

# Deep Learning Week 4

## **CNN & Transfer Learning**

---

Hanyang Artificial Intelligence Group

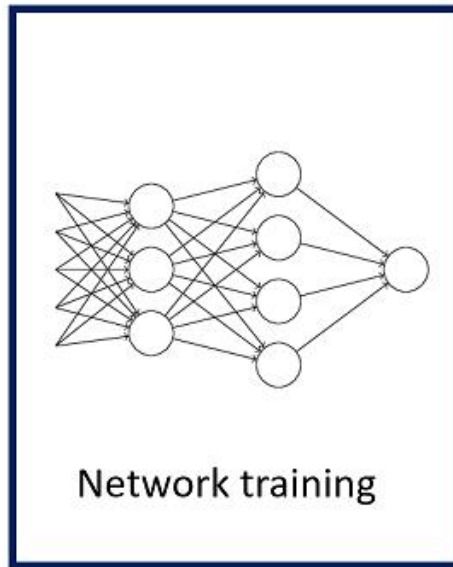


# Recall: MNIST 데이터셋 분류 문제

- MNIST 데이터셋은 28x28 사이즈의 손글씨 데이터셋
- 0~9까지 총 10개의 클래스를 가지고 있으며, 색상 채널이 없는 흑백 이미지



Data & Labels

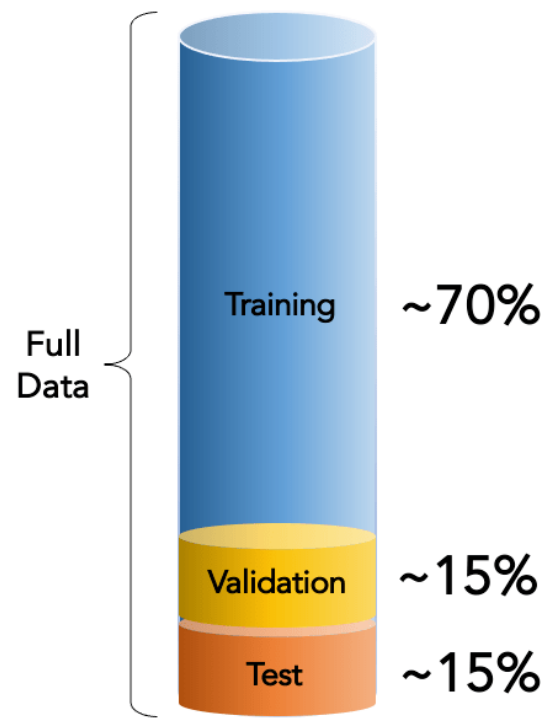
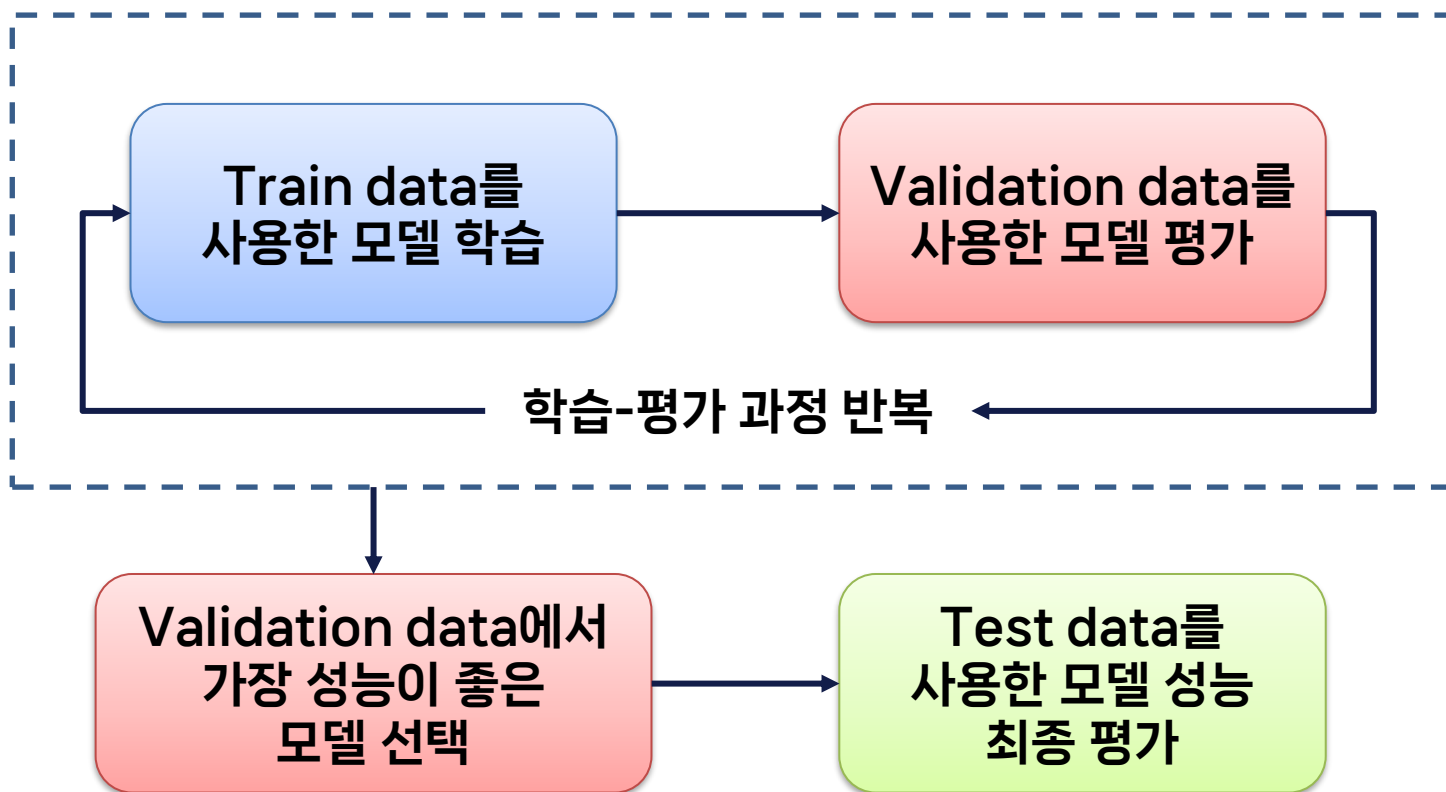


0  
1  
2  
3  
4  
5  
6  
7  
8  
9

# Recall: 학습 및 평가 데이터 분할

## Train? Validation? Test?

- 학습에 사용된 데이터를 모델의 성능을 평가하는데 사용하면 매우 위험!
- 원본 데이터를 분할하여 일부는 학습에 사용하고, 학습에 전혀 관여되지 않은 데이터로 모델의 성능 평가



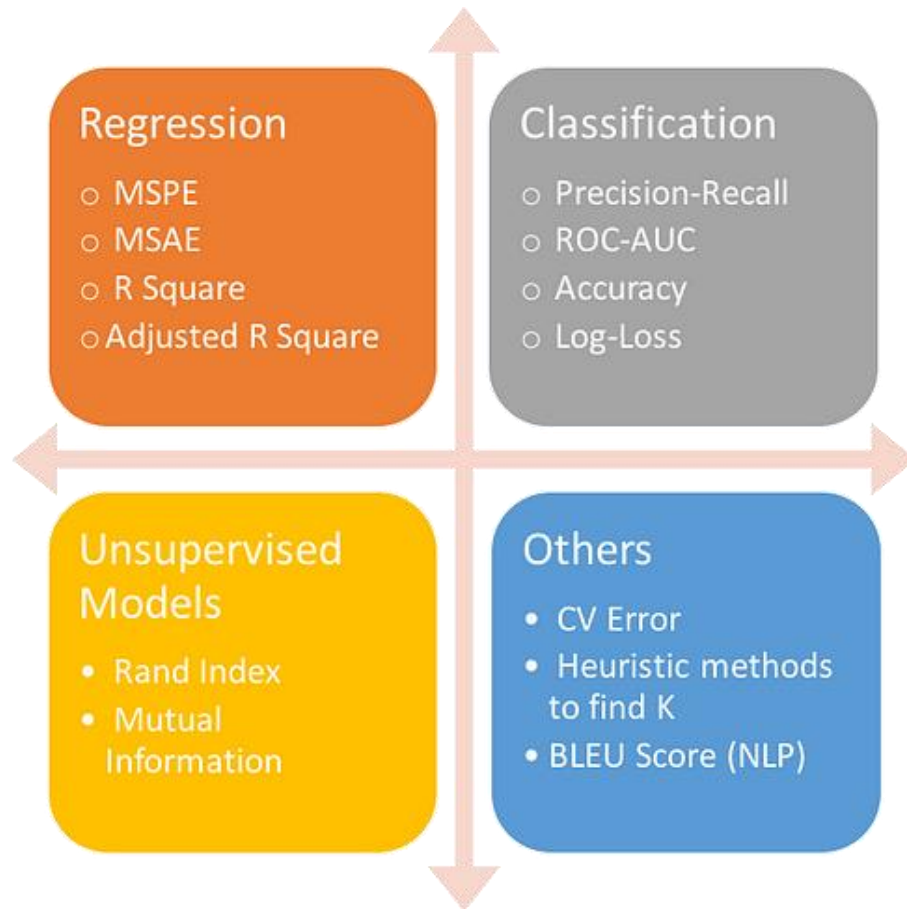
# Recall: 머신러닝 모델의 평가 metric

## Metric 이란?

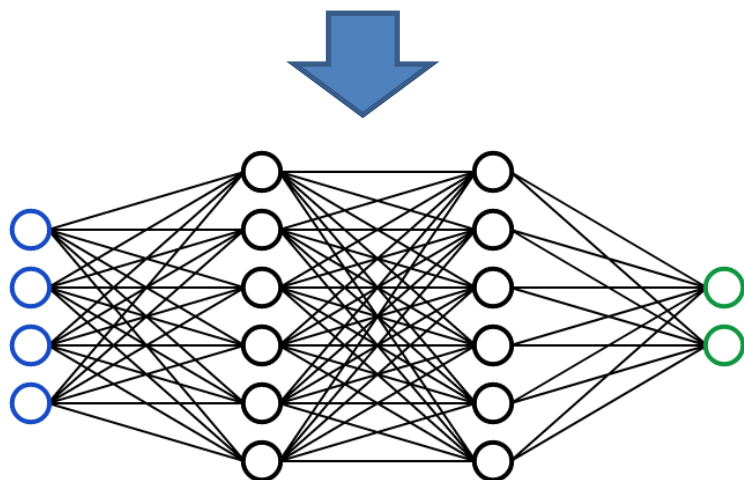
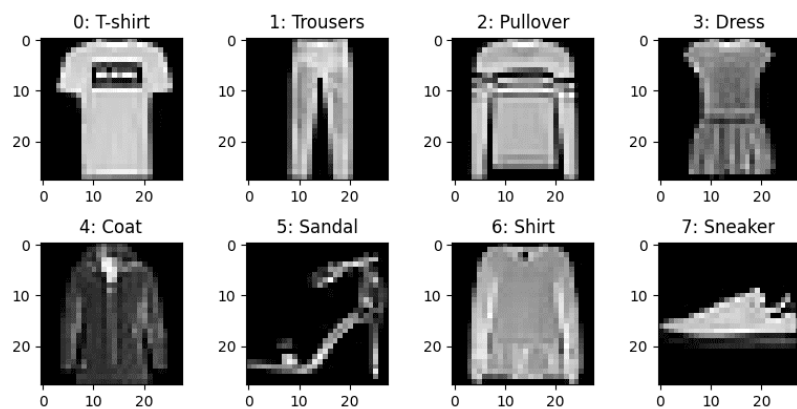
- 모델을 통해 해결하고자 하는 특정 task(분류, 회귀, ...)를 얼마나 잘 수행하는지 평가하기 위한 척도!
- 각각의 task에 적합한 metric들이 존재함
- Loss는 모델의 파라미터를 직접적으로 업데이트하기 위한 기준, metric은 모델이 직접적으로 수행하는 task에 대한 성능을 파악하기 위해 사용
- Metric이 곧 Loss function이거나, 복합적으로 평가하기 위해 여러 metric을 사용하는 경우도 있음
- 분류 문제의 경우

Loss: Binary-Crossentropy

Metric: Accuracy, F1 score

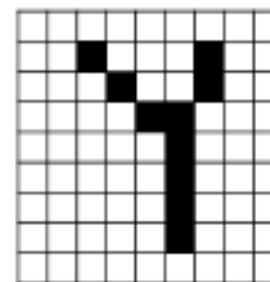
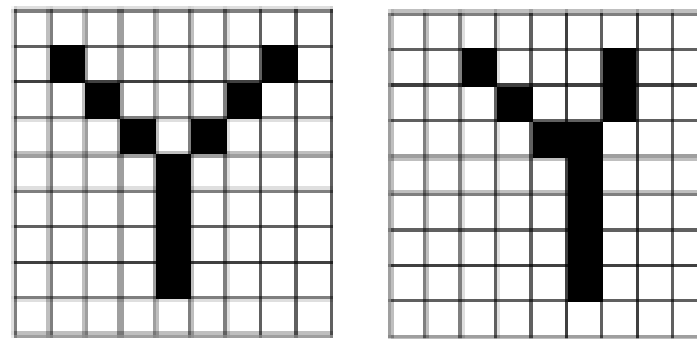
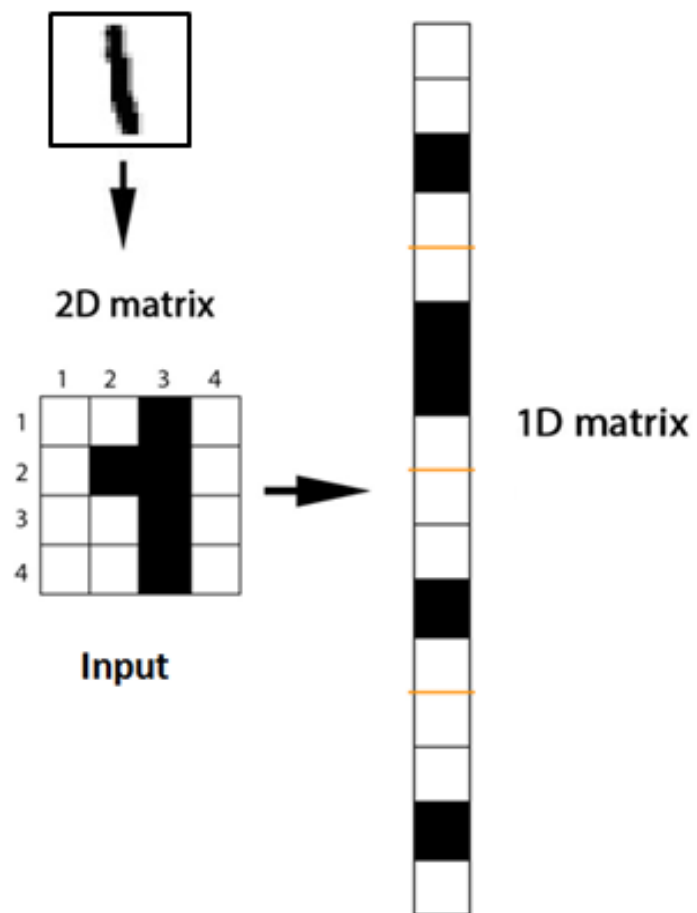


# Fashion MNIST



# Recall: 데이터 전처리 과정

- 2차원 행렬 데이터를 1차원 벡터로 변환(flatten)



↓ 변환

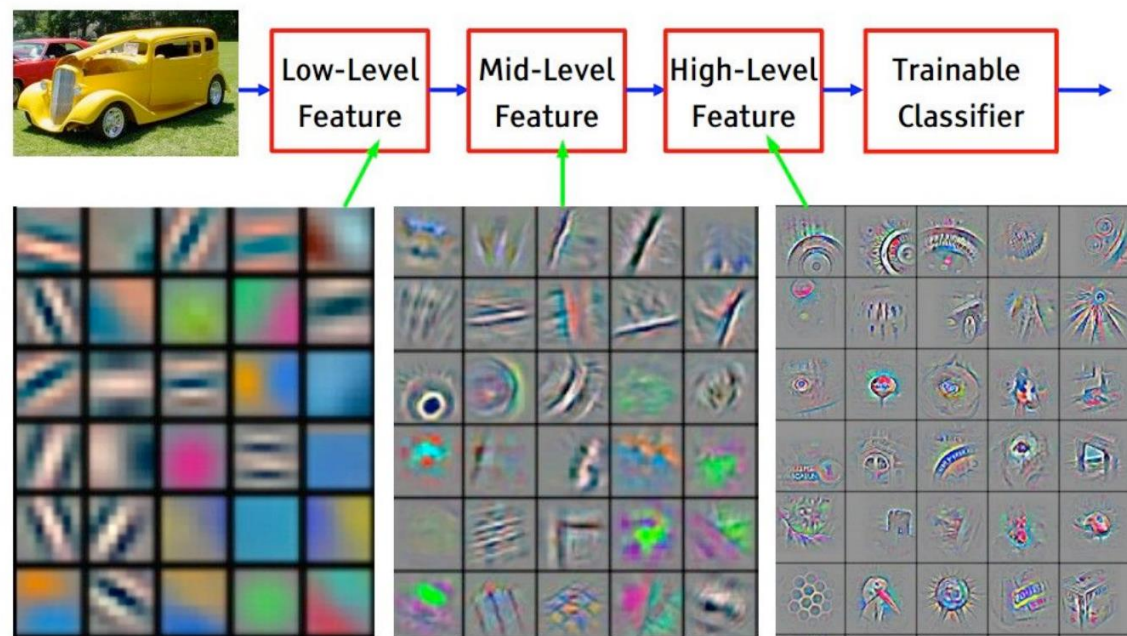
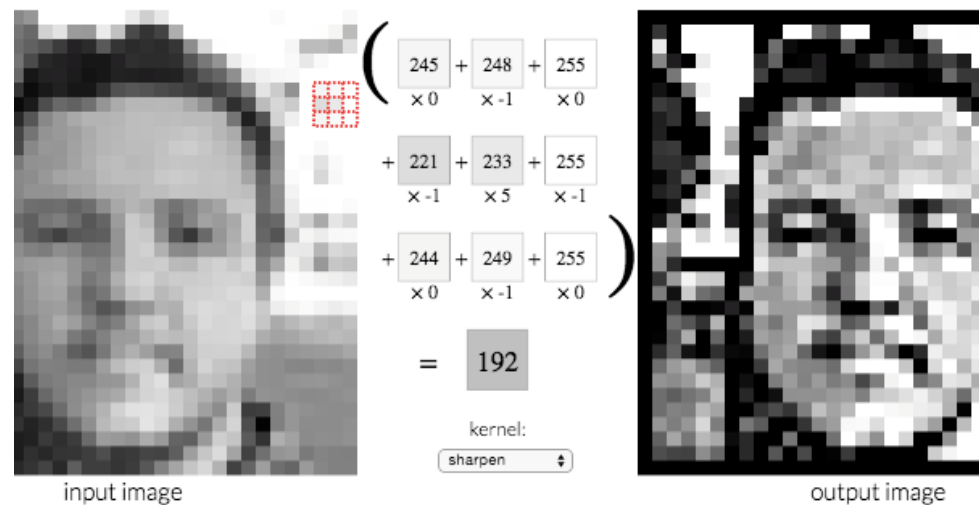




# Convolution Neural Network 소개

## CNN 이란?

- Convolution 연산을 수행하는 레이어로 구성된 네트워크
- 이미지 데이터에서 인접한 픽셀들 사이의 정보를 바탕으로 유용한 feature를 추출해낼 수 있음
- Image classification, semantic segmentation, object detection, image generation, OCR 등 다양한 컴퓨터 비전 task에서 좋은 성능을 보여줌

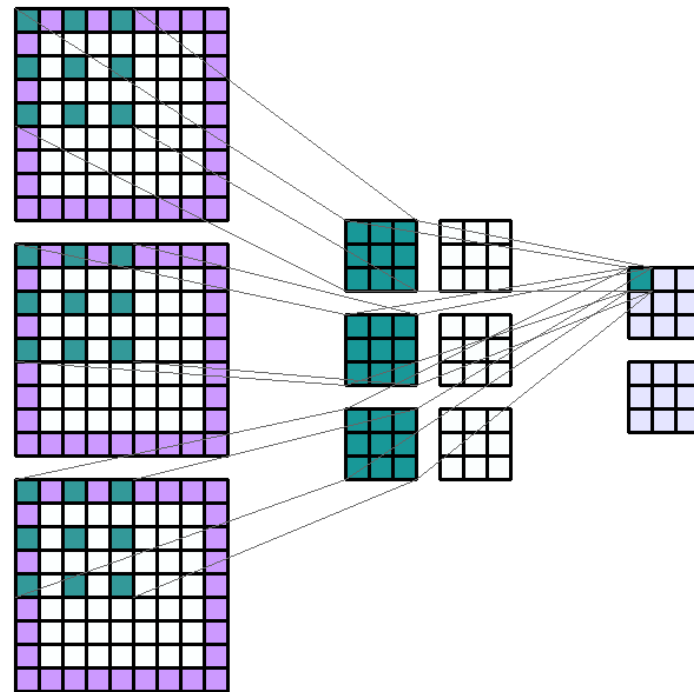
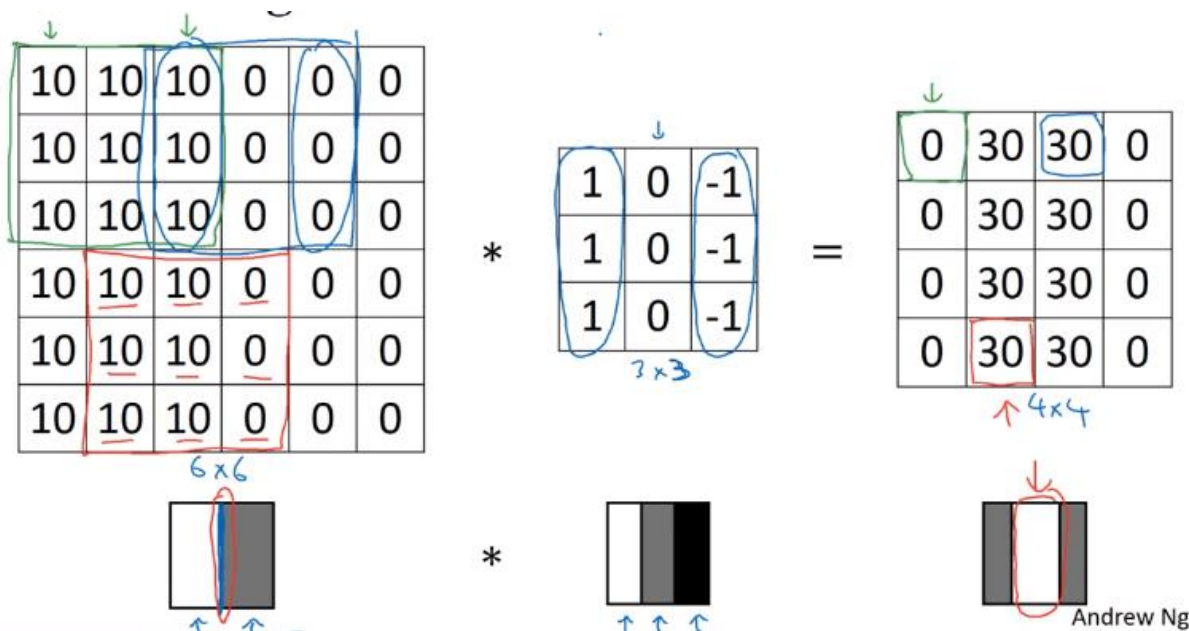


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

# Convolution 연산

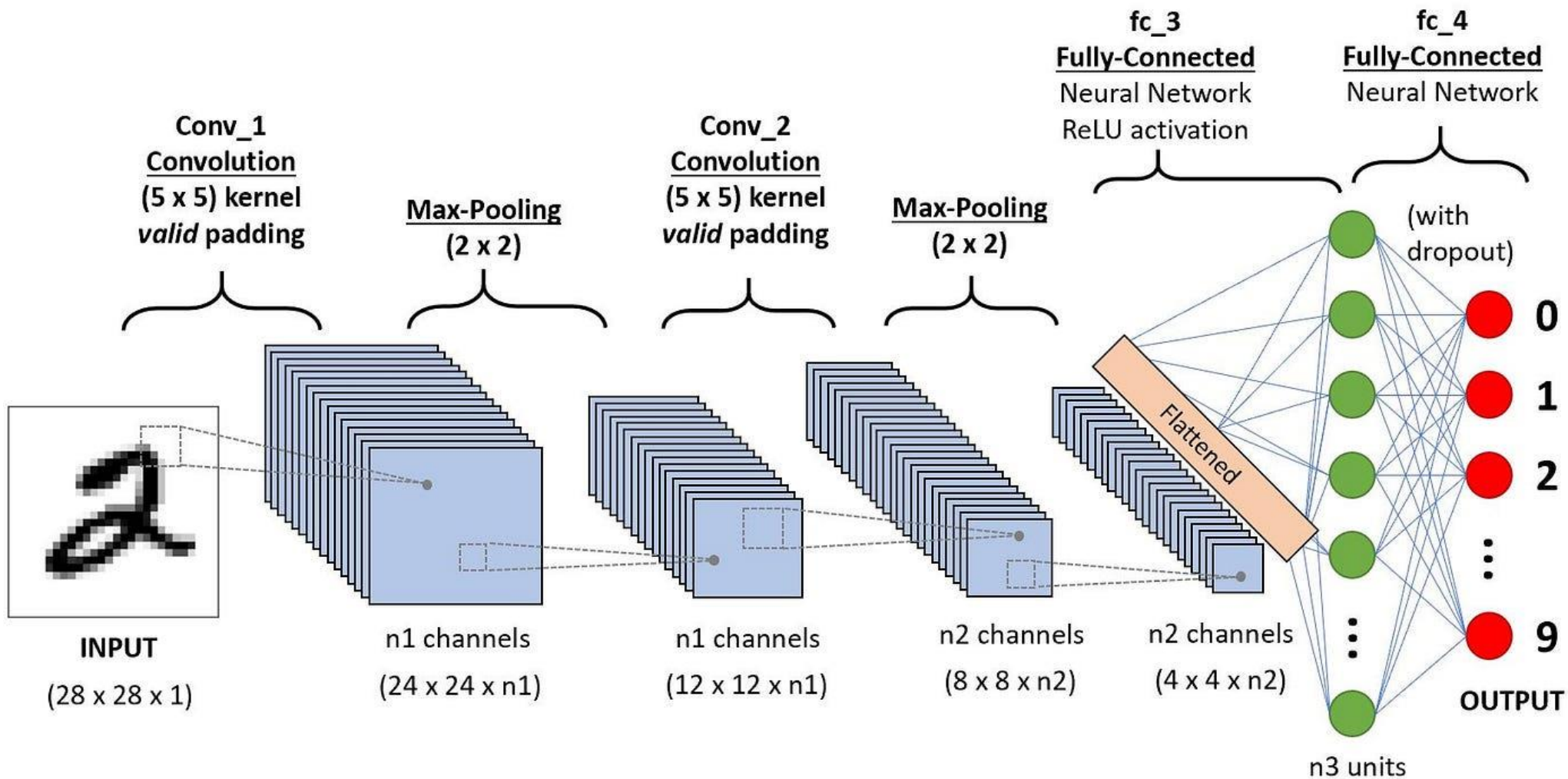
## Convolution 이란?

- 입력 데이터에 존재하는 특정한 패턴이 존재하는지 확인할 수 있는 필터를 입력과 곱하는 연산
- 전통적인 컴퓨터 비전 문제 해결 방식에서는 좋은 feature를 찾기 위한 필터를 사람이 직접 설계
- 딥 러닝을 활용한 CNN 학습 과정에서는 데이터의 feature를 추출해낼 수 있는 좋은 convolution filter를 자동으로 찾게 됨





# MNIST 분류를 위한 CNN의 구조

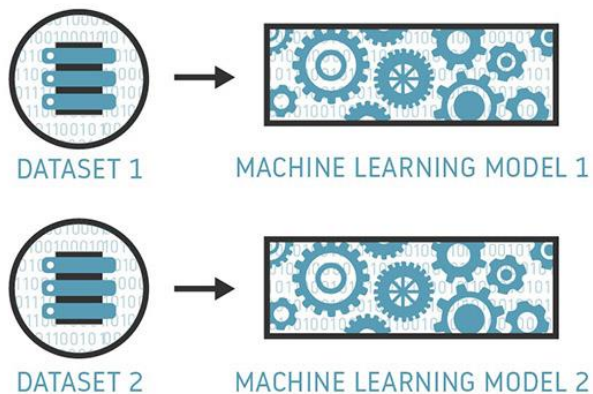


# Transfer learning이란?

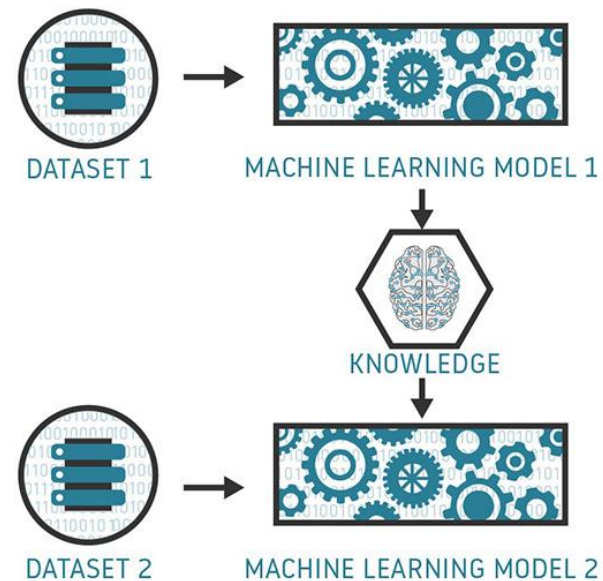
## 이미 학습된 모델의 능력을 재사용한다!

- 유사한 task에 대해 학습된 적 있는 모델의 능력을 활용하여 새로운 task를 해결하도록 적용하는 방법
- 맨땅에서 모델을 학습시키기보다, 이미 다른 이미지 분류를 잘 수행하는 모델을 가져와 Fashion MNIST를 분류하도록 훈련시키면 더 나은 분류성능을 기대할 수 있음

TRADITIONAL MACHINE LEARNING

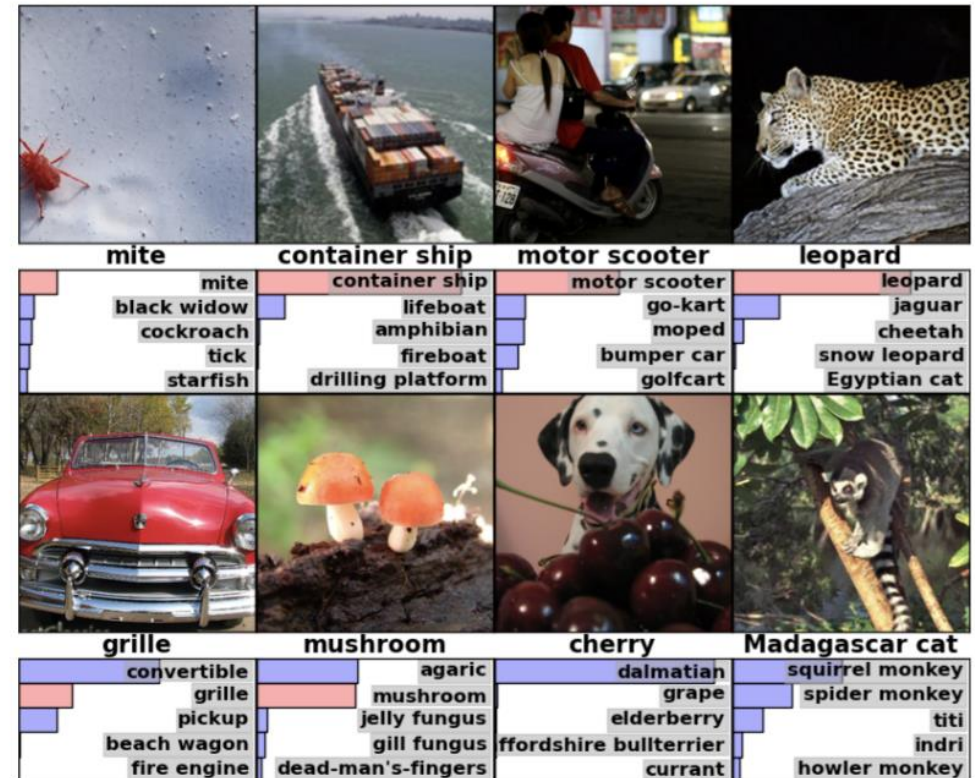
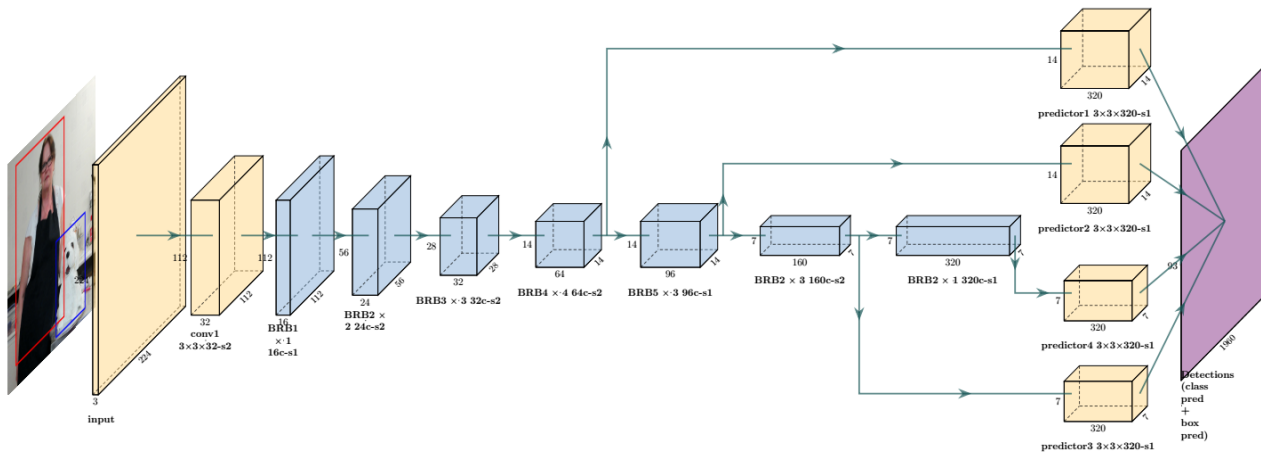


TRANSFER LEARNING



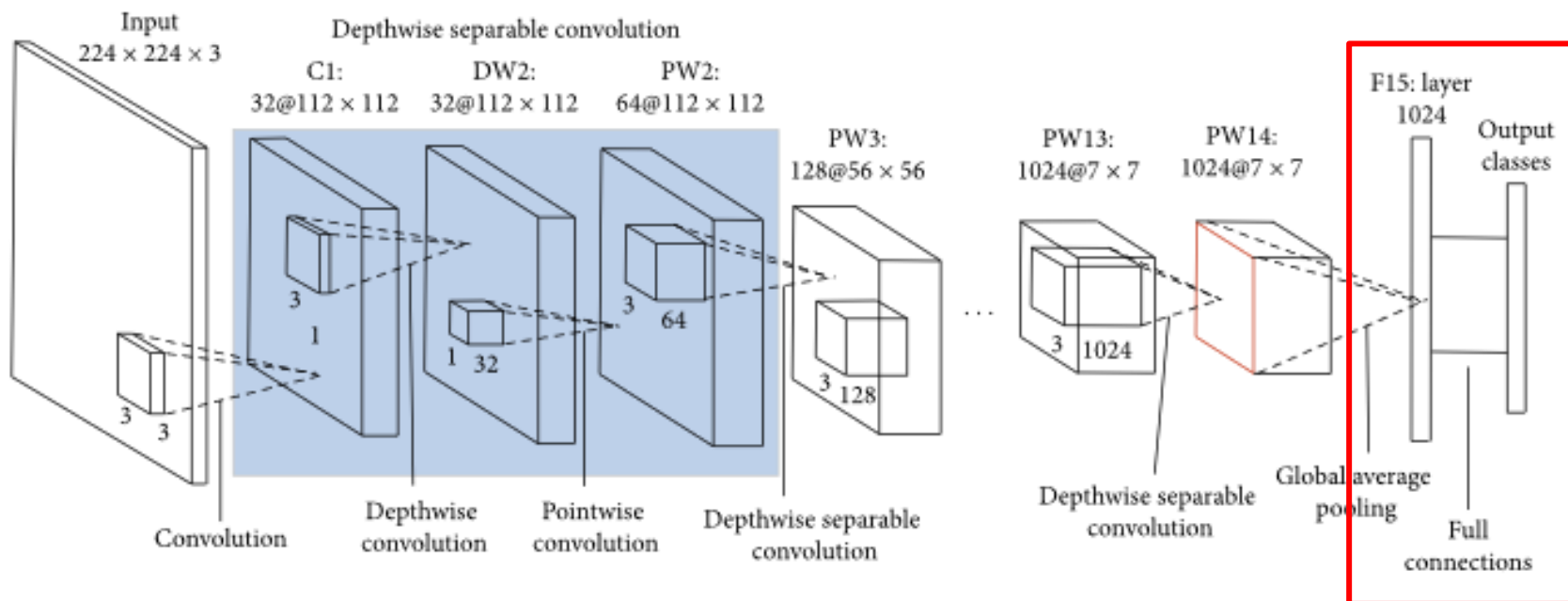
# MobileNet

- <https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet>
- 모바일 기기처럼 제한된 환경에서도 효율적으로 이미지 분류를 수행할 수 있도록 학습된 모델
- 100만개가 넘는 이미지 데이터를 1000개의 카테고리로 라벨링한 ImageNet 데이터셋 분류를 수행하도록 학습됨



# MobileNet으로 Fashion MNIST 분류 모델 만들기

- 이미 엄청나게 많은 이미지의 특성을 학습한 MobileNet을 Fashion MNIST 분류를 수행하도록 학습
- Feature를 추출해주는 부분은 가만히 두고, softmax classification을 수행하는 output layer의 형태를 수정하여 MNIST 분류를 수행할 수 있도록 변경한 후 학습 진행
- 유사한 task를 수행하도록 학습된 능력이 다른 task에 전이되어 도움을 줌: **transfer learning**



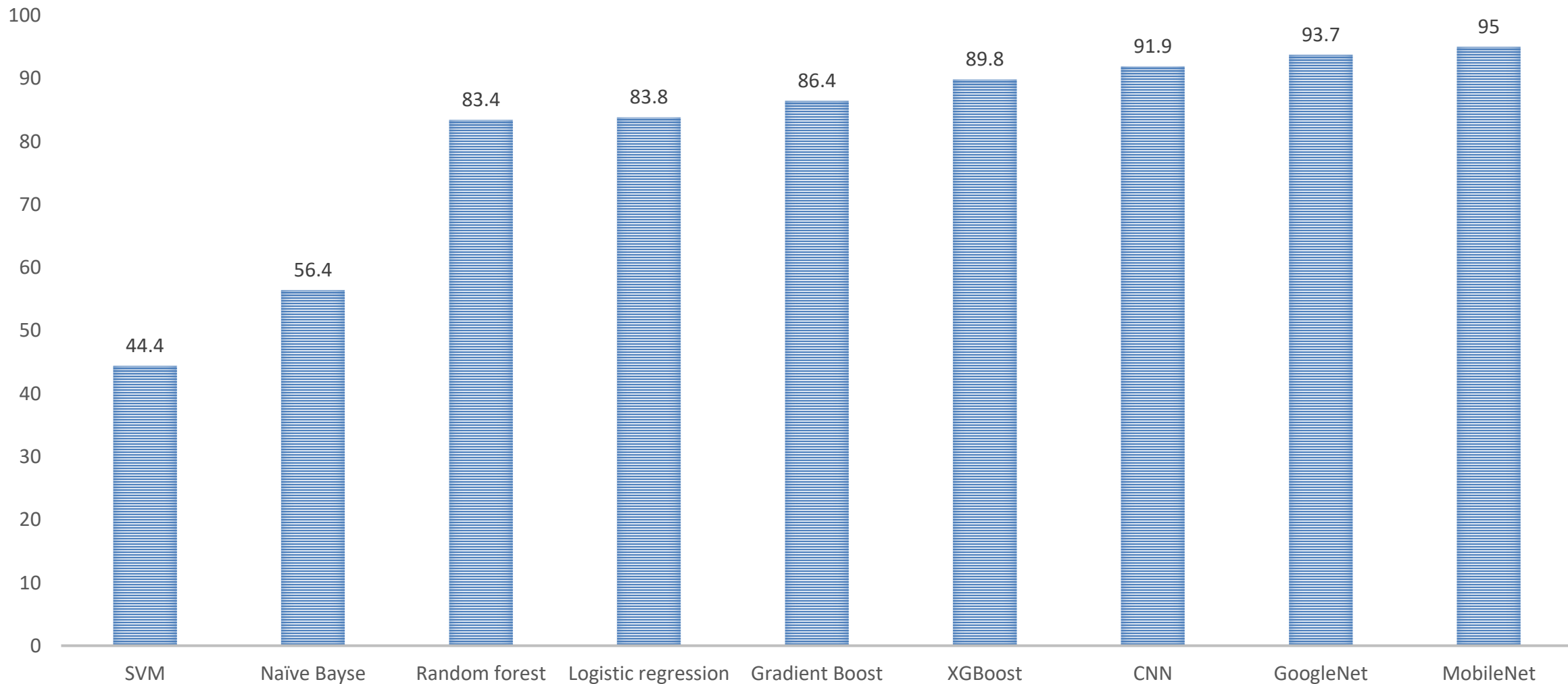
## Output layer

- 기존:  $(1024 * 1000) \rightarrow \text{softmax}$   
 $\rightarrow$  1000개의 이미지 종류 class에 대한 확률
- MNIST:  $(1024 * 10) \rightarrow \text{softmax}$   
 $\rightarrow$  10개의 숫자 class에 대한 확률

# 다양한 모델들의 Fashion MNIST 분류 성능

평가 기준: 분류 정확도(%)

ACCURACY(%)





# SOTA(State-Of-The-Art)

## SOTA란?

특정 task를 수행하기 위한 여러 모델이나 방법론 중,  
현재 가장 성능이 뛰어난 것!

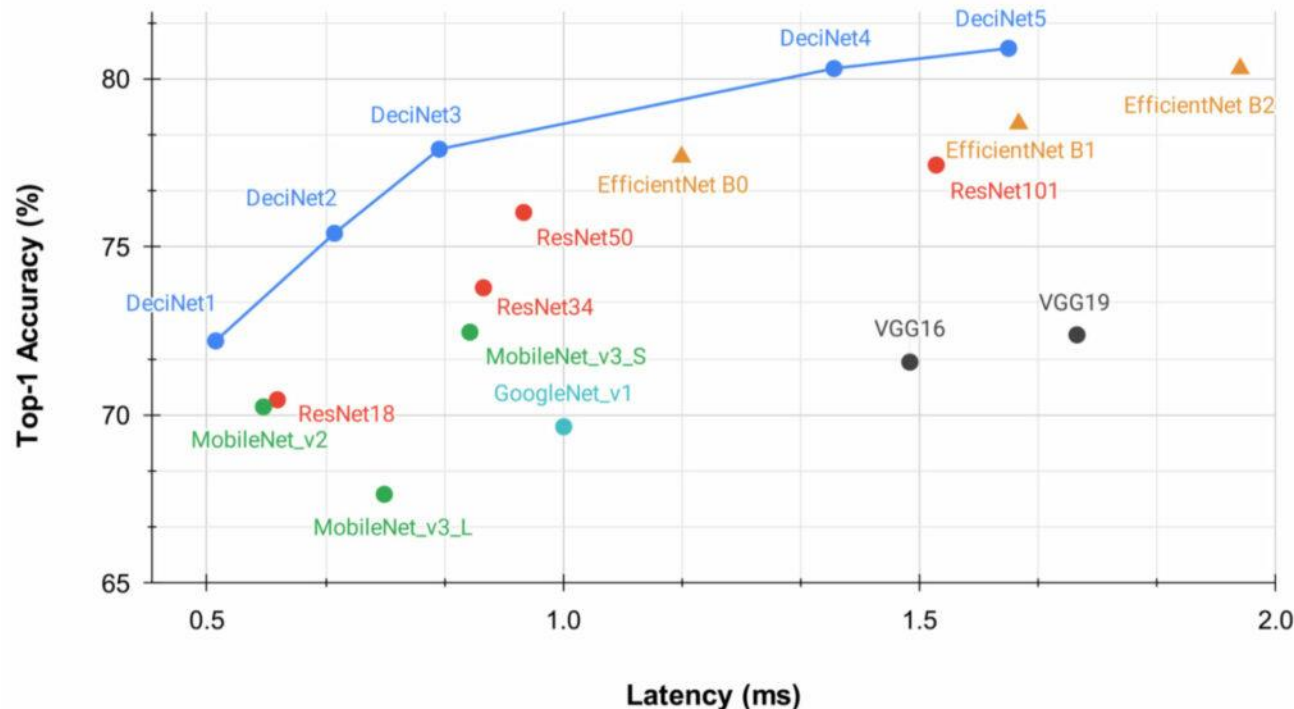
## SOTA를 알아야 하는 이유

딥 러닝은 완성된 기술이 아니며, 끊임없이 변화하고,  
연구자들과 엔지니어들이 늘 새로운 시도를 하기 때문!

가장 뛰어난 알고리즘도 하루아침에 구식이 되어버릴  
수 있는 딥러닝 생태계에서, 최신 연구 결과와  
기술들을 follow-up 하는 것이 중요하다!

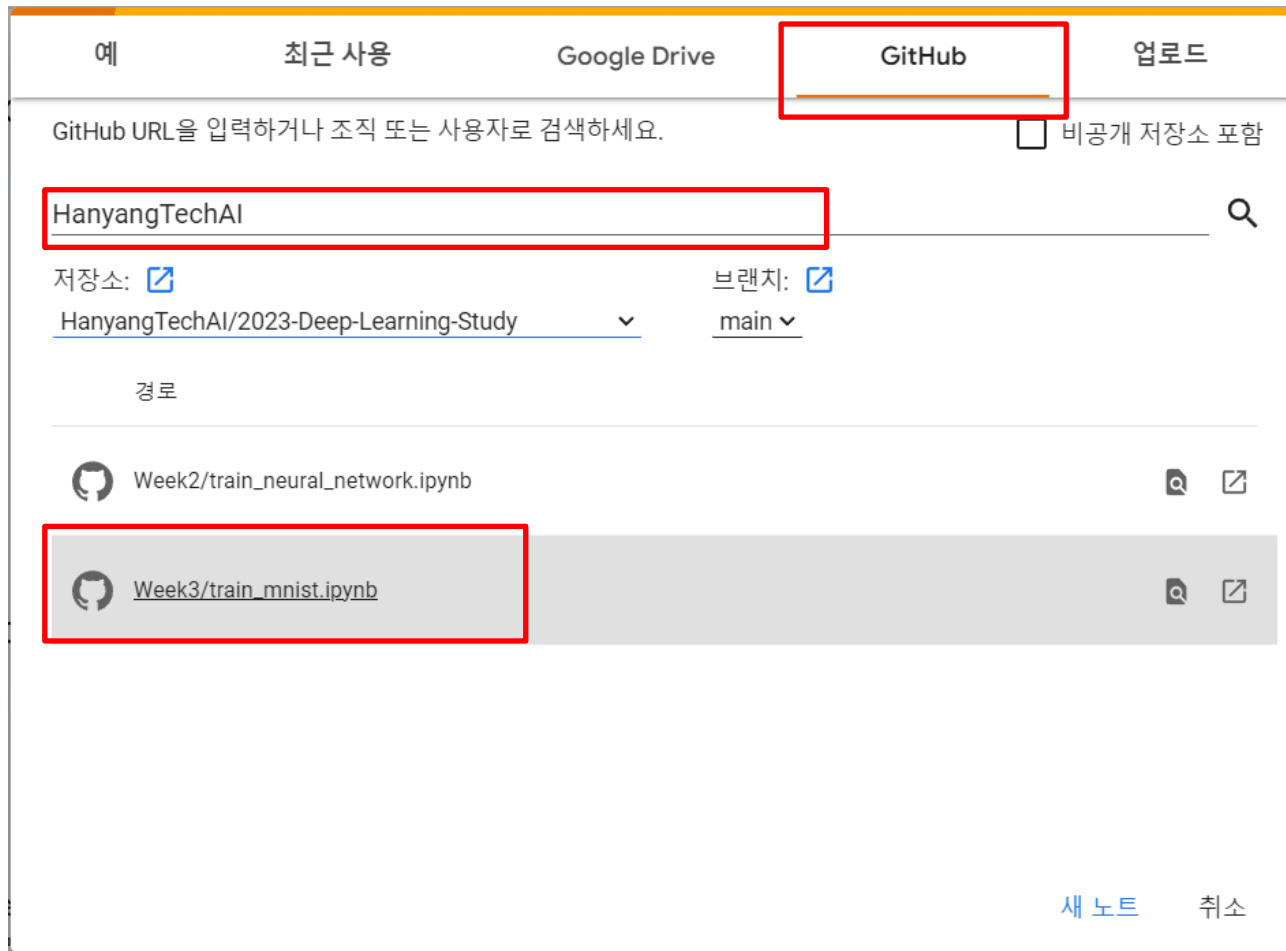
**참고:** <https://paperswithcode.com/>

각종 task에서 SOTA를 달성한 ML 알고리즘의 논문,  
소스코드 등을 정리한 사이트





# Colab에서 github에 업로드된 노트북 불러오기



# Colab에서 GPU 가속 사용하기



train\_fashion\_mnist.ipynb

파일 수정 보기 삽입 런타임 도구 도움말

Colab interface showing the 'Run' menu. The 'Run' menu is open, displaying various execution options. The 'Run' menu item is highlighted with a red box.

- 모두 실행 (Ctrl+F9)
- 이전 셀 실행 (Ctrl+F8)
- 초점이 맞춰진 셀 실행 (Ctrl+Enter)
- 선택항목 실행 (Ctrl+Shift+Enter)
- 이후 셀 실행 (Ctrl+F10)
- 실행 중단 (Ctrl+M |)
- 런타임 다시 시작 (Ctrl+M .)
- 다시 시작 및 모두 실행
- 런타임 연결 해제 및 삭제
- 런타임 유형 변경**
- 세션 관리
- 리소스 보기
- 런타임 로그 보기

노트 설정 (Note Settings) dialog box.

**하드웨어 가속기** (Hardware Accelerator) is highlighted with a red box. The dropdown menu shows 'GPU' selected.

GPU 등급 (GPU Tier) is set to '스탠다드' (Standard).

프리미엄 GPU를 이용하시겠어요? [추가 컴퓨팅 단위 구매](#)

☐ 이 노트를 저장할 때 코드 셀 출력 생략

취소 저장

## 네 번째 과제: Transfer Learning의 효과 분석

### Step 1. 예시 노트북을 실행해보며 CNN 모델로 Fashion MNIST 튜닝하기

- Fashion MNIST 데이터를 분류하기 위한 CNN 모델을 정의하고, 학습 및 평가 과정을 살펴봅시다.

### Step 2. MobileNet의 Fashion MNIST 분류 문제 transfer learning

- 다양한 이미지의 특성을 추출하고 카테고리를 분류할 수 있는 MobileNet의 능력을 이용하여, Fashion MNIST를 분류하는 모델을 학습시켜 봅시다.
- MobileNet에 사전 학습된 파라미터를 불러오지 않고, 이전처럼 빈 모델로 같은 task를 학습시켰을 때 어떻게 평가 결과가 달라지는지(test acc, loss 등)를 비교해 주세요.
- 노트북([https://colab.research.google.com/github/HanyangTechAI/2023-Deep-Learning-Study/blob/main/Week4/fashion\\_mnist\\_cnn.ipynb](https://colab.research.google.com/github/HanyangTechAI/2023-Deep-Learning-Study/blob/main/Week4/fashion_mnist_cnn.ipynb))을 바탕으로 조별로 자유롭게 토의해서, 결과를 Github 이슈(<https://github.com/HanyangTechAI/2023-Deep-Learning-Study/issues/5>)에 업로드해주세요!

# With HAI, Fly High

---

Hanyang Artificial  
Intelligence