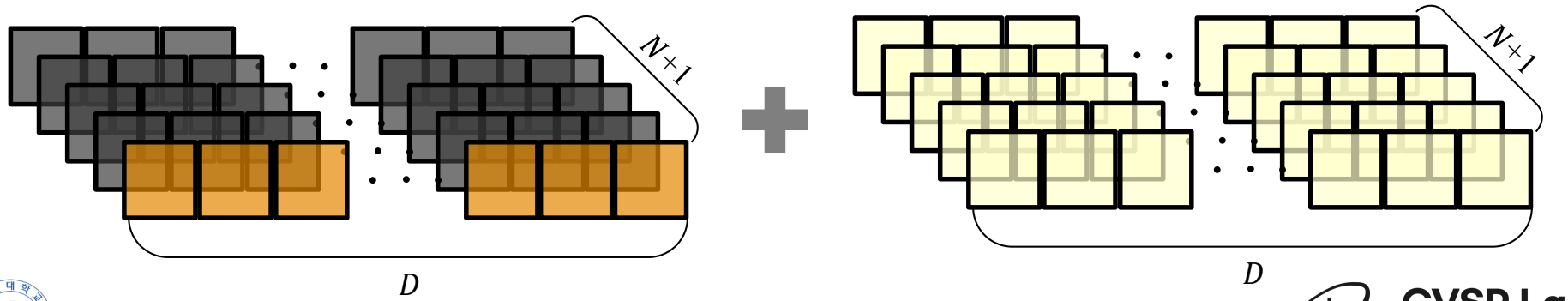
  각 Patch에 대해 Transformer의 입력 차원 D(=d_{model})로 Linear Projection 수행

  - Linear Projection: $E \in R^{(P^2 \cdot C) \times D}$



- 2.2 Add "Class" token & Position Embedding

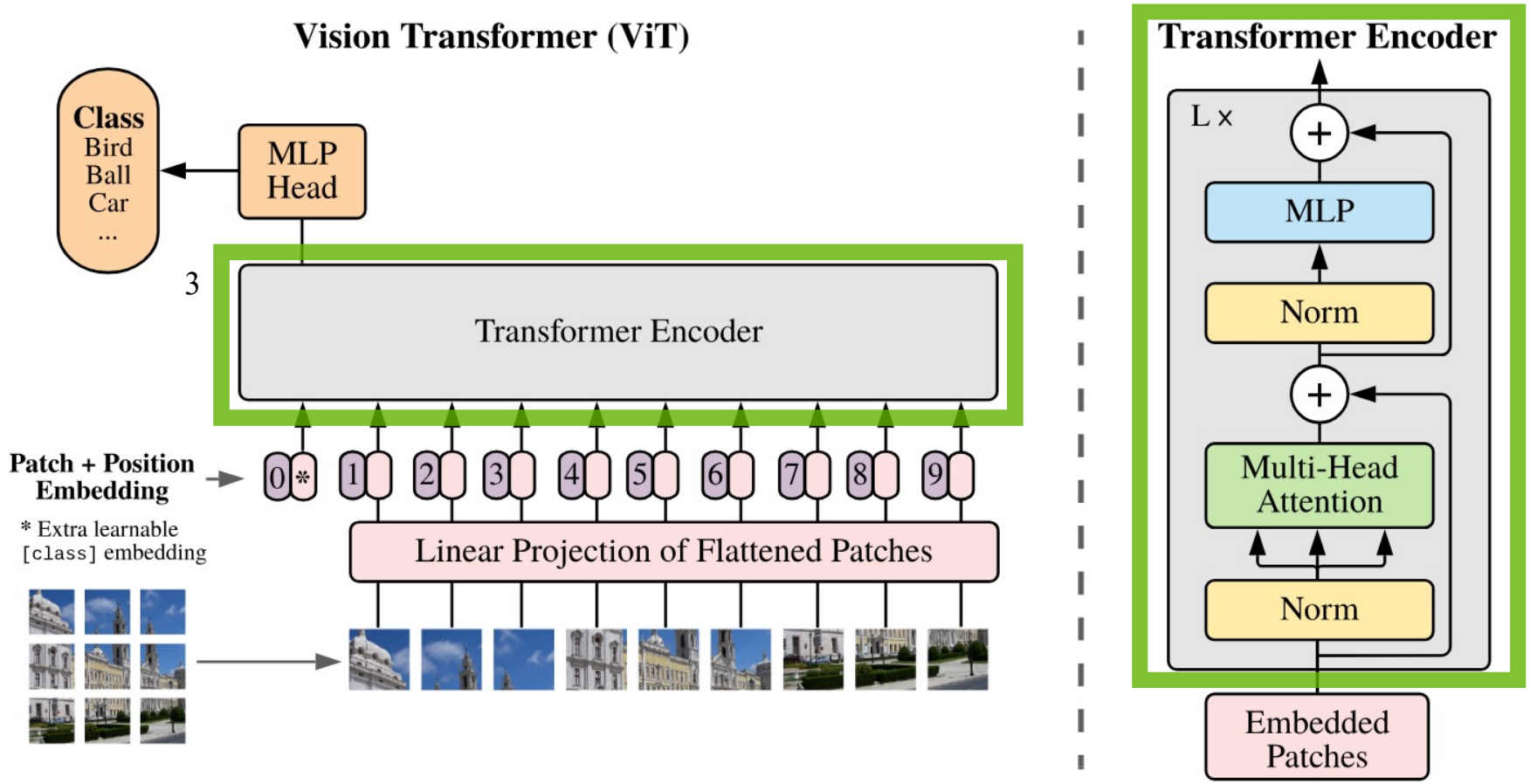
  Patch Embedding 결과에 $1 \times D$ 의 Class token을 concatenate 후,
  동일한 차원의 Position Embedding 행렬을 element-wise sum 수행



CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

- 3. Transformer Encoder



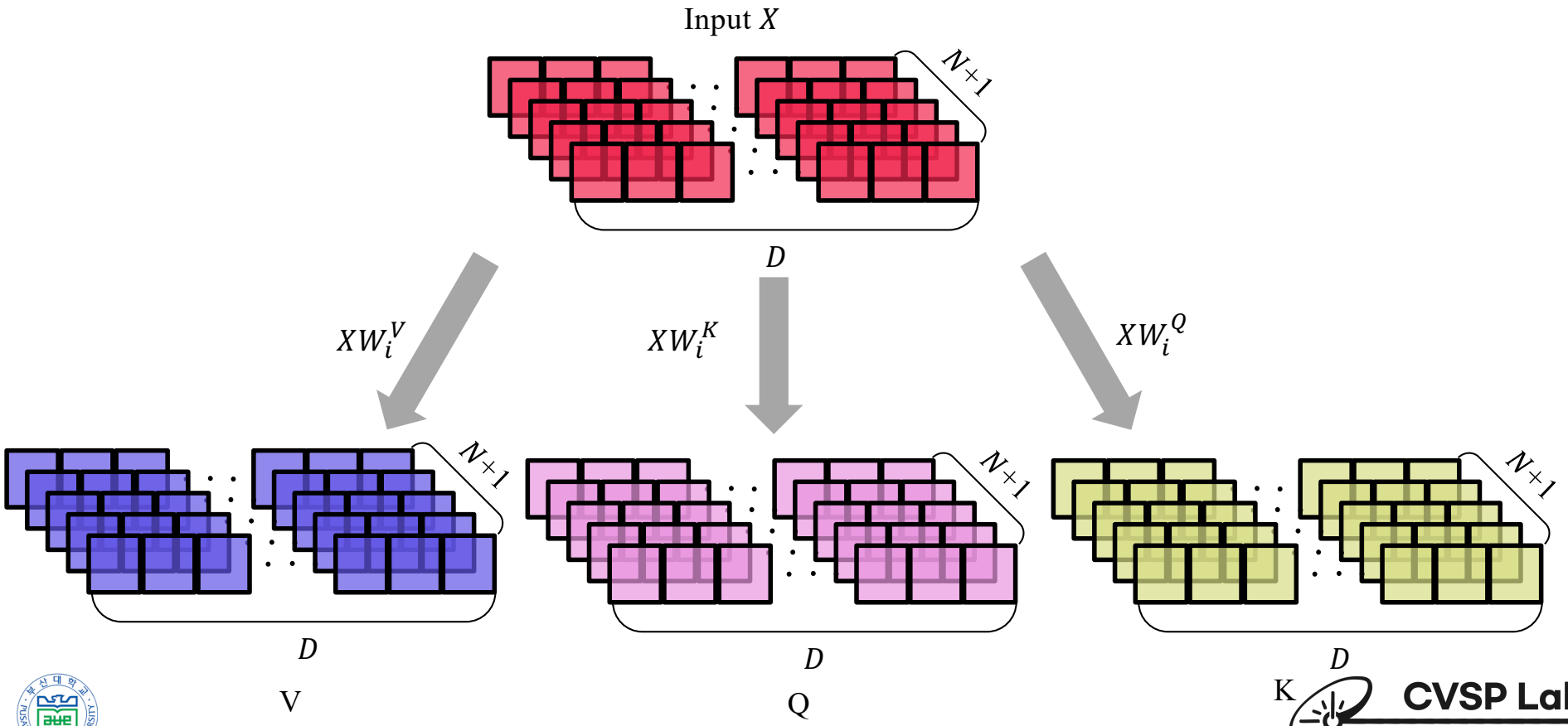
CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

- 3.1 Single Head Self-Attention – Linear Projection

  $X \in R^{(N+1) \times D}$

  - $W_i^V, W_i^k, W_i^Q \in R^{D \times D}$

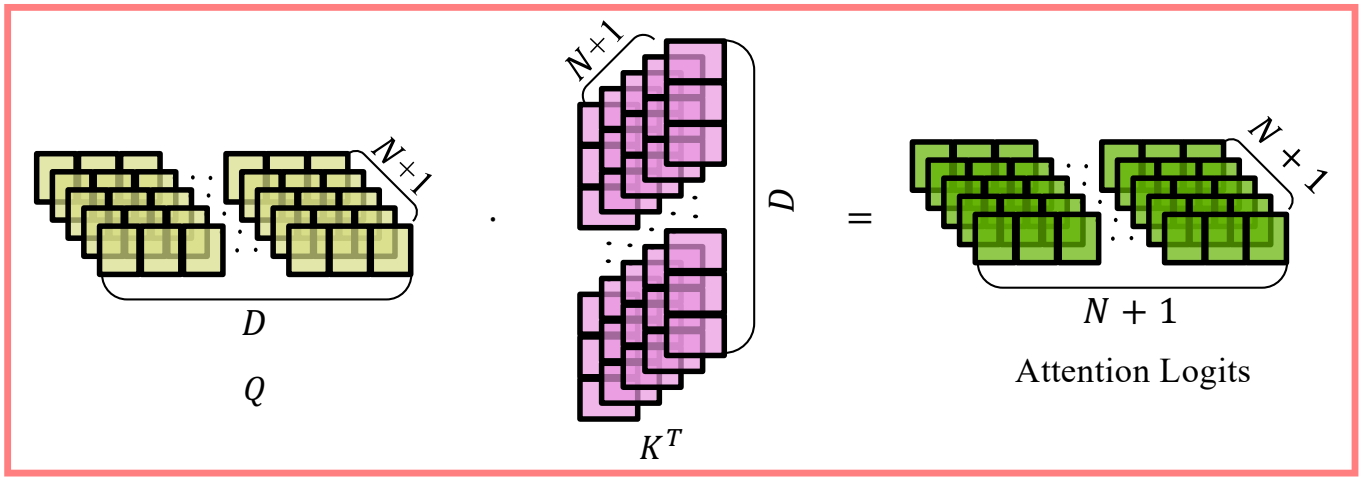


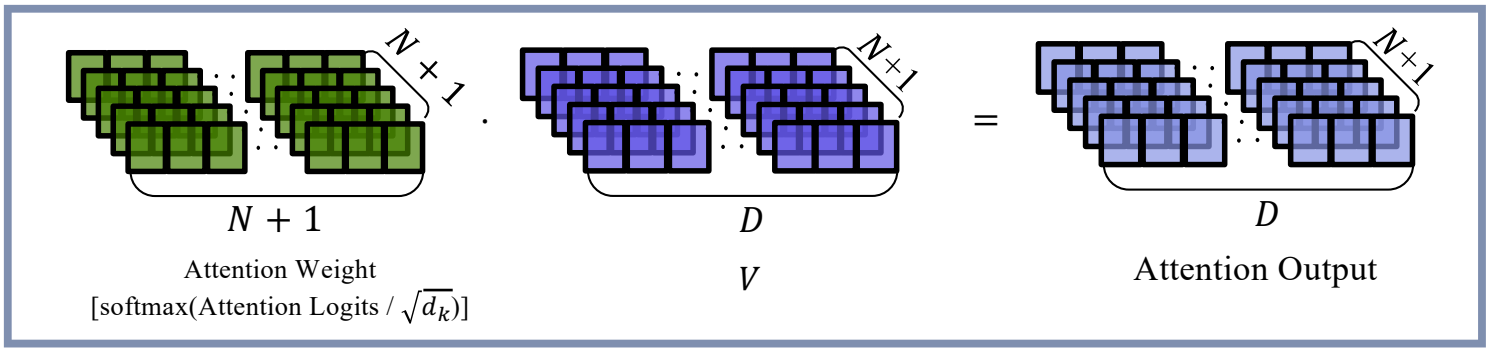
CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

- 3.2 Single Head Self-Attention – Scaled Dot-Product Attention

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

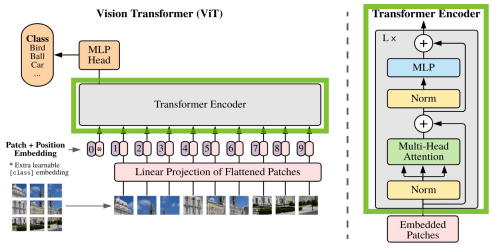


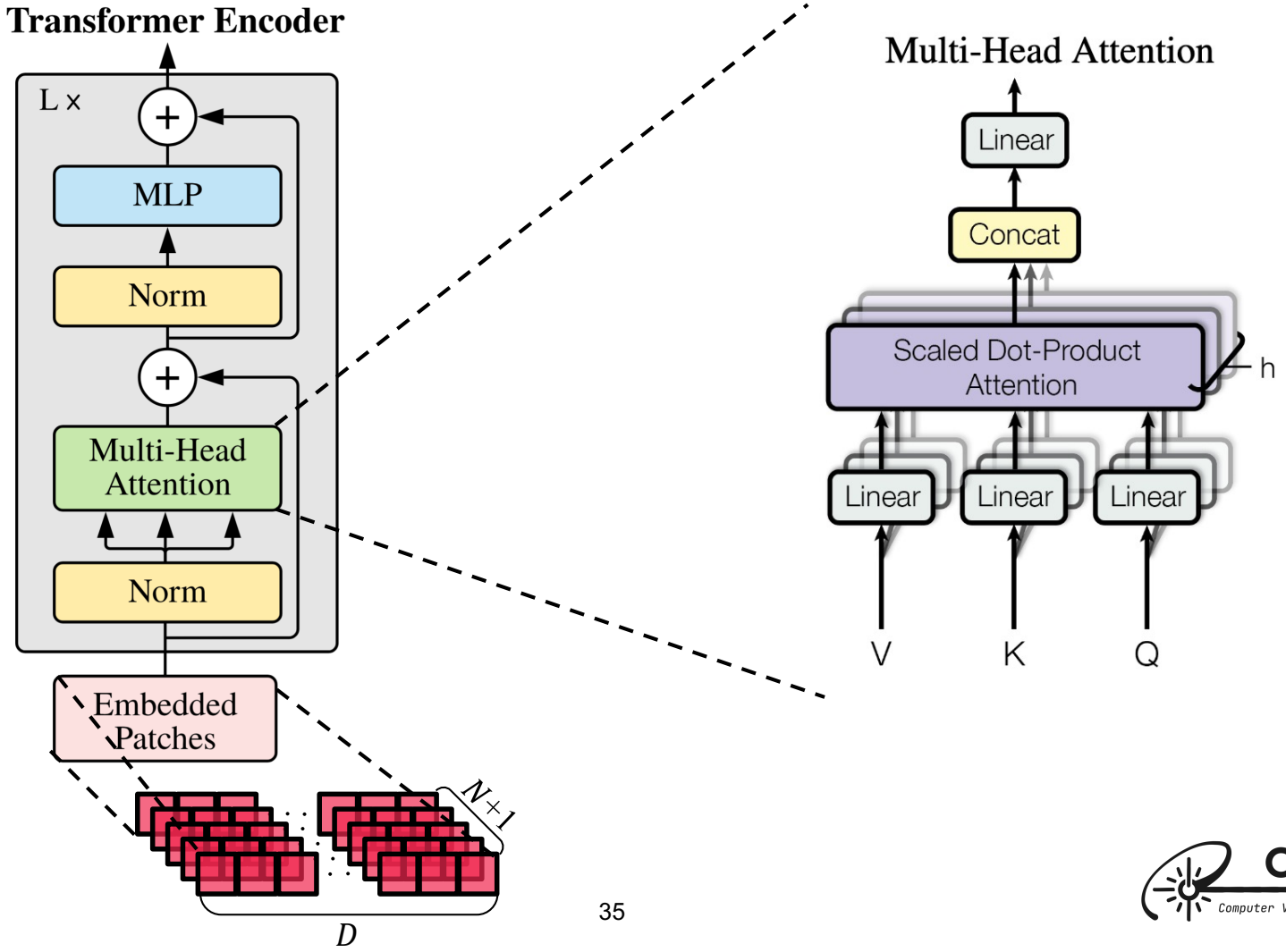


CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

- 3.3 Transformer Encoder





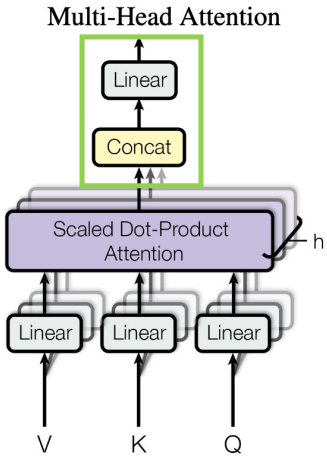
CVSP Lab
Computer Vision & Signal Processing

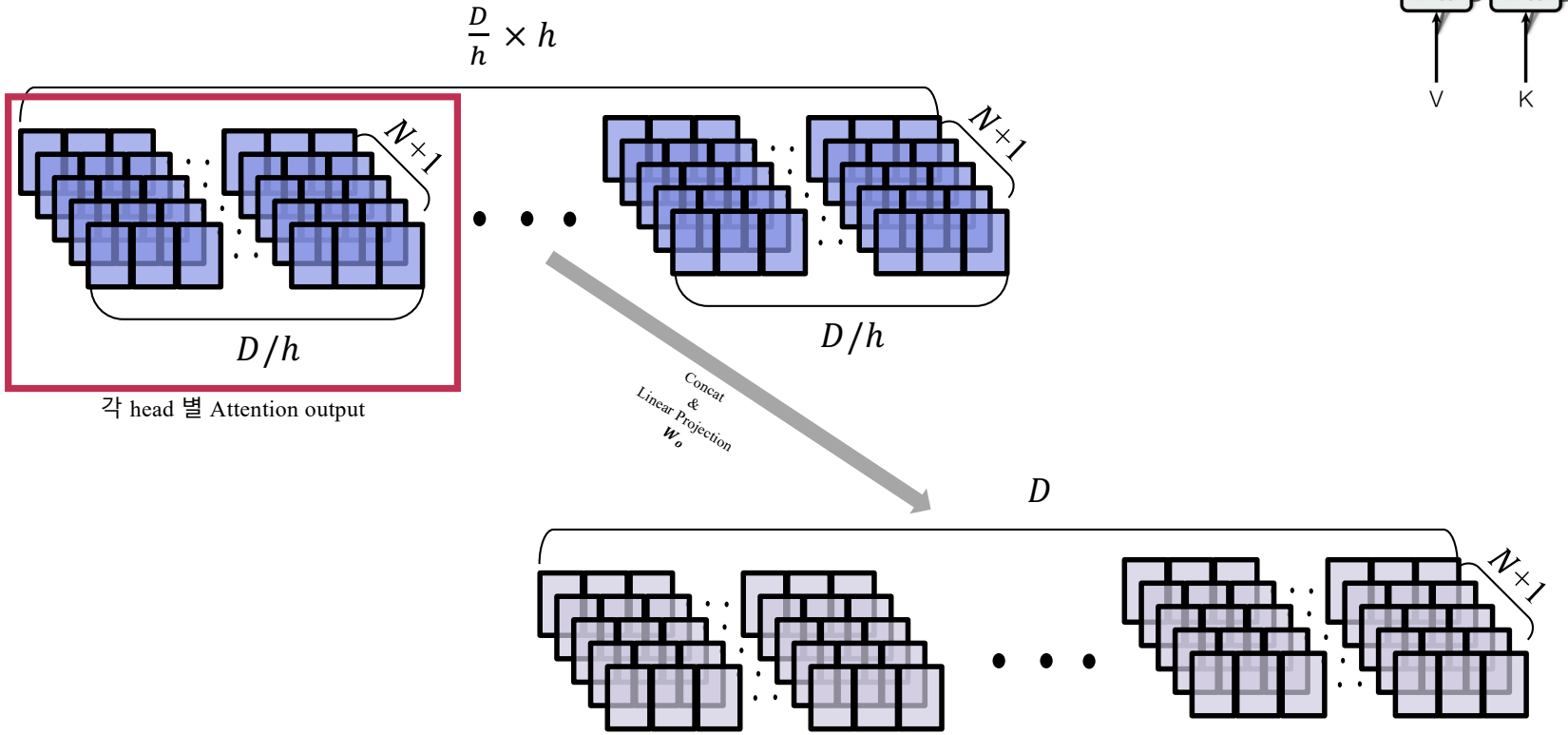
ViT – Architecture & Experiment

- 3.2 Multi-Head Attention

  Head를 나눔으로써, 다양한 관점을 반영한 표현 학습 가능

  $W_o \in R^{D \times D}$



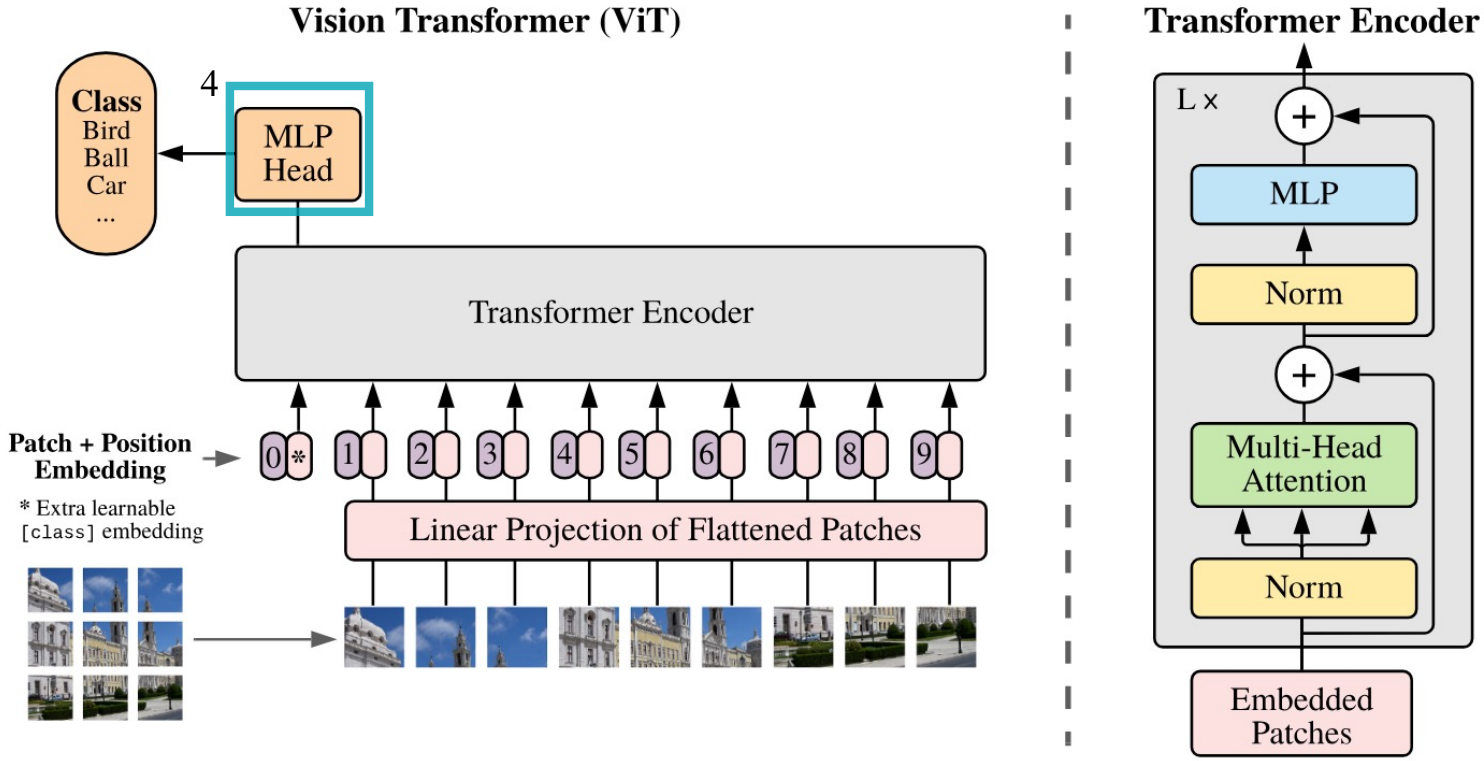


ViT – Architecture & Experiment

- 4. HEAD

  Transformer를 통과해서 class토큰에 image 정보를 잘 취합 후 MLP Head를 통과

  - Classification 수행



CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

- ViT 모델 종류

  Transformer Encoder의 구성요소에 따라 나뉨

  - BERT(언어 모델)의 구조를 따름

  패치의 크기까지 같이 표기

  - e.g. ViT-L/16: ViT-Large 모델, patch size = 16x16

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.

CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

- Experiment (1)

ViT vs. BiT-L(ResNet backbone): pretrained with JFT300M

- 성능 ↑
  - 모든 task에서 **ViT** > BiT-L
- 파라미터 개수 ↓
  - **ViT-L: 307M** < **ViT-H: 632M** < BiT-L: 약 928M
- 계산 비용 (TPUv3-core-days) ↓
  - **ViT** < BiT-L
    - ✓TPUv3-core-days = TPUv3 코어 개수 × 학습 기간(일)
- **결론**
  - **성능, 파라미터 효율성, 계산 비용 모두에서 ViT가 더 좋은 결과**

		Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
Fine-Tuning:	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

CVSP Lab
Computer Vision & Signal Processing

ViT – Architecture & Experiment

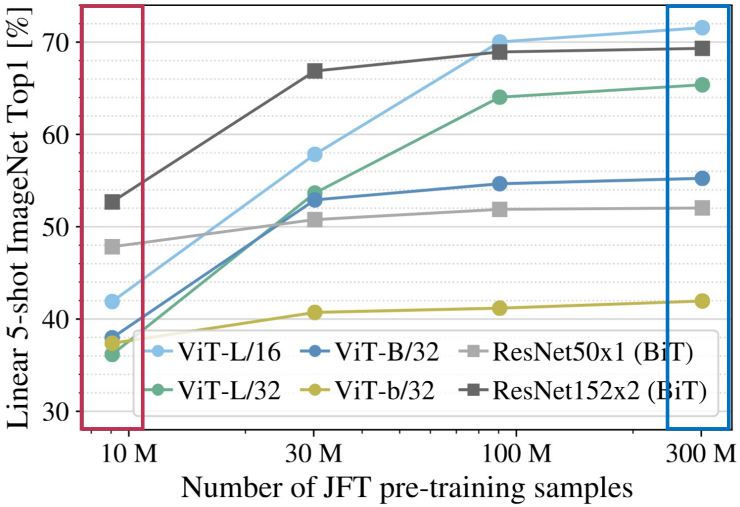
- Experiment (2)

  데이터셋 크기 변화에 따른 성능 변화 추이

  - ResNet Backbone
    - 작은 데이터셋에서 성능이 좋음, 빨리 포화됨
  - ViT
    - 큰 데이터셋에서 성능이 좋음, 늦게 포화됨

  원인

  - **Inductive Bias** 차이로 인해 발생



CVSP Lab
Computer Vision & Signal Processing

ViT vs. CNN – Focusing on Inductive Bias

2) Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261* (2018).

- Inductive Bias [2)]

정의

- Loss가 동등하게 평가하는 여러 해 중에서 특정 해를 선호하게 만드는 기준
  - **모델이 입력 데이터를 어떻게 해석할 것인가**에 대한 가정

예시

- **CNN → filter 범위 내 요소끼리 상호작용 (convolution)**
- RNN → 현 time step과 이전 time step과의 상호작용 ($h_t = f(h_{t-1}, x_t)$)

CVSP Lab
Computer Vision & Signal Processing

ViT vs. CNN – Focusing on Inductive Bias

- ViT 내의 어떤 구조가 Inductive Bias 부족의 영향을 받는가?

  가정 (1)

  - 작은 데이터셋 → ViT의 **position embedding**에 대한 학습이 제대로 되지 않음

  실험대상 : 작은 데이터셋으로 학습된 ViT

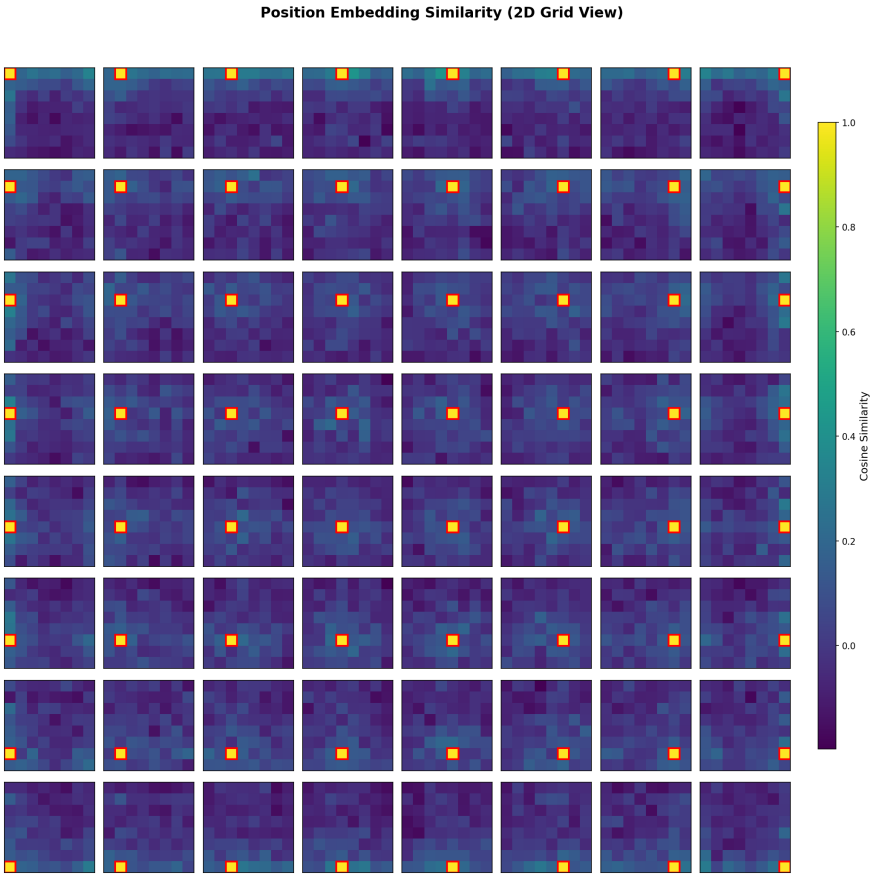
  - Model: ViT-b/4
  - Task: classification
  - Dataset: CIFA10 (작은 데이터셋)
  - Epoch: 150
  - Optimizer: Adam
  - Scheduler: Cosine annealing
  - **Result (test acc)**
    - **ViT-b/4: 80.6% → 성능이 충분히 높지 않음**

CVSP Lab
Computer Vision & Signal Processing

ViT vs. CNN – Focusing on Inductive Bias

- 결과

  patch의 위치 정보는 잘 학습됨



CVSP Lab
Computer Vision & Signal Processing

ViT vs. CNN – Focusing on Inductive Bias

3) Raghu et al. (2021). Do vision transformers see like convolutional neural networks?. Advances in *Neural Information Processing Systems (NeurIPS)*, 34, 12116-12128.
*: Appendix에서 추가 설명

- ViT 내의 어떤 구조가 Inductive Bias 부족의 영향을 받는가?

가정 (2)

- 작은 데이터셋 → **Transformer Encoder 내부에서** Local한 특징을 잘 학습 X

Experiment (3) [3)]

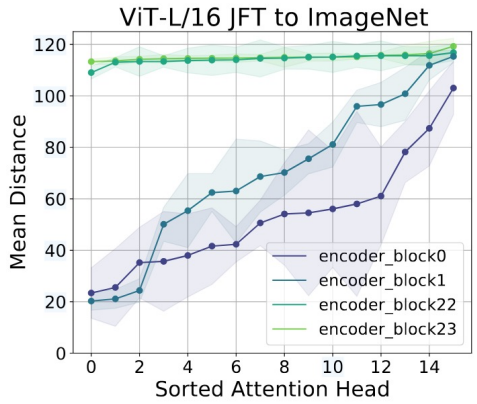
- 학습 Dataset 크기에 따른 Head 및 Layer 당 (Attention) Mean Distance 확인

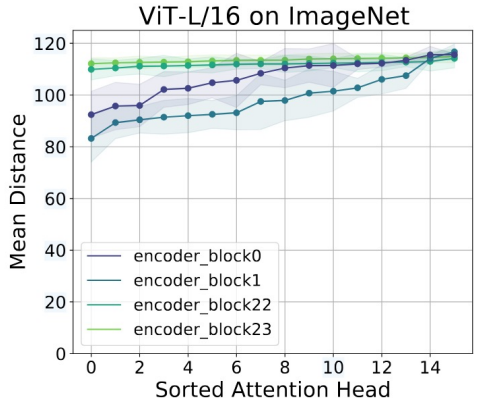
(Attention) Mean Distance*: 각 patch 간 거리와 그에 해당되는 attention weight를 이용하여 구한 **attention이 적용되는 평균 patch 거리**

✓값이 낮을수록, 인접한 patch와 attention을 수행

결론

✓**데이터가 작을수록, 얕은 layer에서 locality를 잘 학습하지 못함**





CVSP Lab
Computer Vision & Signal Processing

ViT vs. CNN – Focusing on Inductive Bias

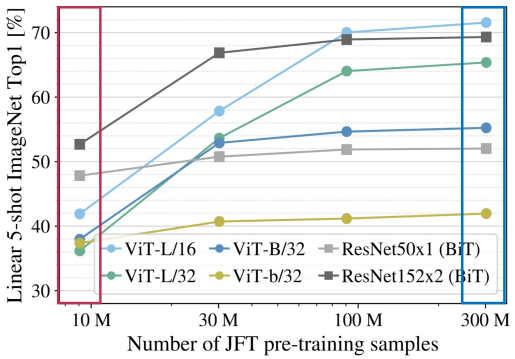
- 결론

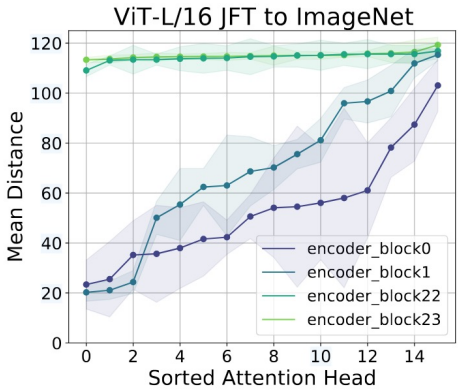
모델의 일반적 특징

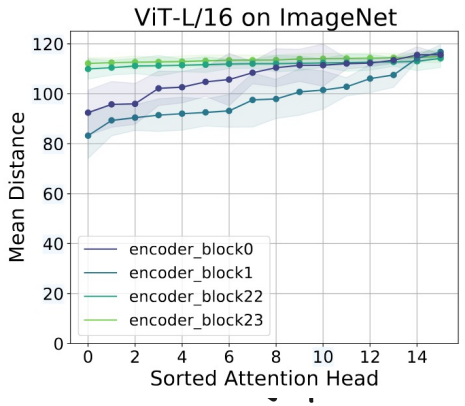
- 얕은 layer에서는 local한 특징을, 깊은 layer에서는 global한 특징을 학습하려 함

Inductive Bias

- CNN: 이미지에서 local한 부분에 집중하도록 하는 Inductive Bias **존재**
  - 학습 데이터가 작을 때 locality를 배우기 **수월함**
  - **Filter 범위의 한정된 locality** 학습 → 데이터셋이 커질수록 빨리 saturate
- ViT: Inductive Bias의 **부족**
  - 학습 데이터가 작을 때 locality를 배우기 **어려움** → 성능 하락
  - **한정되지 않은 유연한 locality**를 학습 → 데이터셋이 커져도 saturate되는 정도 약함







SP Lab
Signal Processing

Appendix

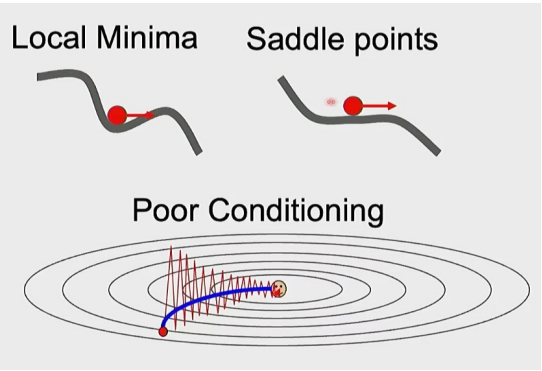
- Optimization – Adding Momentum

Stochastic Gradient Descent (SGD)

- 단점
  - **Local minima 혹은 Saddle point → Gradient가 0가 되어 학습 중단**
  - 단일 Learning rate 사용 → 각 feature에 대한 최적의 lr X → 진동 및 수렴 속도 하락 <<설명 명확X

SGD + **Momentum**

- Momentum: $v_t = \gamma v_{t-1} + \eta \nabla_\theta L(\theta)$
- 가중치 업데이트: $\hat{\theta} = \theta - v_t$
- 단일 SGD에서의 단점 해소
  - Local minima 또는 Saddle points에서 벗어날 수 있음
  - Minimum point로의 방향으로 momentum이 적용되어 진동 및 노이즈 감소, 수렴 속도 증가



CVSP Lab
Computer Vision & Signal Processing

Appendix

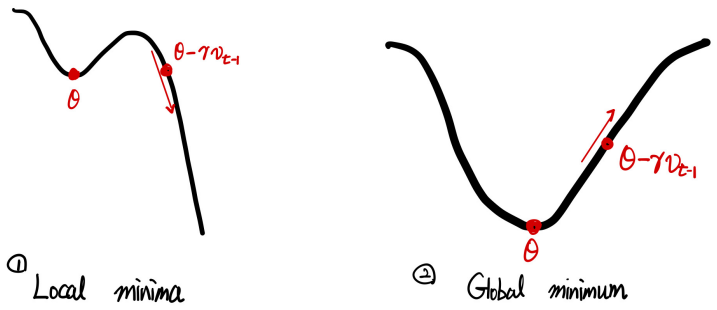
- Optimization – Adding Momentum

SGD + **Momentum**

- 단점
  - **Momentum으로 인해 Global minimum에서 심하게 진동 → 수렴속도 느려짐**
  - Momentum 계수(γ)를 조정 → Momentum scale이 변화 → 학습률도 그에 맞게 조정

NAG

- Momentum: $v_t = \gamma v_{t-1} + \eta \nabla_\theta L(\theta - \gamma v_{t-1})$
- 가중치 업데이트: $\hat{\theta} = \theta - v_t$
- 일반적인 Momentum에서의 단점 해소
  - 현재 Momentum을 이용하여 다음 step의 gradient 정보를 확인, 업데이트



CVSP Lab
Computer Vision & Signal Processing

Appendix

- Optimization – Adaptive Learning Rate

Stochastic Gradient Descent (SGD)

- 단점
  - Local minima 혹은 Saddle point → Gradient가 0가 되어 학습 중단
  - **단일 Learning rate 사용 → 각 feature에 대한 최적의 lr X → 진동 및 수렴 속도 하락**

AdaGrad

- 가중치 업데이트: $\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{G_{t-1}+\epsilon}} \cdot \nabla_\theta L(\theta_{t-1})$ ← element-wise product
- G_t : t 시점에서 각 feature 당 Gradient 누적량
  - $G_t = G_{t-1} + (\nabla_\theta(L(\theta_t))^2$
- SGD의 단일 Learning Rate 문제 해결
  - **각 feature에 맞는 learning rate를 사용함으로써 해결**

CVSP Lab
Computer Vision & Signal Processing

Appendix

- Optimization – Adaptive Learning Rate

AdaGrad

- 단점
  - $\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{G_{t-1}+\epsilon}} \cdot \nabla_\theta L(\theta_{t-1})$ 에서 G_{t-1}이 계속 커지므로 step size가 점점 감소
  - **학습의 후반부에서는 step size가 0과 가까워져 파라미터가 거의 업데이트되지 않는 현상 발생**

RMSprop

- 가중치 업데이트: $\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{G_{t-1}+\epsilon}} \cdot \nabla_\theta L(\theta_{t-1})$
- G_t: t 시점에서 각 feature 당 Gradient 누적량 (벡터)
  - $G_t = \gamma G_{t-1} + (1-\gamma)(\nabla_\theta (L(\theta_t))^2$
- AdaGrad의 단점 해결
  - 지수 이동 평균(Exponential Moving Average, EMA)을 적용하여 G_t의 값이 무한히 커지지 않고 일정한 수준을 유지
  - **이는 step size가 0에 가까워지는 것을 방지, 지속적인 학습 가능**

CVSP Lab
Computer Vision & Signal Processing

Appendix

- Optimization – Momentum + Adaptive Learning Rate

Adam

- Momentum과 Adaptive Learning rate 방식을 결합 (Momentum + RMSProp)
- Momentum: $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
  - EMA 적용 → momentum의 scale이 gradient와 같아짐 → β의 값과 무관하게 학습률 설정 가능
  - **EMA에서 오는 단점 (초기 momentum이 너무 작아짐)이 발생**
- Bias Correction: EMA 단점 해결
  - Momentum: $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$
  - RMSProp: $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$

- 가중치 업데이트: $\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}+\epsilon}} \cdot \hat{m}_t$

CVSP Lab
Computer Vision & Signal Processing

Appendix

- Grad-CAM

  특정 layer 기준, 각 feature map의 중요도에 따른 가중합을 수행

  - 목표 class와 관련 있는 패턴 부각, 무관한 패턴 희석

  식: $L^c_{Grad-CAM} = ReLU(\sum_k a^c_k A^k)$

  - A^k: 특정 layer의 k번째 feature map
  - a^c_k : class score에 대한 특정 layer의 k번째 feature map의 중요도(민감도)

    $a^c_k = \frac{1}{i+j} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$

  - ReLU 사용 → Gradient가 양수인 부분만 확인 → class에 대한 중요한 feature만 확인

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

  1. Baseline 비교

  - VGG11, VGG13, VGG16, VGG19 모델을 동일한 hyperparameter로 30 epoch동안 학습
  - 다른 seed로 4번 실험 수행

  결과

Model	Mean Loss	std	count
VGG11	0.6563	0.0143	4
VGG13	0.5927	0.0217	4
VGG16	**0.5783**	**0.0151**	**4**
VGG19	0.5980	0.0241	4

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

2. Learning Rate Grid Search (1)

- 아래와 같이 Grid Search 수행

  - Variable: lr = 5e-4, 1e-4, 5e-3, 1e-3
  - Fix:
    - ✓Epochs: 30
    - ✓Batch size: 512
    - ✓Weight decay: 0
    - ✓Optimizer: Adam
    - ✓Scheduler: None

결과

Model	lr	Val loss
VGG16	5e-4	0.884
	1e-4	1.114
	5e-3	0.699
	1e-3	**0.631**

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

3. Learning Rate Grid Search (2)

- 아래와 같이 Grid Search 수행

  - Variable: lr = 1e-3, 1.25e-3, 1.5e-3, 1.75e-3, 2e-3, 2.25e-3, 2.5e-3, 2.75e-3, 3e-3
  - Fix:
    - ✓Epochs: 30
    - ✓Batch size: 512
    - ✓Weight decay: 0
    - ✓Optimizer: Adam
    - ✓Scheduler: None

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

3. Learning Rate Grid Search (2)

- 결과: lr=2.25e-3 결정

Model	lr	Val loss
VGG16	1e-3	0.6312
	1.25e-3	0.5910
	1.5e-3	0.5988
	1.75e-3	0.5982
	2e-3	0.5782
	2.25e-3	**0.5699**
	2.5e-3	0.6256
	2.75e-3	0.6273
	3e-3	0.5835

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

4. Weight Decay Grid Search

- 아래와 같이 Grid Search 수행

  - Variable: Weight Decay = 0.4e-5, 0.6e-5, 0.8e-5, 1.2e-5, 1.4e-5, 1.6e-5
  - Fix:
    - ✓Epochs: 30
    - ✓Batch size: 512
    - ✓Learning rate: 2.25e-3
    - ✓Optimizer: Adam
    - ✓Scheduler: None

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

  4. Weight Decay Grid Search

  - 결과: weight decay=0.8e-5 결정

Model	Weight Decay	Val loss
VGG16	0.4e-5	0.589
	0.6e-5	0.559
	0.8e-5	**0.556**
	1.2e-5	0.629
	1.4e-5	0.596
	1.6e-5	0.619

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

5. Scheduler

- 아래와 같이 수행
  - Variable: Cosine Annealing / Step
  - Fix:
    - ✓Epochs: 30
    - ✓Batch size: 512
    - ✓Learning rate: 2.25e-3
    - ✓Weight Decay: 0.8e-5
    - ✓Optimizer: Adam

- 결과
  - **Cosine Anneling: val acc = 85%**
  - Step: val acc = 80%

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

6. FC layer Dropout

- 아래와 같이 Grid Search 수행

  - Variable: FC dropout = 0.1, 0.2, 0.3, 0.4, 0.5
  - Fix:
    - ✓Epochs: 30
    - ✓Batch size: 512
    - ✓Learning rate: 2.25e-3
    - ✓Weight Decay: 0.8e-5
    - ✓Optimizer: Adam
    - ✓Scheduler: Cosine Anneling

Appendix

- VGG Hyperparameter Tuning

  6. FC layer Dropout

  - 결과: FC dropout =0.3 결정

Model	FC dropout	Val loss
VGG16	0.1	0.670
	0.2	0.649
	0.3	**0.595**
	0.4	0.603
	0.5	0.596

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

7. Data Augmentation

- Flip + random crop

Fix:

- ✓ Epochs: 40
- ✓ Batch size: 512
- ✓ Learning rate: 2.25e-3
- ✓ Weight Decay: 0.8e-5
- ✓ Optimizer: Adam
- ✓ Scheduler: Cosine Anneling

Val acc: 88.78%

CVSP Lab
Computer Vision & Signal Processing

Appendix

- VGG Hyperparameter Tuning

7. Data Augmentation

- Flip + random crop + Cutout
  - Fix:
    - ✓Epochs: 40
    - ✓Batch size: 512
    - ✓Learning rate: 2.25e-3
    - ✓Weight Decay: 0.8e-5
    - ✓Optimizer: Adam
    - ✓Scheduler: Cosine Anneling
  - Val acc: 82.47%

- Flip + random crop + Cutout이 Augmentation 정도가 심해 학습이 덜 되었다고 판단 → epoch 늘려주기
  - 80 Epochs – val acc: 87.74%
  - **150 epochs – val acc: 89.12%**
    - ✓**Test acc: 92.62%**

CVSP Lab
Computer Vision & Signal Processing