[010238] 딥러닝시스템

세종대학교 소프트웨어융합대학 지능기전공학부

비전 시스템을 위한 딥러닝

Deep Learning for Vision Systems

본 자료는 한빛미디어에서 나온 〈비전 시스템을 위한 딥러닝〉 책을 참고하여 제작 되었음을 알립니다.

합성곱 신경망

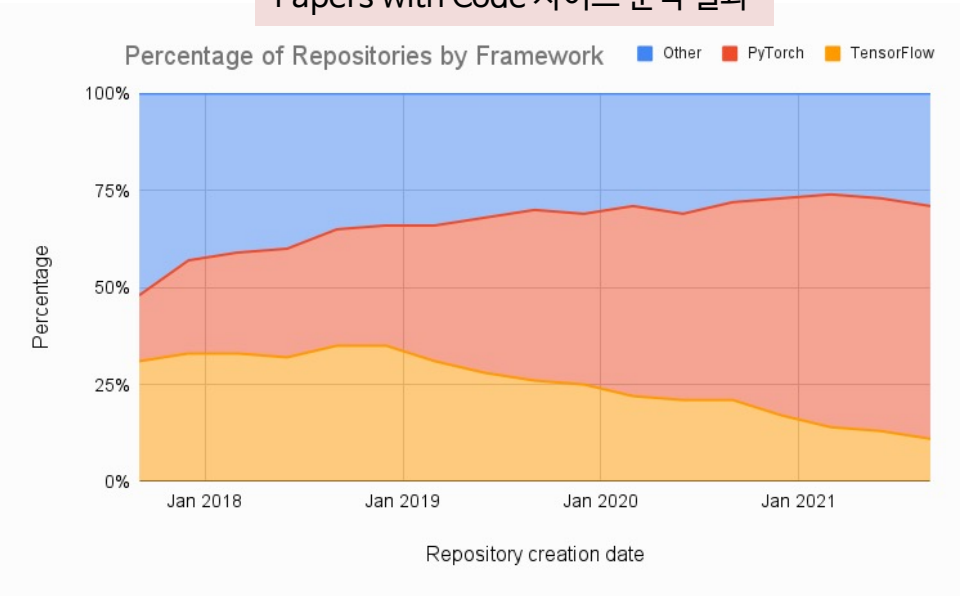
Convolutional Neural Networks

본 자료는 한빛미디어에서 나온 〈비전 시스템을 위한 딥러닝〉 책을
참고하여 제작 되었음을 알립니다.

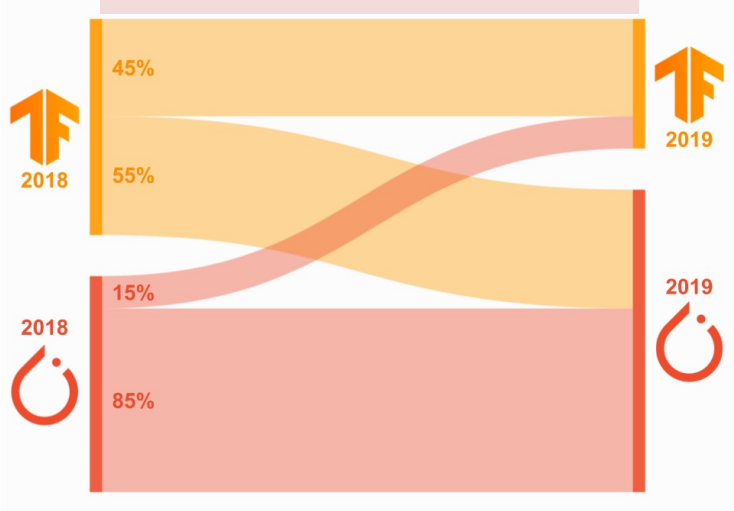
3. 합성곱 신경망

- 딥러닝 프레임워크
  - 모델 가용성 (Pytorch 〉 TensorFlow)

Papers with Code 사이트 분석 결과



빠른 모델 구현을 위한 연구자들의 선택



https://velog.io/@freejack/PyTorch-vs-TensorFlow-in-2022
https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/

3. 합성곱 신경망

- **딥러닝 프레임워크**
  - 배포 인프라 (TensorFlow 〉 Pytorch)
    - 텐서플로우는 처음부터 배포 지향 애플리케이션을 위한 필수 프레임워크 였음
    - 파이토치는 배포 관점에서 극도로 부진했지만 최근 몇 년 동안 격차를 좁히기 위해 노력함

TensorFlow	Pytorch
▪ TensorFlow Serving (클라우드 배포) ▪ TensorFlow Lite (모바일 배포)	▪ TorchServe (클라우드 배포) ▪ PyTorch Live (모바일 배포)

https://velog.io/@freejack/PyTorch-vs-TensorFlow-in-2022
https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/

3. 합성곱 신경망

- 딥러닝 프레임워크
  - 생태계 (TensorFlow == Pytorch)
    - 산업 환경에서 딥러닝 엔지니어링을 수행하는 경우 TensorFlow를 사용할 가능성이 높음
    - 최근 SOTA 모델을 액세스해야 하는 경우 Pytorch를 사용할 가능성이 높음

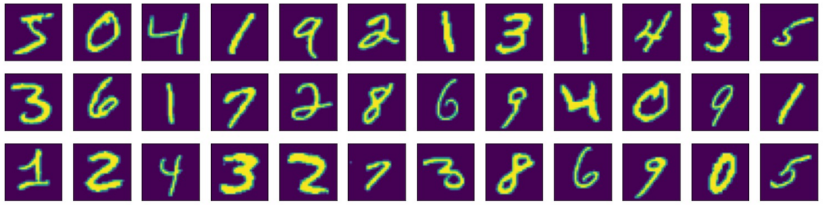
TensorFlow	Pytorch
▪ TensorFlow Hub (이미지, 비디오, 오디오, 텍스트 등 지원) ▪ Model Garden ▪ TensorFlow Extended (배포용) ▪ Vertex AI (교육용) ▪ TensorFlow.js (자바스크립트 지원용) ▪ TensorFlow Cloud (클라우드 지원용) ▪ Colab (클라우드기반 노트북 환경)	▪ Pytorch Hub (비전, 오디오, NLP등) ▪ SpeechBrain (음성툴킷) ▪ TorchElastic (분산 훈련용) ▪ TorchX (배포용 SDK) ▪ Lightning (교육용)

https://velog.io/@freejack/PyTorch-vs-TensorFlow-in-2022
https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/

3. 합성곱 신경망

- **수업에서 사용하는 딥러닝 프레임워크**
  - 케라스 소개 (Keras)
    - TensorFlow 2.0에 포함
    - 딥러닝 모델을 간편하게 만들고 훈련시킬 수 있는 파이썬을 위한 딥러닝 프레임워크
    - 신속하게 실험을 해야 하는 연구자들을 위해 개발되었음
    - MIT 라이센스를 따르므로 상업적인 프로젝트에도 자유롭게 사용 가능 (2018년 기준)
    - 파이썬 2.7에서 3.6까지 모든 버전과 호환
    - 공홈: https://keras.io/
    - 공홈-한글화: https://keras.io/ko/
  - 케라스의 특징
    - 동일한 코드로 CPU와 GPU 에서 실행할 수 있음
    - 사용하기 쉬운 APU를 가지고 있어 딥러닝 모델의 프로토타입을 빠르게 만들 수 있음
    - (컴퓨터 비전을 위한) 합성곱 신경망, (시퀀스 처리를 위한) 순환 신경망을 지원하며 이 둘을 자유롭게 조합하여 사용할 수 있음

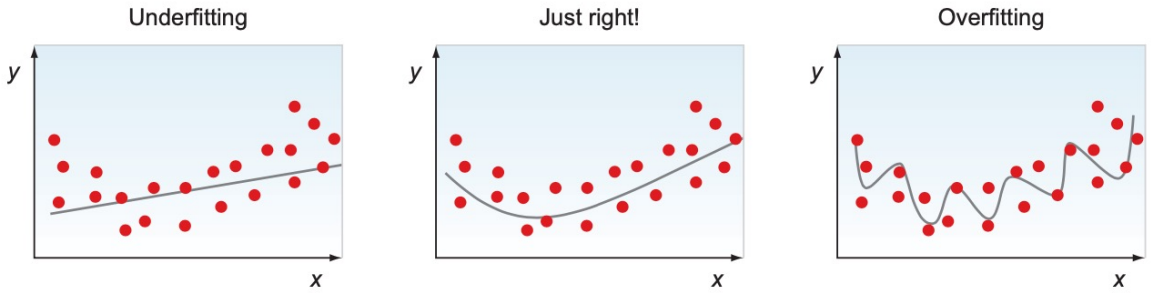
3. 합성곱 신경망



- **3.4 CNN을 이용한 이미지 분류**
  - CNN 모델을 학습하여 MNIST 데이터셋의 손글씨 이미지를 분류하는 것
  - 학습데이터: MNIST
    - 딥러닝계의 'Hello World' 데이터 셋
    - 0~9숫자 손글씨 데이터
    - 28x28 크기의 흑백 이미지 10가지 클래스로 구성
  - 학습모델
    - CNN 2개 + FC 2개
  - 테스트결과: 약 99% 정확도
  - 실습코드
    - CPU 를 활용한 교재 코드
      - https://www.kaggle.com/yukyungchoi/2022-dl-w3p1
    - Keras 메뉴얼 샘플 코드
      - https://keras.io/examples/vision/mnist_convnet/

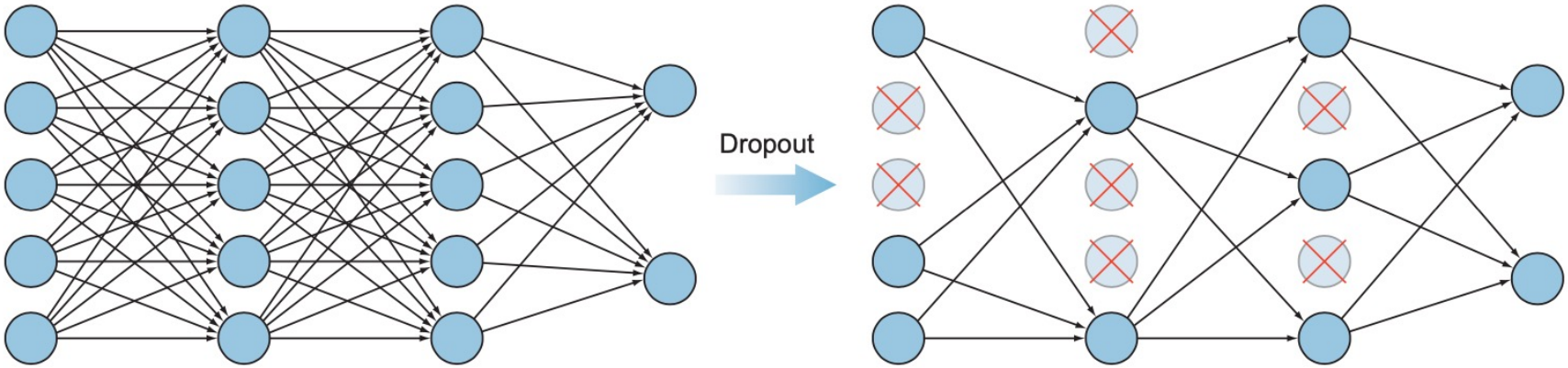
3. 합성곱 신경망

- **3.5 과적합을 방지하기 위해 드롭아웃층 추가하기**
- **3.5.1 과적합이란**
  - 머신러닝에서 모델의 성능이 잘 나오지 않는 이유 중 한가지
  - 과소적합 underfitting
    - 모델이 학습 데이터에 부합하지 못하는 현상
    - 모델이 데이터를 나타내기에 너무 단순한 경우에 발생
  - 과적합 overfitting
    - 모델이 학습 데이터에 지나지게 부합하는 현상 (학습 오차가 매우 적음)
    - 모델이 표현력이 매우 좋은 신경망
    - 처음 보는 데이터는 잘 예측하지 못해 일반화 성능이 떨어지는 경우



3. 합성곱 신경망

- **3.5.2 드롭아웃층이란**
  - 과적합을 방지하기 위한 수단 중 가장 널리 사용
  - 드롭아웃층을 적용하면 층을 구성하는 뉴런(노드)의 일정 비율을 비활성화 함
    - 비활성화란 순방향 혹은 역전파 계산에 참여하지 않는다는 뜻
  - 이 비율은 하이퍼 파라미터로 지정
  - 학습단계에서만 사용되는 층 (테스트 단계에서는 사용되지 않음)
  - 모든 노드가 협력하여 영향력이 지나치게 약하거나 강한 노드가 생기는 것을 방지



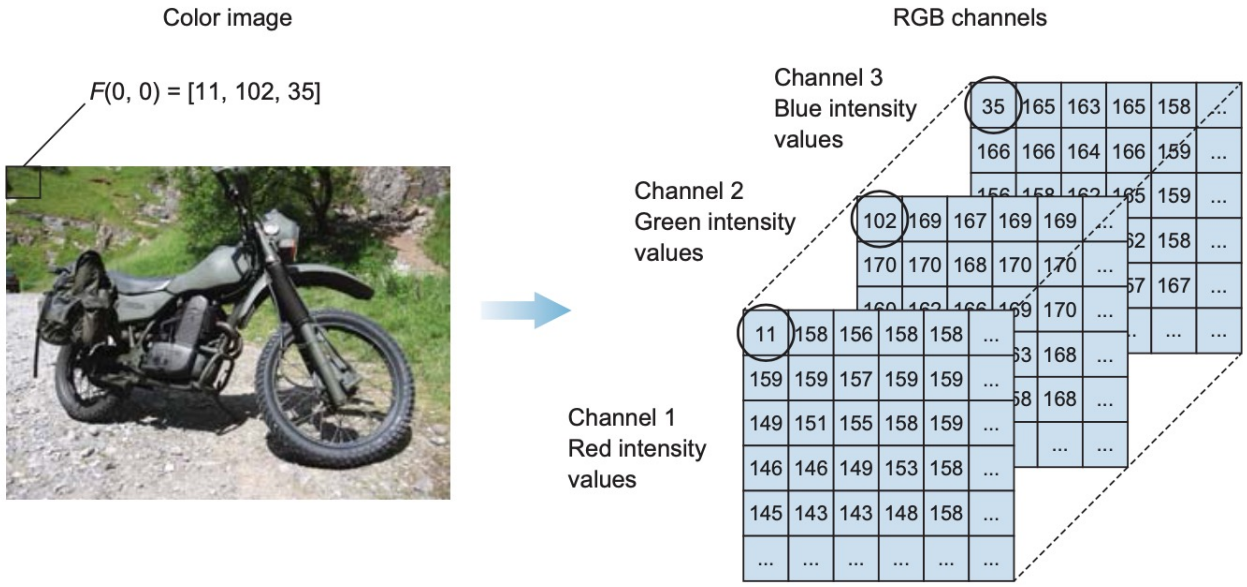
3. 합성곱 신경망

- **3.5.3 드롭아웃층이 필요한 이유**
  - 뉴런은 학습 과정을 거치며 상호 의존 관계를 구축
  - 상호 의존관계는 각 뉴런의 영향력을 결정하며 과적합으로 이어지는 원인이 됨
  - 노드 중 일부를 무작위로 비활성화한다면 다른 노드는 비활성화된 노드가 가진 특징 없이 패턴을 학습해야 하며, 모든 특징이 이런식으로 비활성화될 수 있으므로 가중치가 특징 간에 고르게 분산되는 효과가 발생하며 더 잘 학습된 뉴런으로 이어짐
    - 드롭아웃은 일종의 앙상블 학습 기법임

- **3.5.4 CNN 구조 중 어디에 드롭아웃층을 끼워 넣어야 할까**
  - 드롭아웃층은 추출된 특징의 1차원 벡터 변환이 끝난 다음부터 마지막 출력 층 사이에 배치하는 것이 일반적임

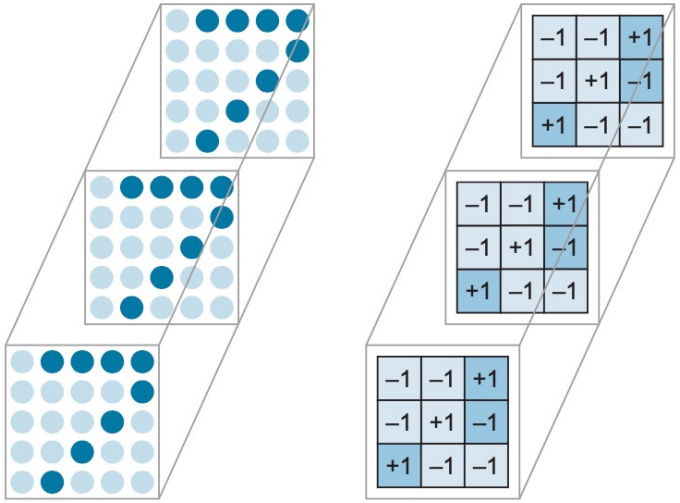
3. 합성곱 신경망

- 3.6 컬러 이미지의 합성곱 연산
  - 컴퓨터가 본 회색조 이미지는 너비, 높이를 가진 2차원 행렬
    - 각 픽셀 값은 원색의 강도, 밝기값을 의미
  - 컴퓨터가 본 컬러 이미지는 너비, 높이, 깊이를 가진 3차원 행렬
    - 2차원 행렬 3개 (빨간색, 녹색, 파란색에 해당)



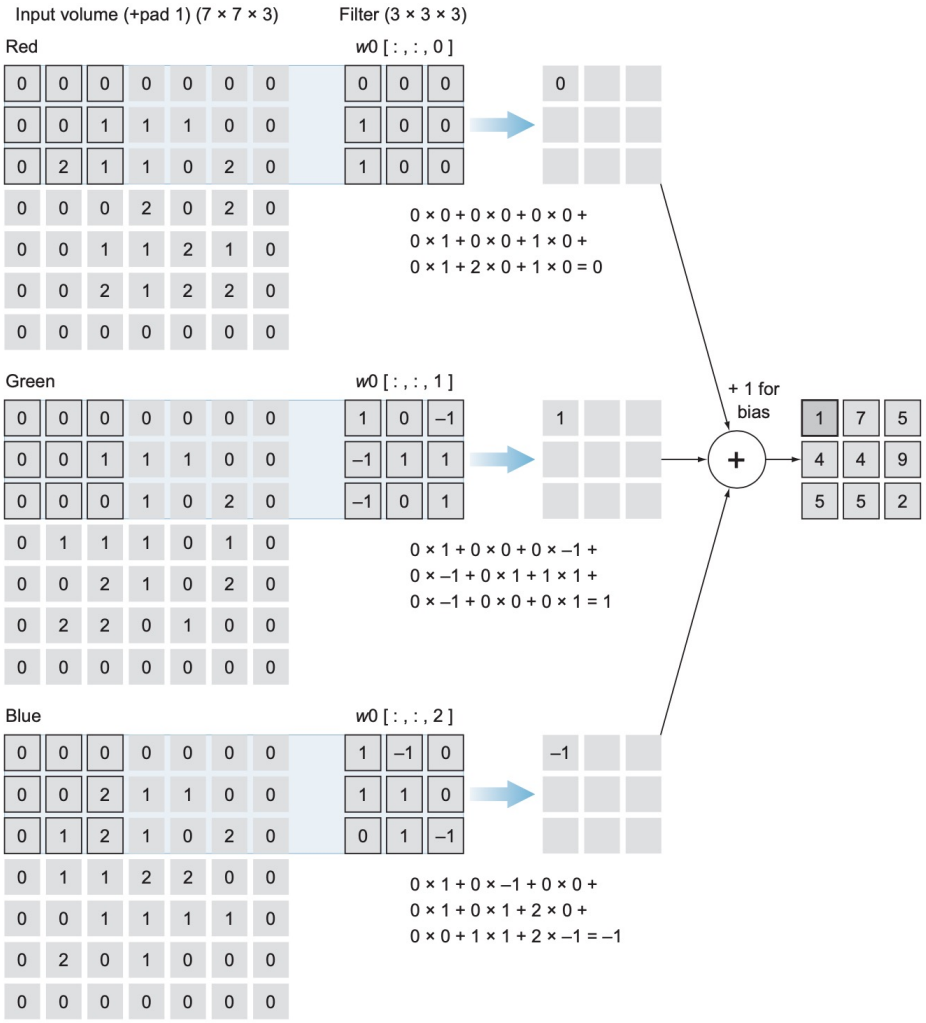
3. 합성곱 신경망

- **3.6.1 컬러 이미지를 대상으로 합성곱 연산하기**
  - 컬러 이미지의 합성곱은 회색조 이미지와 마찬가지로 합성곱 커널을 입력 이미지 위로 이동시키며 특징 맵을 계산
  - 컬러 이미지의 합성곱 연산은 합해야 할 항수가 세배로 늘어난 것만 제외하면 회색조 이미지의 합성곱과 동일
    - 색상 채널별로 별도의 필터를 갖음



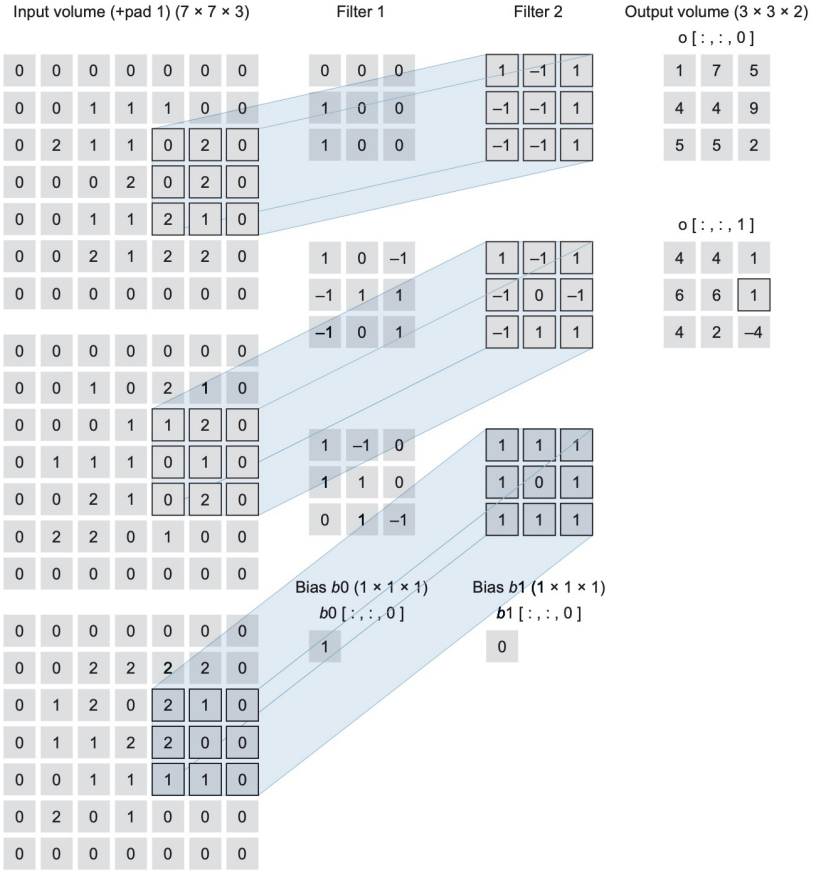
3. 합성곱 신경망

- 3.6.1 컬러 이미지를 대상으로 합성곱 연산하기

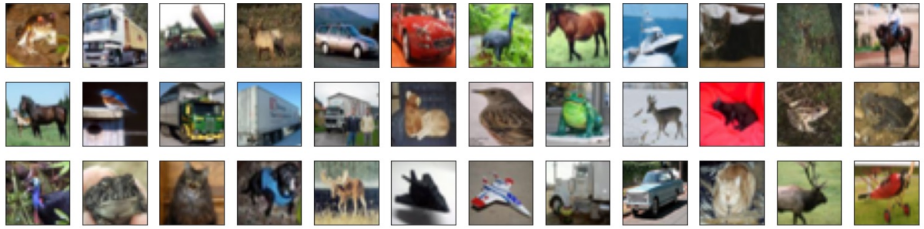


3. 합성곱 신경망

- **3.6.2 계산 복잡도의 변화**
  - 컬러 이미지는 회색조 이미지보다 공간 복잡도를 크게 증가시킴
  - 따라서 색상이 큰 영향을 못 미치는 경우에는 학습 데이터를 회색조 이미지로 변환



3. 합성곱 신경망



- **3.7 프로젝트: 컬러 이미지 분류 문제**
  - CNN 모델을 학습하여 CIFAR-10 데이터셋의 이미지를 분류하는 것
  - 학습데이터: CIFAR-10
    - 컴퓨터 비전 분야에서 널리 알려진 사물 인식 문제 데이터 셋
    - 8천만 장으로 구성된 '80 Million Tiny Image' 데이터셋의 서브셋
    - 32x32 크기의 컬러 이미지 6만장으로 클래스당 6천 장씩 10가지 클래스로 구성
  - 학습모델: AlexNet 간략버전
    - CNN 3개 + FC 2개
  - 테스트결과: 약 67% 정확도
  - 실습코드
    - CPU 를 활용한 코드 (한 에포크당 평균 23초)
      - https://www.kaggle.com/yukyungchoi/2022-dl-w3-project-cpu
    - GPU 를 활용한 코드 (한 에포크당 평균 5초)
      - https://www.kaggle.com/yukyungchoi/2022-dl-w3-project-gpu