

## [CUAI] CS231n 스터디 1팀

2025.03.18

발표자 : 황민아

## 스터디원 소개 및 만남 인증



### Study Info

- 스터디 시작일 : 2025.03.17(월) ~ 6/2(월)
- 스터디 장소 : 중앙대학교 스터디룸
- 스터디 시간 : 매주 월요일 1회 18:00~20:00
- 스터디 구성원 : 황민아, 조한서, 김승미, 김성민

## 목차

1. 스터디 진행 방식
2. 이번 주 학습 내용

# 스터디 진행 방식

[1주차] 3/17

1. Introduction to CNN for Visual Recognition
2. Image Classification
3. Loss Functions and Optimization

[2주차] 3/24

4. Introduction to NN
5. CNN
6. Training Neural Networks, Part 1

[3주차] 3/31

7. Training Neural Networks, Part 2
8. Deep Learning Software
9. CNN Architectures

중간고사 4/22~28

[4주차] 5/5

10. RNN
11. Detection and Segmentation

[5주차] 5/12

11. Detection and Segmentation
12. Visualizing and Understanding

[6주차] 5/19

13. Generative Models
14. Reinforcement Learning

[7주차] 5/26

15. Efficient Methods and Hardware for Deep Learning

[8주차] 6/2

16. Adversarial Examples and Adversarial Training

기말고사 6/17~23

+ 해당 주차에 맞는 assignments 진행 예정

Assignment #1: Image Classification, kNN, SVM, Softmax, Fully Connected Neural Network

Assignment #2: Fully Connected and Convolutional Nets, Batch Normalization, Dropout, Pytorch & Network Visualization

Assignment #3: Network Visualization, Image Captioning with RNNs and Transformers, Generative Adversarial Networks, Self-Supervised Contrastive Learning

# 1) Introduction to Neural Network

Computer Vision의 최종 목표 ? 사람처럼 볼 수 있는 기계를 만들자!

## Visual recognition problems

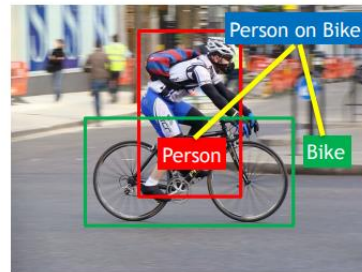
: image classification  
( object detection, image captioning.. )



This image is licensed under CC BY-NC-SA 2.0; changes made

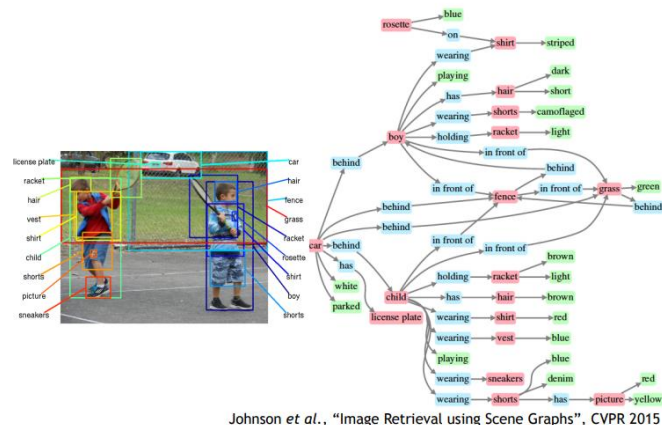


- Object detection
- Action classification
- Image captioning
- ...



# LEO

## Computer Vision의 최종 목표 ? 사람처럼 볼 수 있는 기계를 만들자



Johnson *et al.*, "Image Retrieval using Scene Graphs", CVPR 2015

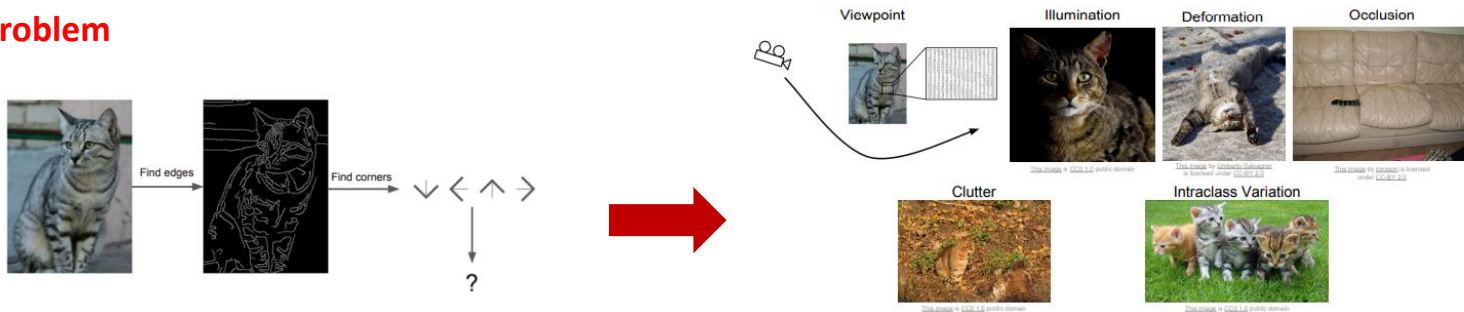
## Convolutional Neural Networks (CNN) : Important tool for object recognition !

## The quest for visual intelligence goes far beyond object recognition...

## 2) Image Classification

A core task in Computer Vision = classification

### Problem



Robust 하지 않다는 단점이 존재하여 분류가 잘 이루어지지 않는다.

### Solution : Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

```
def train(images, labels):  
    # Machine learning!  
    return model  
  
def predict(model, test_images):  
    # Use model to predict labels  
    return test_labels
```

Example training set



-> 객체의 특징을 규정하지 않고, 다양한 사진들과 label을 수집하고, 이를 이용해 모델을 학습하여 사진을 새롭게 분류하도록 함.

## 2) Image Classification

### 1. Nearest Neighbor

: 입력 받은 데이터를 저장한 다음 새로운 입력 데이터가 들어오면, 기존 데이터에서 비교하여 가장 유사한 데이터를 찾아내는 방식.

### 2. K-Nearest Neighbor : 다수결 투표

#### Distance Metric

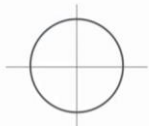
L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$

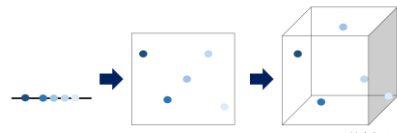


: Distance metric으로 이미지를 비교하고 입력과 비슷한 이미지를 찾아내도록 함.

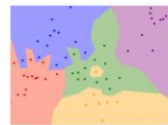


**BUT 이미지 분류에 잘 쓰지 않는다.**

고차원 데이터에서 비효율적이다. (차원의 저주)  
계산 비용이 너무 크고, 일반화 능력이 부족.



Generalize



K = 1



K = 3



K = 5

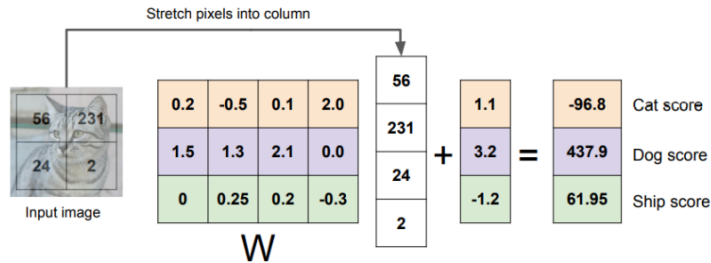
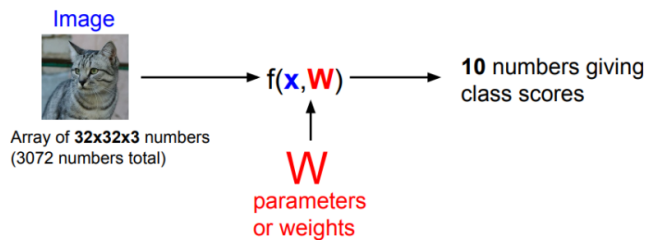
(K가 증가할 수록 decision boundary 가 부드러워지는 것을 확인 가능하다.)



## 2) Image Classification

### 3. Linear Classification

- NN(Neural Network)과 CNN의 기반이 되는 알고리즘
- parametric model의 기초
- 이미지 데이터(x)에 가중치(w)를 곱하고, 편향값(b)을 더하는 방식



이미지를 숫자로 변환한 다음, 그 숫자에 특정 가중치(w)를 곱해서 점수를 계산.

$$f(x, W) = Wx + b$$

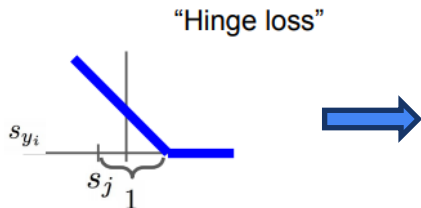
w 가중치 설정이 중요 !

### 3) Loss function and Optimization

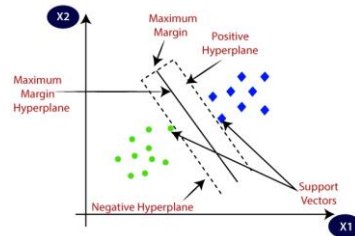
Loss를 줄이기 위해 가중치(w)를 조정

#### 1. Multiclass SVM loss

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$
$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$



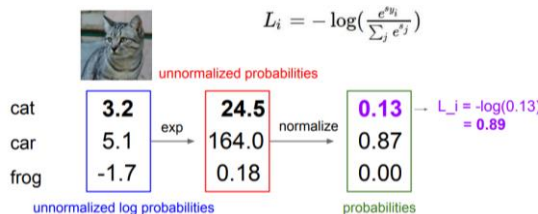
정답 클래스의 점수가 다른 클래스보다 최소 1 이상 크도록 학습합니다



이를 통해 결정 경계(Decision Boundary)를 최대한 멀리 확보하여 일반화 성능을 높임.

#### 2. Cross Entropy loss - Softmax Classifier ( Multinomial logistic regression )

•SVM과 달리 점수를 확률 분포로 변환하여 학습하는 방식. 분포 차이를 통해 loss를 계산합니다.



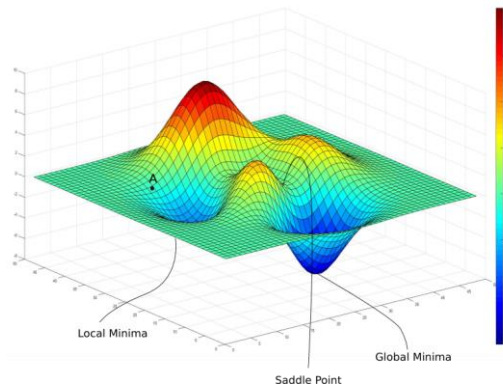
### 3) Loss function and Optimization

그렇다면, 어떻게 Optimize 할 것인가?

loss를 최소화 시킬 수 있는 weight를 찾는 것

**Gradient Descent**

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$



**Stochastic gradient descent (SGD)**

: 모든 데이터를 한 번에 처리하는 것은 비효율적이므로 일부 데이터를 샘플링해 학습

감사합니다