

Artificial Intelligence

Assignment #3.

Dissecting the CNNs

—
Byeon, Seunggyu

sg0919@deu.ac.kr

Dept. of Computer Eng.

Note) Some open-source fonts, such as [Pretendard](#) fonts, [Naver Nanum](#) fonts and [Google Raleway](#) fonts, are used in this slide. Download and Install them, if you see broken characters



동의대학교
DONG-EUI UNIVERSITY

Machine Learning

Assignment #2. Hand-crafting MLP

—

- a. CNN Operations
- b. CNN Learning / MLP Learning
- c. Performances
- d. Discussions



인공지능 (3학년 1학기)

주차별 학습계획 (Weekly Schedule)

주	구 분	학 습 내 용	활동사항
1	Fundamental of AI	Course Introduction / AI ?	Assignment #1
2		Fundamental Mathematics for Understanding AI	
3		Fundamentals of Machine Learning	Assignment #1
4	Artificial Neural Networks	Basic Model of Neural Networks	
5		Practical Exercises with Basic Neural Networks	Assignment #2
6		CNN: Convolutional Neural Networks	
7		RNN: Recurrent Neural Networks	
8		Mid Term Exam	
9	Cutting-Edge Themes	Practical Exercises with CNN and RNN	Assignment #3
10		GAN: Generative Adversarial Network	
11		Practical Exercises with GAN	Assignment #4
12		Reinforcement Learning and Game Intelligence	
13		Spatial Exploration and Problem Solving	
14		XAI: Explainable Artificial Intelligence, and AGI	
15		Final Exam	

Part. 1

Understanding CNN Operations

—

Resizing

Two-Dimensional Convolution

Max Pooling

CNN Operations in LeNet-5



동의대학교
DONG-EUI UNIVERSITY

Understanding CNN Operations

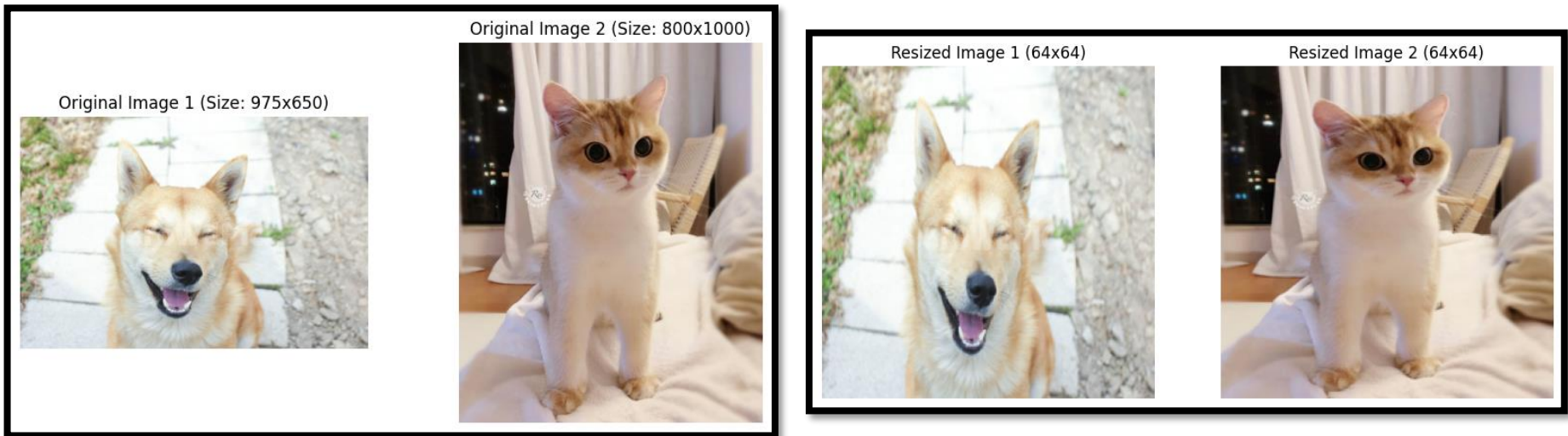
Needs for Resizing

- **Consistency:**
 - CNN은 일관된 입력 크기를 필요로 함
 - 모든 입력 이미지를 동일한 크기로 조정해야 학습에 필요한 일관성을 유지할 수 있음
- **Computational Efficiency:**
 - 메모리 사용량을 줄이고,
 - 학습 속도를 향상시킬 수 있음
 - 특히, 큰 데이터셋을 다룰 때 중요하게 작용함
- **Performance Optimization:**
 - 특징 추출이 더욱 효율적으로 이루어질 수 있음
 - 너무 크거나 작은 이미지는 중요한 특징을 손실하거나 불필요한 정보를 포함할 수 있음
- 코드의 주석에 따라 Resizing을 수행하는 코드를 완성해보자

Understanding CNN Operations

Needs for Resizing

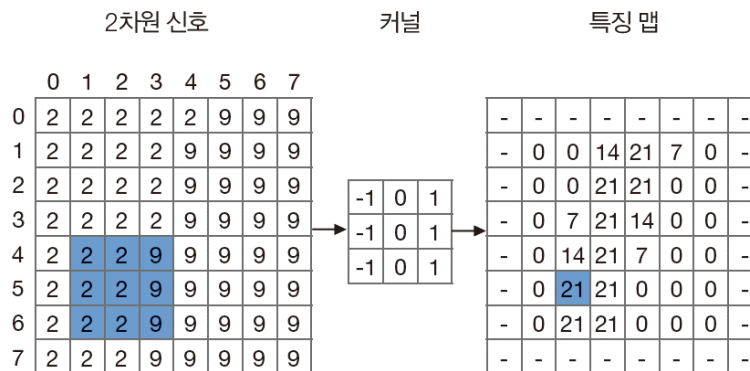
- 코드의 주석에 따라 Resizing을 수행하는 코드를 완성해보자
- 참고 결과물
 - [D3-Example-Dog.jpg](#) 와 [D3-Example-Cat.jpg](#)
 - 실제로는 두 묶음의 그림이 아래위로 출력됨



Understanding CNN Operations

Two-Dimensional Convolution

- **Filtering:**
 - 커널이 입력 이미지를 순회, 각 위치에서 필터와 이미지 세그먼트 간의 원소별 곱셈을 수행
- **Feature Extraction:**
 - 이 과정에서 필터는 이미지의 특정 패턴이나 특성(예: 가장자리, 각도, 질감)을 감지



- **Emphasizing Feature:**
 - 필터 설계에 따라 다양한 이미지 특성을 강조할 수 있음
 - 예를 들어, 수직 필터는 수직 가장자리를, 수평 필터는 수평 가장자리를 감지
- **Diversity:**
 - CNN은 여러 필터를 동시에 사용하여 다양한 특징을 감지, 이를 통해 복잡한 특징을 학습

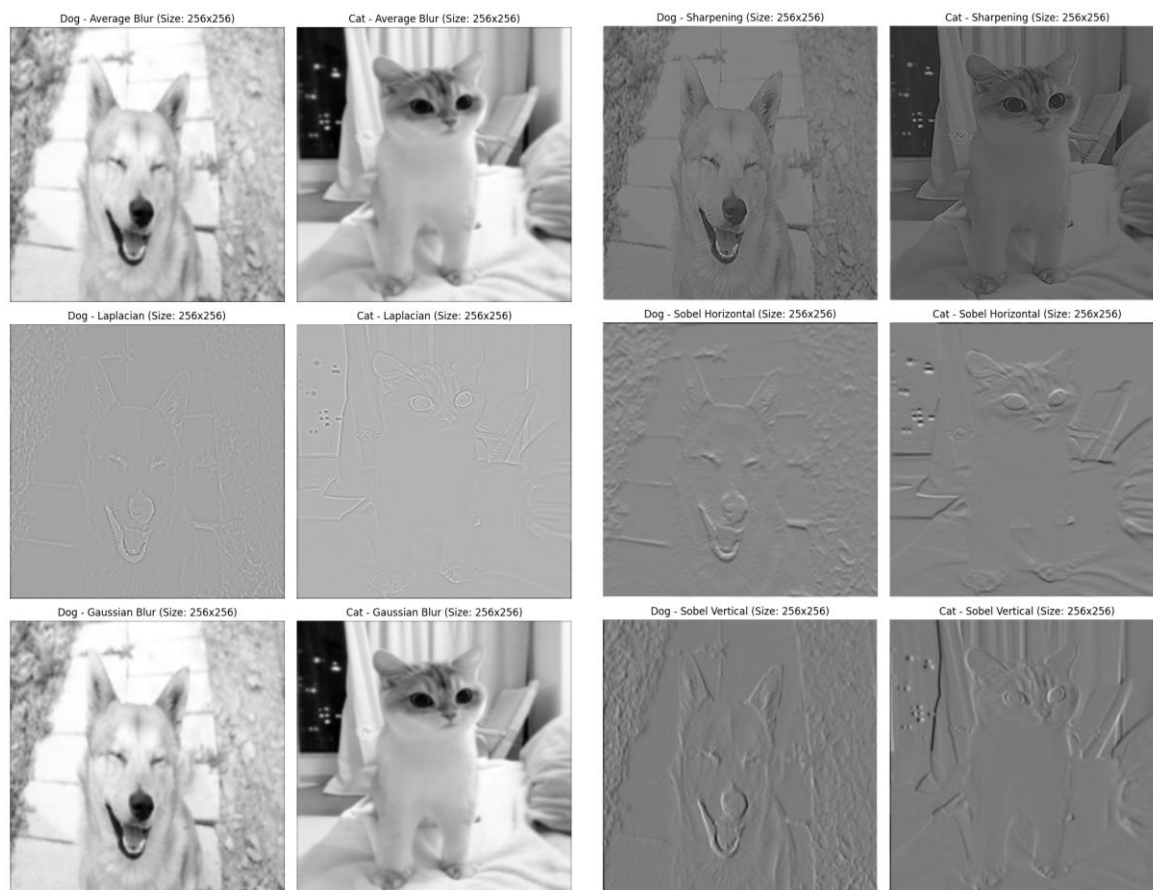
Understanding CNN Operations

Two-Dimensional Convolution

- 코드의 주석에 따라 2d Convolution을 수행하는 코드를 완성해보자

- 참고 결과물

- 여기서부터는
 참고 이미지의 크기와
 결과물의 크기가 다를 수 있음



Understanding CNN Operations

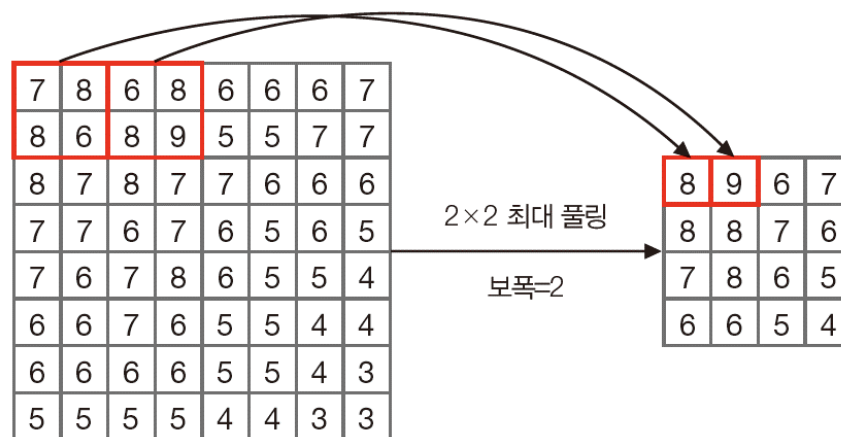
Max Pooling

- **Pooling Window** (or Pooling Filter):

- 일반적으로 2x2 크기의 윈도우를 사용
- 입력 맵을 순회하며 같은 연산을 수행

- **Maximum Value Extraction:**

- 각 윈도우 내에서 최대값만을 추출
- 나머지 값은 버림



- **Dimensional Reduction:**

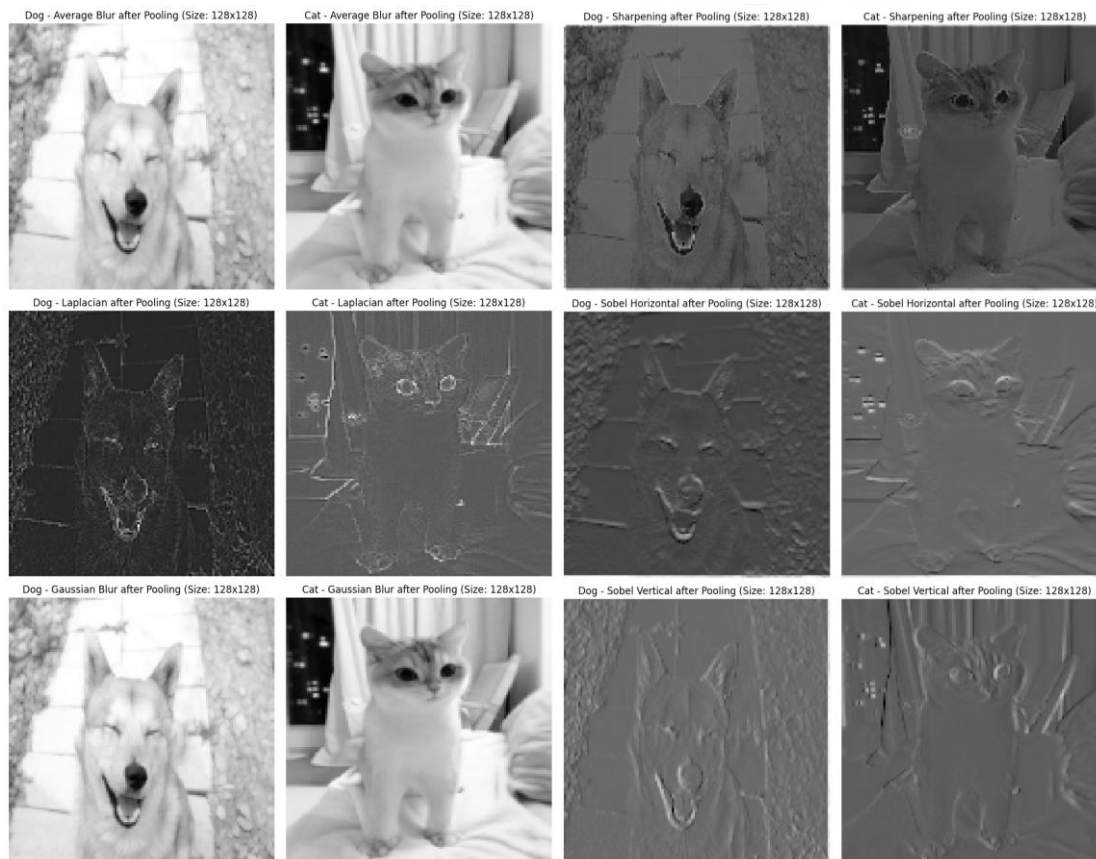
- 결과, 원본의 특징 맵보다 크기가 작은 새로운 맵이 생성됨
- 이 과정은 입력 데이터의 공간적 크기를 줄이는 역할을 함

Understanding CNN Operations

Max Pooling

- 코드의 주석에 따라 2x2 Max Pooling을 수행하는 코드를 완성해보자

- 참고 결과물



Understanding CNN Operations

CNN Operations in LeNet-5

- LeNet-5의 C-P-C-P-C-FC-FC 연산을 모방하여 일부 연산을 수행하자
 - 이 중 C-P-C-P의 연산을 수행해보자
 - ✓ Convolution에 의해 이미지의 윤곽선 특징이 추출되고
 - ✓ MaxPooling에 의해 이미지가 축약되는 과정을 살펴보기 위함
 - 입력: $1 \times 32 \times 32$ 텐서 (32×32 크기의 개, 고양이 이미지 특징 맵)
 - 빌딩 블록1
 - ✓ 컨볼루션은 5×5 커널을 6개 사용하지만 여기서는 하나만 사용함
(Stride 1, padding='same'): 32×32 텐서 \rightarrow 32×32 텐서
 - ✓ 풀링은 2×2 커널을 사용 (Stride 2): 32×32 텐서 \rightarrow 6×16 텐서
 - ✓ 동일 과정을 1회 더 반복
 - 가능하다면 연산의 매개변수를 바꾸어 실제 LeNet-5처럼 동작하게 해보자

Understanding CNN Operations

CNN Operations in LeNet-5

- 참고 결과물

Dog - After First Convolution (Size: 256x256)



Dog - After First Pooling (Size: 128x128)



Dog - After Second Convolution (Size: 128x128)



Dog - After Second Pooling (Size: 64x64)



Cat - After First Convolution (Size: 256x256)



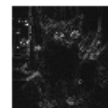
Cat - After First Pooling (Size: 128x128)



Cat - After Second Convolution (Size: 128x128)



Cat - After Second Pooling (Size: 64x64)



Part. 2

CNN Learning

—

Preparing Image Data

Constructing CNN Networks

Facilitating CNN Learning



동의대학교
DONG-EUI UNIVERSITY

CNN Learning

Preparing Cat-Dog Image Data

- 데이터셋 압축 파일
 - D3-Cat-vs-Dog-Tr.zip
 - D3-Cat-vs-Dog-Te.zip # 추후 필요시 사용

이름	유형
training_set	파일 폴더



이름	유형
cats	파일 폴더
dogs	파일 폴더



이름	유형
_DS_Store	파일
cat.1	JPG 파일
cat.2	JPG 파일
cat.3	JPG 파일
cat.4	JPG 파일
cat.5	JPG 파일



이름	유형
_DS_Store	파일
dog.1	JPG 파일
dog.2	JPG 파일
dog.3	JPG 파일
dog.4	JPG 파일
dog.5	JPG 파일

CNN Learning

Preparing Cat-Dog Image Data

- 데이터셋 압축 파일
 - D3-Cat-vs-Dog-Tr.zip
 - D3-Cat-vs-Dog-Te.zip # 추후 사용
- 참고 결과물
 - 8005개의 이미지
 - 이미지 당 32 x 32의 픽셀
 - 두 개 열을 사용하는 레이블
 - ✓ 실제로는 하나의 레이블만 필요하지만,
 - ✓ one-hot 인코딩 과정에서 0과 1을 분리하기 위해 두 개의 열을 사용하게 됨

```
Number of images: 8005
Shape of images: (8005, 32, 32, 1)
Shape of labels: (8005, 2)
```

CNN Learning

Preparing Cat-Dog Image Data

- 불러온 이미지 출력하기
 - 100개의 이미지가 10 x 10으로 출력되게 할 것
- 참고 결과물 (일부)



CNN Learning

Constructing CNN Networks

- CNN 망을 구성할 것
 - Input-C-P-C-P-C-F-FCN-Output
 - 디테일은 주석을 참고
- 성능 비교 대상인 MLP 망을 구성할 것
 - Input-FCN-Output

CNN Learning

Constructing CNN Networks

- 참고 결과물
 - CNN vs. MLP

Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
conv2d_97 (Conv2D)	(None, 32, 32, 6)	156	flatten (Flatten)	(None, 1024)	0
max_pooling2d_94 (MaxPooling2D)	(None, 16, 16, 6)	0	dense (Dense)	(None, 84)	86100
conv2d_98 (Conv2D)	(None, 12, 12, 16)	2416	dense_1 (Dense)	(None, 1)	85
max_pooling2d_95 (MaxPooling2D)	(None, 6, 6, 16)	0	===== Total params: 86185 (336.66 KB) Trainable params: 86185 (336.66 KB) Non-trainable params: 0 (0.00 Byte) =====		
conv2d_99 (Conv2D)	(None, 2, 2, 120)	48120			
flatten_60 (Flatten)	(None, 480)	0			
dense_179 (Dense)	(None, 84)	40404			
dense_180 (Dense)	(None, 1)	85			
===== Total params: 91181 (356.18 KB) Trainable params: 91181 (356.18 KB) Non-trainable params: 0 (0.00 Byte) =====					

CNN Learning

Facilitating CNN Learning

- 신경망 입출력 데이터셋 설정하기
 - 2열로 표현된 y에서 고양이를 0, 개를 1로 표현한 데이터만 추출하기
 - Training Set과 Test Set을 3:1로 분할하기
 - 초 매개변수 설정하기
 - ✓ Optimizer: adam
 - ✓ Loss: Binary Cross Entropy
 - ✓ Metric: Accuracy
- 학습 수행
 - 출력되는 각 Metric의 변화를 살펴보자
 - 최종 선택된 본인의 모델의 손실과 정확도는 얼마인가?

CNN Learning

Facilitating CNN Learning

- 외부 테스트셋에 관한 성능 비교
 - 외부 테스트셋: [D3-Cat-vs-Dog-Te.zip](#)
 - 위 테스트셋을 불러와서 lenet_model과 mlp_model로 성능을 비교하라
- 참고 정확도
 - CNN: 70% 이하 수준
 - MLP: 60% 남짓 수준

CNN Learning

Facilitating CNN Learning

- 두 모델의 성능을 높이기 위한 방안을 제시하고 검증하라
 - Part. 3까지 마무리 지은 다음에 시도할 것을 권장
 - 예) 망 구조의 다변화
 - ✓ C-P 망의 추가
 - ✓ FC 계층의 추가
 - ✓ Dropout의 사용
 - ✓ Weight Decay의 사용
 - ✓ Loss 함수의 변경 등
 - ✓ 가능한 한 모든 매개변수와 초 매개변수의 변경
 - 여기서 성능이란, 학습 과정에 사용되지 않은 별도의 데이터에 대한 성능을 의미한다.

Part. 3

Visualizing CNN Kernels

—

Convolution Kernels

Data Processing Stages

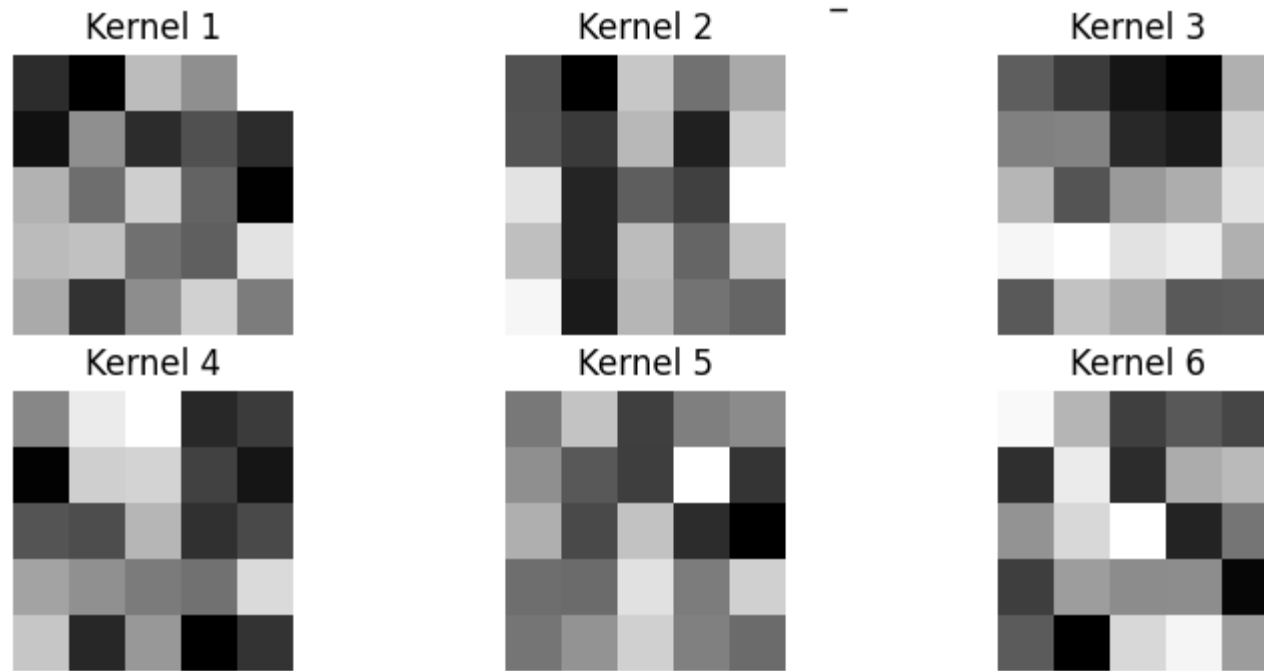


동의대학교
DONG-EUI UNIVERSITY

Visualizing CNN Kernels

Convolution Kernels

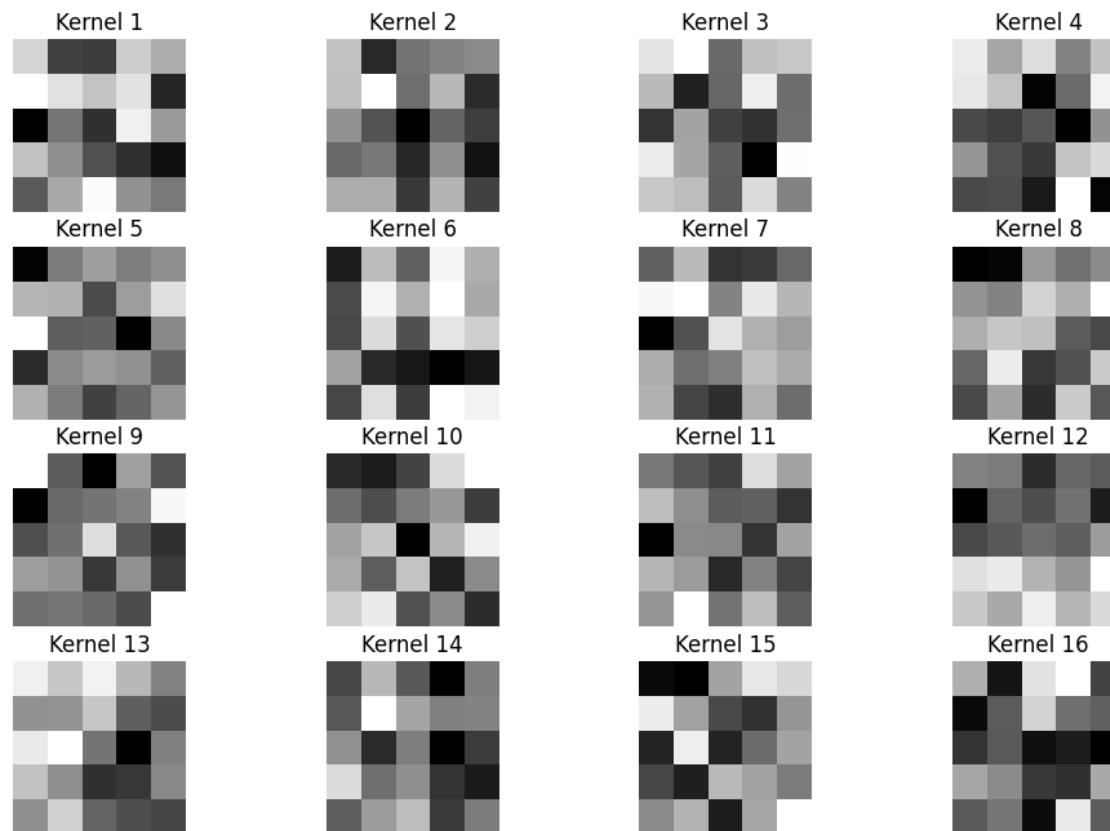
- 학습된 커널의 시각화
 - 예제 출력: C1층



Visualizing CNN Kernels

Convolution Kernels

- 학습된 커널의 시각화
 - 예제 출력: C2층



- 학습된 커널이 어떠한 의미를 갖는지 설명할 수 있겠는가?

Visualizing CNN Kernels

Data Processing Stages

- 최초의 예제 그림 두 장의 가공
 - 학습한 LeNet-5 모델에 입력하고 이미지가 처리되는 과정을 살펴보자
 - 먼저, 도중에 데이터를 추출 가능하도록 모델을 가이드라인에 따라 수정
 - ✓ 기본 코드는 모두 작성되어 있고, 매개변수만 조정하면 된다
 - 개와 고양이 그림이 단계적으로 어떻게 가공되며 최종 출력이 어떻게 되는지 살펴보자

Visualizing CNN Kernels

Data Processing Stages

- 참고 출력

Layer 1 output for Image 1 (Dog)



Layer 3 output for Image 1 (Dog)



```
1/1 [=====] - 0s 108ms/step
```

```
Prediction for Image 1 (Dog): Dog with probability 0.6845976114273071
```

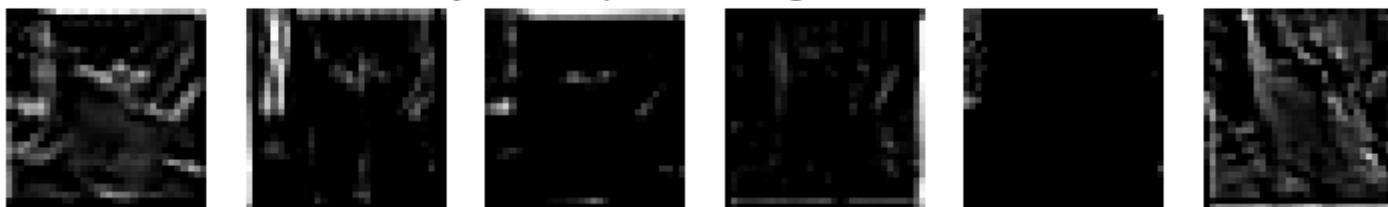
```
1/1 [=====] - 0s 77ms/step
```

Visualizing CNN Kernels

Data Processing Stages

- 참고 출력

Layer 1 output for Image 2 (Cat)



Layer 3 output for Image 2 (Cat)



1/1 [=====] - 0s 26ms/step

Prediction for Image 2 (Cat): Cat with probability 0.48545801639556885

감사합니다.

