

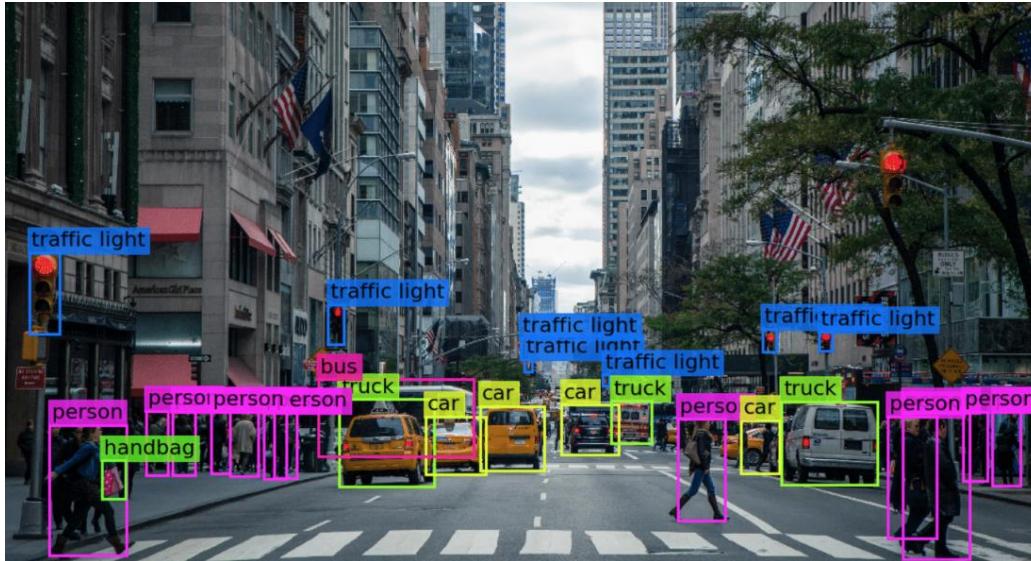
12주. 인공지능과 컴퓨터 시각

한림대학교 소프트웨어융합대학

시각이란?



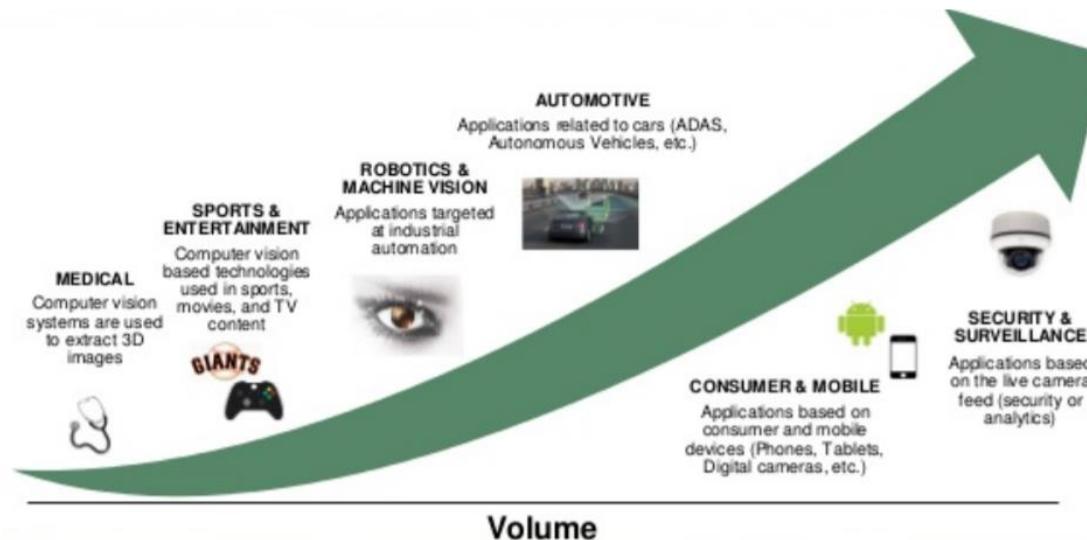
컴퓨터의 시각



컴퓨터 시각 (computer vision)

- 물체 탐지
- 물체 분류
- 장면 이해
- 물체 추적
- 사람 자세 추정
-

컴퓨터 비전의 응용



Slide source: World Capital Partners, 2017

- 의료 영역
- 게임과 엔터테인먼트
- 로봇 공학
- 자율자동차
- 모바일 환경
- 가상 증강현실
- 보안과 감시카메라
- 영화 및 그래픽

이미지 분류 소개

■ 컴퓨터 시각의 핵심



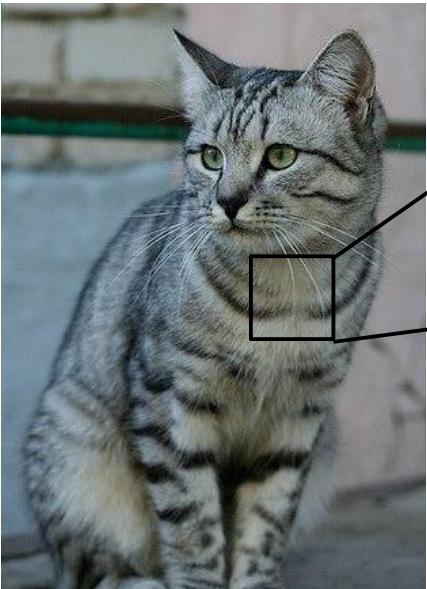
This image by Nikita is lic
ensed under CC-BY2.0

(assume given a set of labels)
{dog, cat, truck, plane, ...}



cat

The Problem: 의미와 표현의 차이



```
[ [105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
[ 91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
[ 76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
[ 99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
[106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
[114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
[133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
[128 137 144 140 109 95 86 78 62 65 63 63 60 73 86 101]
[125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
[127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
[115 114 109 123 158 148 131 118 113 109 100 92 74 65 72 78]
[ 89 93 90 97 108 147 131 118 113 114 113 109 106 95 77 80]
[ 63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
[ 62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
[ 63 65 75 88 89 71 62 81 126 138 135 105 81 98 110 118]
[ 87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
[118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
[164 146 112 80 82 120 124 104 76 48 45 66 88 101 102 109]
[157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
[130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
[128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
[123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
[122 121 182 80 82 86 94 117 145 148 153 102 58 78 92 107]
[122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```

컴퓨터의 관점

This image by Nikita is licensed under CC-BY 2.0

The Problem: 의미와 표현의 차이

```
[[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
 [ 91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
 [ 76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
 [ 99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
 [106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
 [114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
 [133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
 [128 137 144 140 109 95 86 70 62 65 63 63 60 73 86 101]
 [125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
 [127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
 [115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
 [ 89 93 90 97 108 147 131 118 113 114 113 109 106 95 77 80]
 [ 63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
 [ 62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
 [ 63 65 75 88 89 71 62 81 120 138 135 105 81 98 110 118]
 [ 87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
 [118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
 [164 146 112 80 82 120 124 104 76 48 45 66 88 101 102 109]
 [157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
 [130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
 [128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
 [123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
 [122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
 [122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```

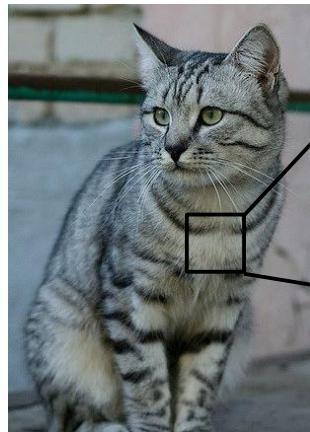
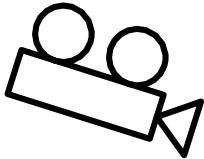


고양이

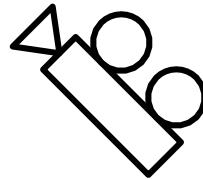
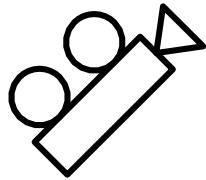
컴퓨터의 관점

Challenges: 관점에 따른 차이

- 보는 위치에 따라 숫자가 크게 달라진다



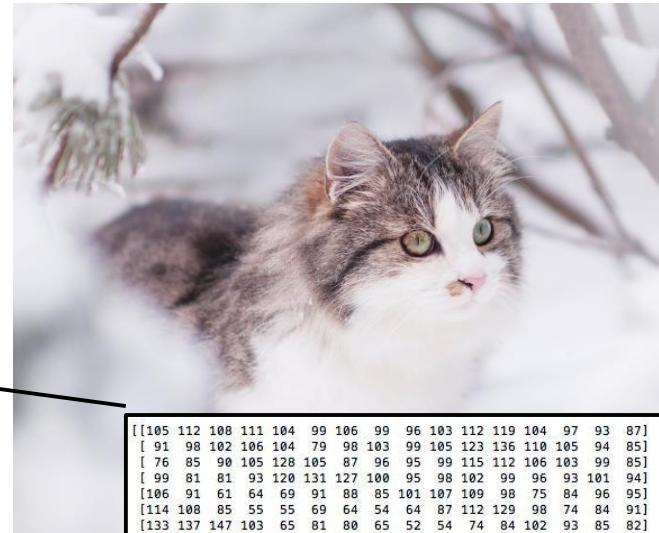
```
[1105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
[1 91 108 106 103 112 119 104 97 93 87]
[1 76 85 90 105 128 105 97 96 95 99 115 122 106 93 99 85]
[1 99 81 81 93 120 131 127 108 95 98 102 99 96 93 101 94]
[106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
[114 108 85 55 58 69 64 54 64 87 112 128 98 74 84 91]
[123 133 144 140 149 159 155 152 150 147 144 141 138 135 132 130 127]
[128 137 144 149 109 95 86 78 62 65 63 63 68 73 86 101]
[125 133 148 137 119 121 117 94 65 79 80 65 65 54 64 72 98]
[127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
[115 120 130 138 130 128 126 131 111 96 89 75 61 64 72 78]
[1 90 33 96 97 108 147 118 113 114 113 109 99 95 96 103]
[1 63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
[1 62 65 82 89 78 71 88 101 124 126 119 101 107 111 131 119]
[1 63 65 75 88 88 71 62 81 120 138 135 105 81 90 118 118]
[1 89 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
[118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
[164 146 112 88 82 128 124 104 76 48 45 66 88 101 102 109]
[157 170 157 120 92 86 114 132 112 97 69 55 78 82 99 94]
[130 132 132 130 139 130 139 130 130 130 130 130 130 130 130 130]
[128 132 96 117 98 144 128 115 104 107 102 93 87 91 79 79]
[123 107 96 86 83 112 153 149 122 109 104 75 88 107 112 99]
[122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
[122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84])
```



Challenges: 배경과 섞임



This image is CC0 1.0 public domain



```
[[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
 [ 91 98 102 106 104 79 98 103 99 105 123 136 118 105 94 85]
 [ 76 85 92 105 128 105 87 96 95 99 115 112 106 103 99 85]
 [ 99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
 [106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
 [114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
 [133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
 [128 137 144 140 109 95 86 70 62 65 63 63 60 73 86 101]
 [125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
 [127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
 [115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
 [ 89 93 90 97 108 147 131 118 113 114 113 109 106 95 77 80]
 [ 63 77 86 81 77 79 102 123 117 115 117 125 130 115 87]
 [ 62 65 82 89 78 73 80 101 124 126 119 101 107 114 131 119]
 [ 63 65 75 88 89 71 62 81 120 130 135 105 81 98 110 118]
 [ 87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
 [118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
 [164 146 112 80 82 120 124 104 76 48 45 66 88 101 102 109]
 [157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
 [130 128 134 161 139 106 109 118 121 134 114 87 65 53 69 86]
 [128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
 [123 107 96 86 83 112 153 149 122 109 104 75 88 107 112 99]
 [122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
 [122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```

Challenges: 조명



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)

Challenges: Occlusion (가려짐)



[This image](#) is CC0 1.0 public domain



[This image](#) is CC0 1.0 public domain



This image by [jonsson](#) is licensed
under CC-BY 2.0

Challenges: Deformation (변형)



This image by [Umberto Salvagnin](#)
is licensed under [CC-BY 2.0](#)



This image by [Umberto Salvagnin](#)
is licensed under [CC-BY 2.0](#)



This image by [sare bear](#) is
licensed under [CC-BY 2.0](#)



This image by [Tom Thai](#) is
licensed under [CC-BY 2.0](#)

Challenges: 같은 종류간의 차이



[This image](#) is CC0 1.0 public domain

이미지 분류기

■ 프로그래밍을 해서 이미지를 분류해보자!

- 뭐라고 코딩하면 될까?

```
def classify_image(image):  
    # Some magic here?  
    return class_label
```

- 예1. 숫자 100개 중에 가장 큰 수를 찾으시오.
 - 첫번째 수를 가장 큰수로 놓자
 - 숫자열의 수를 하나씩 차례대로 보자 {19, 3, 5, 13, 23, 101... }
 - 더 큰 숫자가 나타나는지 확인...

이미지 분류기

- 예2. 이미지(숫자 더미)가 나타내는 의미를 파악하시오.

```
[ [105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
[ 91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
[ 76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
[ 99 81 81 93 128 131 127 100 95 98 102 99 96 93 101 94]
[106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
[114 108 85 55 69 64 54 64 87 112 129 98 74 84 91]
[133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
[128 137 144 148 109 95 86 70 62 65 63 63 60 73 86 101]
[125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
[127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
[115 114 109 123 158 148 131 116 113 109 100 92 74 65 72 78]
[ 89 93 90 97 108 147 131 116 113 114 113 109 106 95 77 80]
[ 63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
[ 62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
[ 63 65 75 88 89 71 62 81 120 138 135 105 81 98 110 118]
[ 87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
[118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
[164 146 112 88 82 128 124 104 76 48 45 66 88 101 102 109]
[157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
[130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
[128 112 96 117 158 144 120 115 104 107 102 93 87 81 72 79]
[123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
[122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
[122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```



고양이 ?!?

이미지 분류기

우주에 로켓쏘기



vs 고양이 분류하는 컴퓨터 만들기



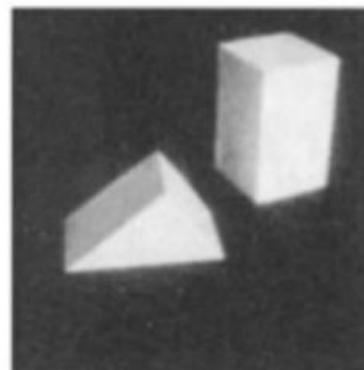
= 고양이

너무 어려워서 2012년(*)까지도
컴퓨터가 제대로 풀지 못했던 문제

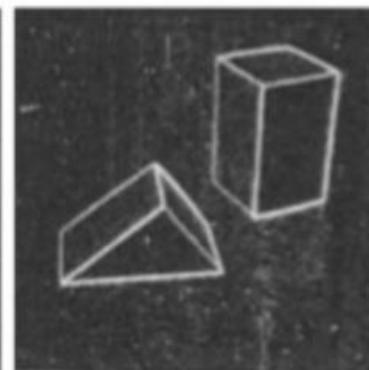
(*) 우주에 사람을 보내고, 손바닥만한 스마트폰과 120층 빌딩을 만드는 21세기

컴퓨터 비전의 역사

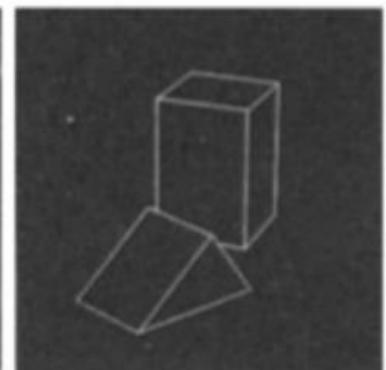
■ 1960년대: 최초의 컴퓨터 비전 연구



Input image



2×2 gradient operator



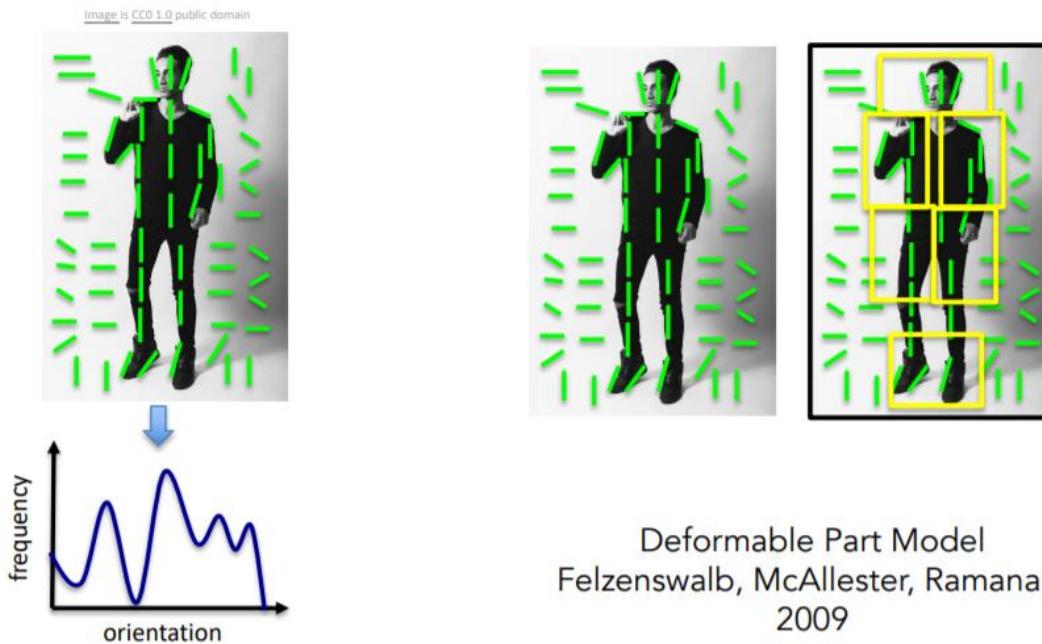
computed 3D model
rendered from new viewpoint

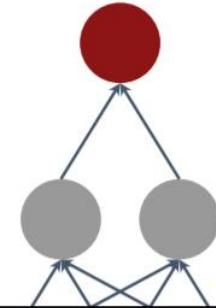
Larry Roberts
1963, 1st thesis of Computer Vision

컴퓨터 비전의 역사

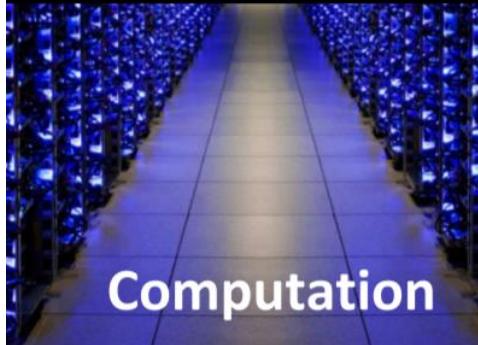
■ 1990~2010년

- 다양한 컴퓨터 시각 구현 시도
→ 매우 복잡하지만, 의미있는 성능을 내는데는 실패

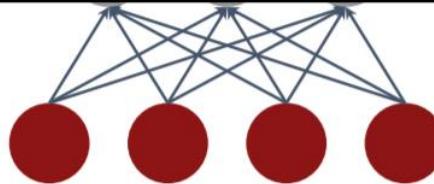




The Deep Learning Revolution



Computation



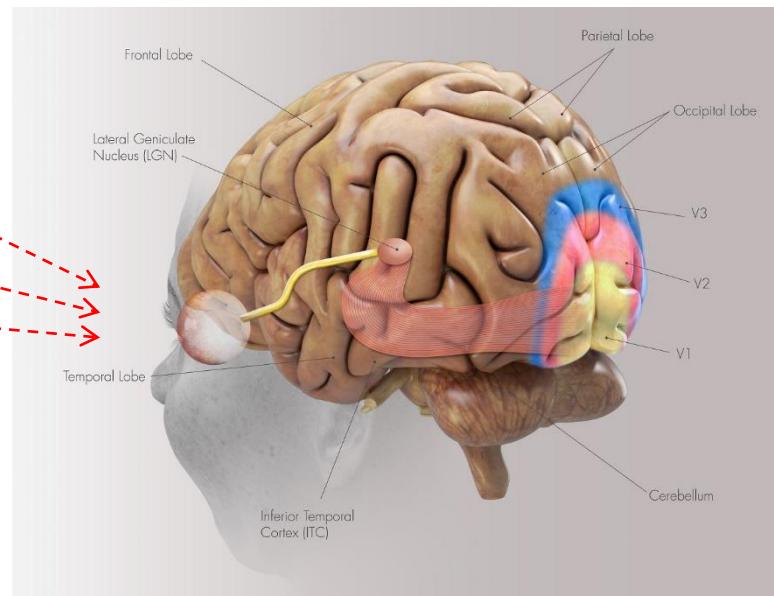
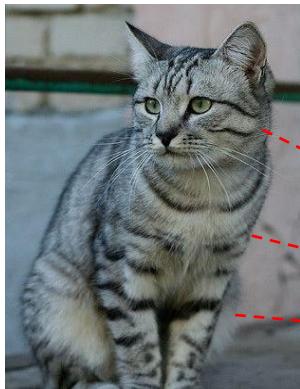
Algorithms



Data

사람은 어떻게 하고있나?

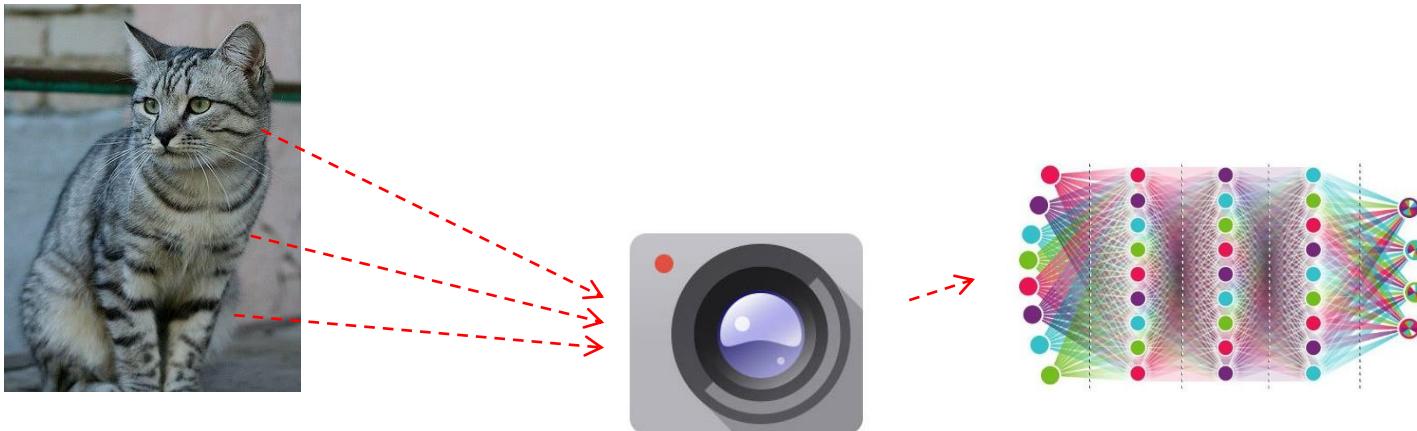
- 눈을 이용한 이미지 입력
- 뇌를 통한 이미지 패턴 인식
 - 축적한 데이터를 통해 패턴 학습
 - 뇌 안에서 무슨 일이 일어나는지 아직 정확히 모른다.
(뇌 → 거대한 용량을 가진 패턴 학습기)



source: <https://becominghuman.ai/from-human-vision-to-computer-vision-how-far-off-are-we-part1-3-b35d37a196a4>

인공지능은 어떻게하면 좋을까?

- 카메라를 이용한 이미지 입력
- 뉴럴네트워크(인공뇌)를 통한 이미지 패턴 인식
 - 뇌와 뉴런의 구조를 모방하여 만든 인공지능
 - 패턴을 학습하는데 쓰인다 → 뇌와 비슷한 느낌
(뉴럴네트워크 → 거대한 용량을 가진 패턴 학습기)



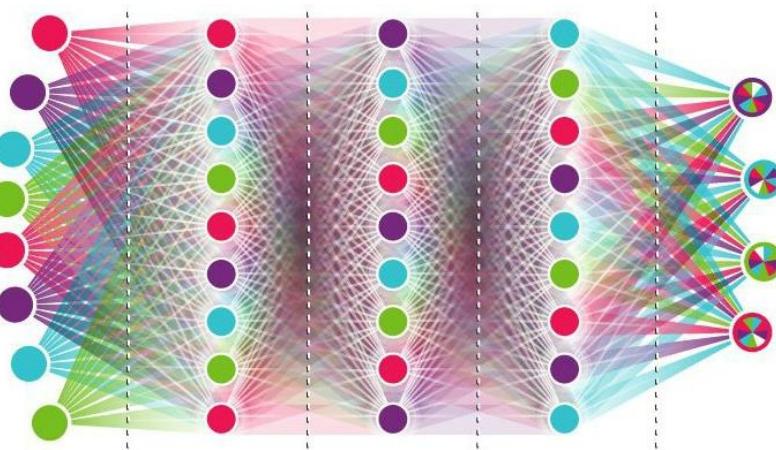
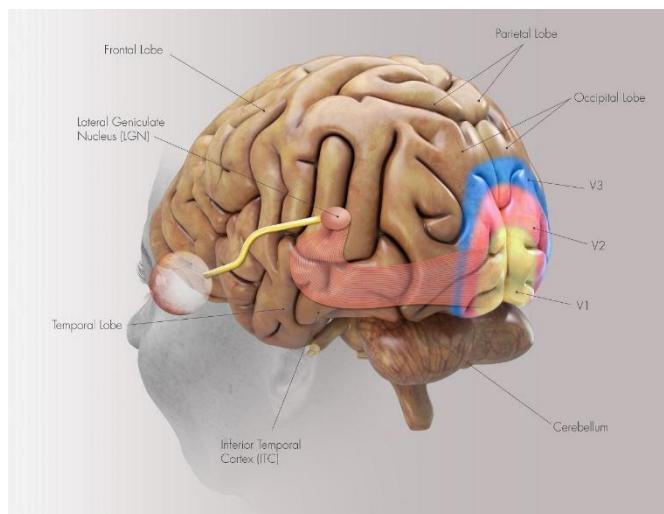
뇌와 뉴럴네트워크

■ 공통점:

- 무언가 거대한 용량을 가진 패턴 학습기
- 안에서 무슨 일이 일어나는지 완벽히 이해하기 힘들다

■ 차이점:

- **뇌:** 수만~백만년의 자연선택 및 진화의 산물
- **뉴럴네트워크:** 수학적으로 잘 설계된 인공 구조

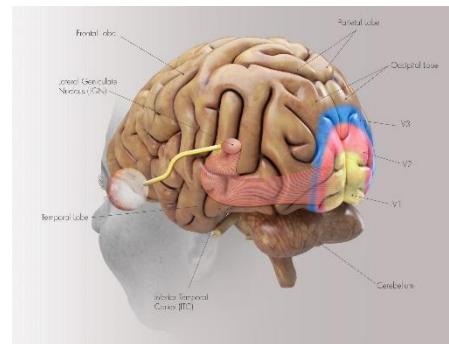


학습이란?

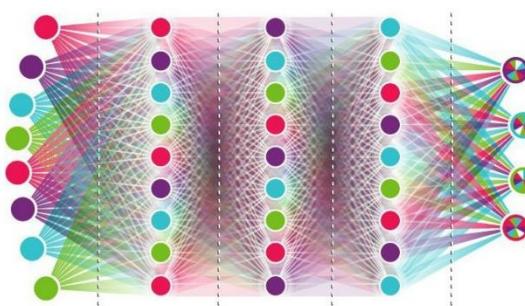
■ 정답과 피드백(feedback)을 통한 정확한 패턴 학습



→ →



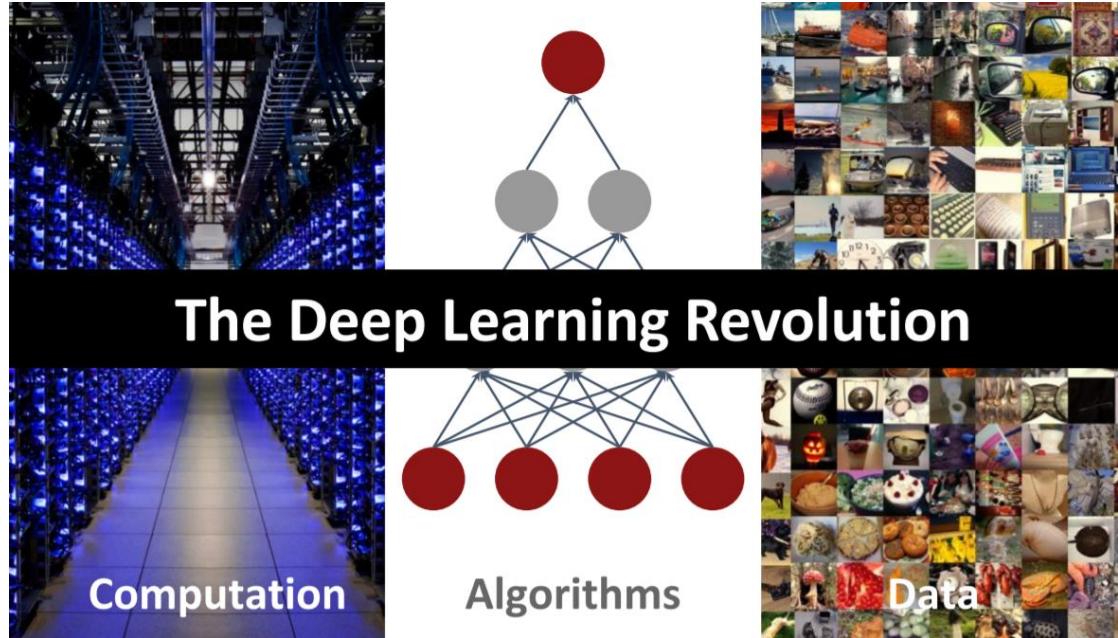
→ → Dog?



← ← 빼비빅. 답은 고양이다.

피드백

딥러닝 혁명



- **딥러닝(Deep Learning):** 현대 인공지능 기술의 중추
 - 컴퓨터 성능 향상
 - 알고리즘의 발달
 - 빅데이터

데이터의 중요성

■ 기계학습 – Data-Driven Approach

- 이미지와 정답 데이터를 수집한다.
- 머신러닝 알고리즘으로 학습한다.
- 새로운 이미지를 판별한다.

airplane



automobile



bird



cat



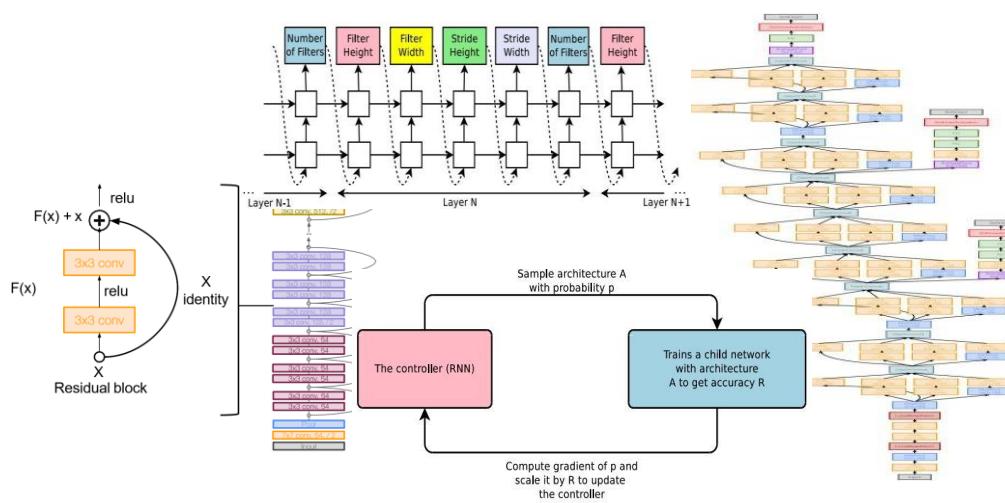
deer



이미지 인식을 위한 딥러닝 구조

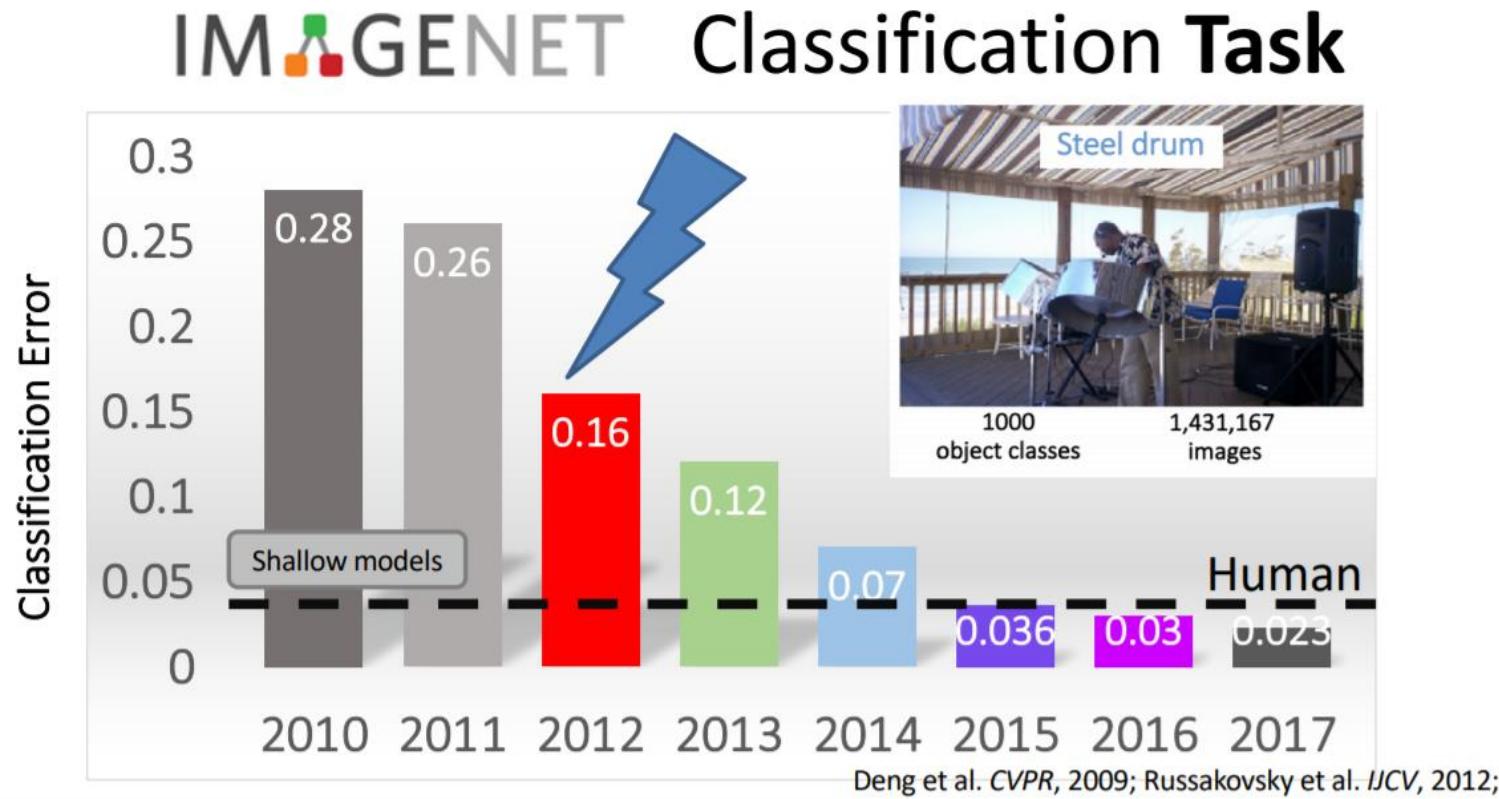
■ 정형화된 구조

- 컴퓨터 과학자/공학자들의 피땀 어린 산물
- 내용을 다 이해 못해도 사용할 수 있다
 - 엔진 만드는 법을 몰라도 운전할 수 있듯



딥러닝 혁명

- 결국 2015년 이후 사람의 성능을 뛰어넘었다



요약

■ 이미지 인식은 컴퓨터에게 무척 어려운 일

- 하지만 6살짜리 어린이도 쉽게 할수있다
- 21세기 초에도 아무도 해결 못한 영역



■ 딥러닝 혁명

- 빅데이터
- 딥러닝 알고리즘
- 하드웨어의 발전

→ 결국 2015년 이후 인간의 점수를 컴퓨터가 추월

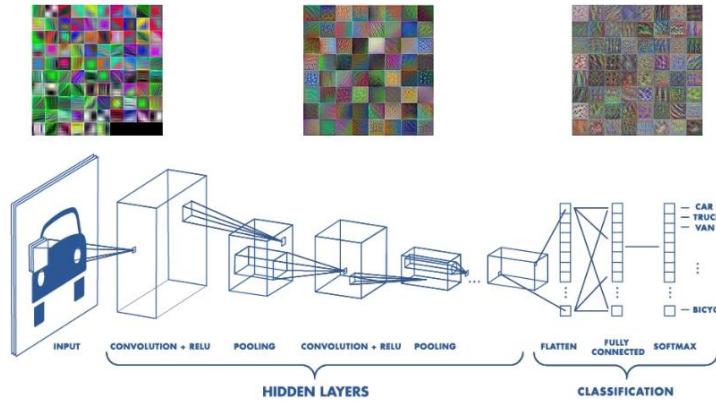
```
[[105 112 108 111 184 99 106 99 96 103 112 119 104 97 93 87]
 [ 91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
 [ 76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
 [ 99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
 [106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
 [114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
 [113 137 147 183 65 81 65 52 54 74 84 102 93 85 82]
 [120 125 137 148 140 104 105 86 82 62 56 63 80 86 101
 [126 133 148 137 119 121 117 94 75 79 80 85 54 64 72 80]
 [127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
 [115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
 [ 89 93 98 97 108 147 131 118 113 114 113 109 106 95 77 88]
 [ 63 77 86 81 77 79 102 123 117 115 117 125 130 115 87]
 [ 62 65 82 89 78 71 88 101 124 126 119 101 107 114 131 119]
 [ 63 65 75 88 89 71 62 81 128 138 135 105 81 98 110 118]
 [ 87 65 71 87 108 95 69 45 76 130 126 107 92 94 105 112]
 [118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
 [164 146 112 80 82 128 124 104 76 48 45 66 88 101 102 109]
 [157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
 [130 128 134 161 139 100 109 111 121 134 114 87 65 53 69 86]
 [128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
 [123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
 [122 121 102 88 82 86 94 117 145 148 153 162 58 78 92 107]
 [122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```

컴퓨터 시각 혁명의 중심에는 딥러닝(Deep Learning)이 있다.

Part 2. CNN

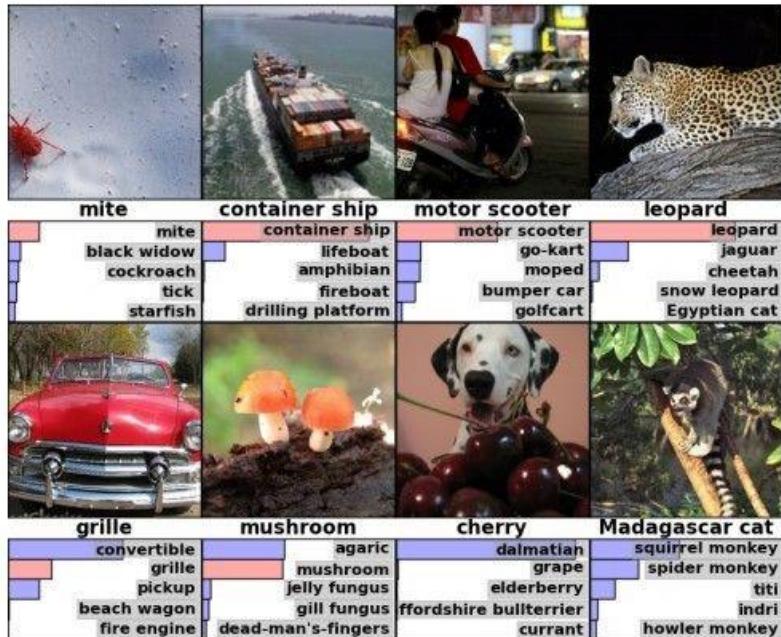
이미지 인식을 위한 딥러닝 구조

Convolutional Neural Network



Fast-forward to today: ConvNets are everywhere

분류



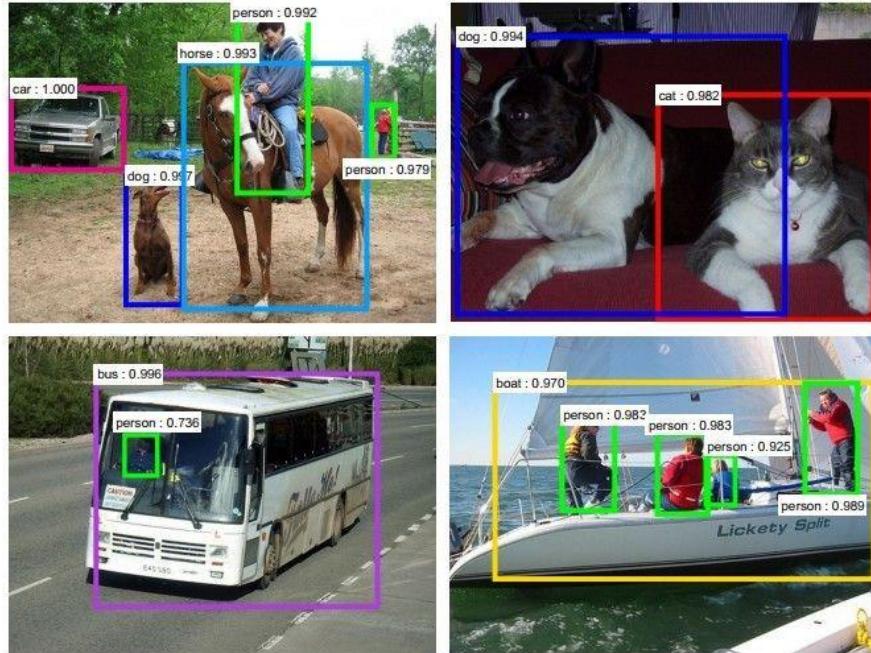
정보 검색



Figures copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Fast-forward to today: ConvNets are everywhere

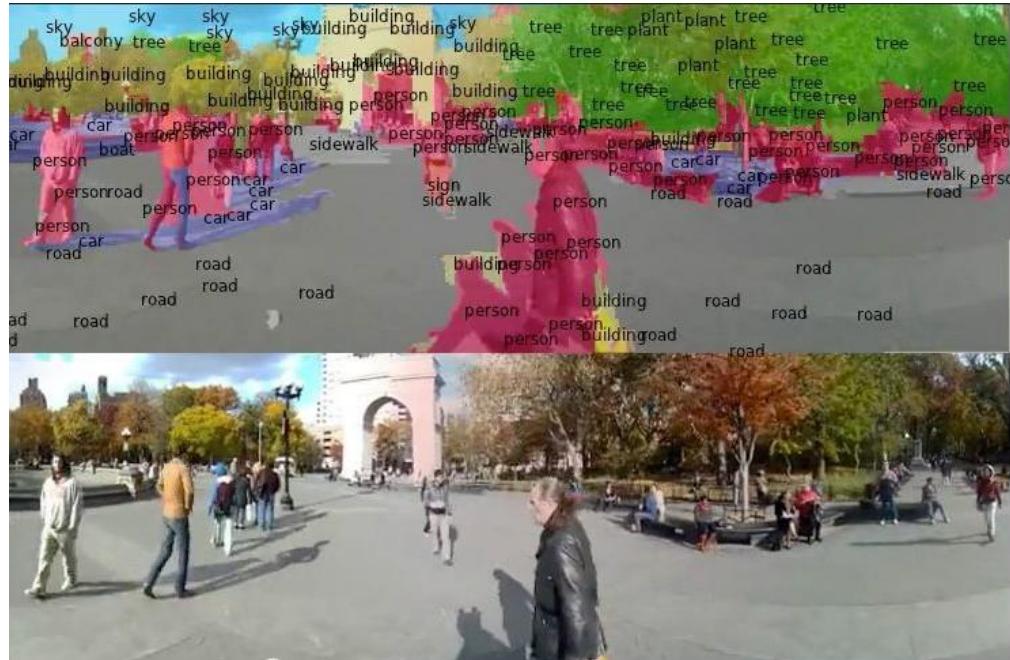
탐지



Figures copyright Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2015. Reproduced with permission.

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

영역 세분화(segmentation)



Figures copyright Clement Farabet, 2012.
Reproduced with permission.

[Farabet et al., 2012]

Fast-forward to today: ConvNets are everywhere



self-driving cars

Photo by Lane McIntosh. Copyright CS231n 2017.



[This image](#) by GBPublic_PR is
licensed under [CC-BY 2.0](#)

NVIDIA Tesla line

(these are the GPUs on rye01.stanford.edu)

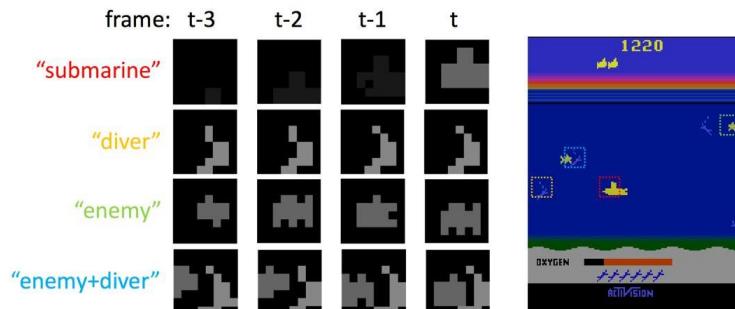
Note that for embedded systems a typical setup would involve NVIDIA Tegras, with integrated GPU and ARM-based CPU cores.

Fast-forward to today: ConvNets are everywhere



Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014]

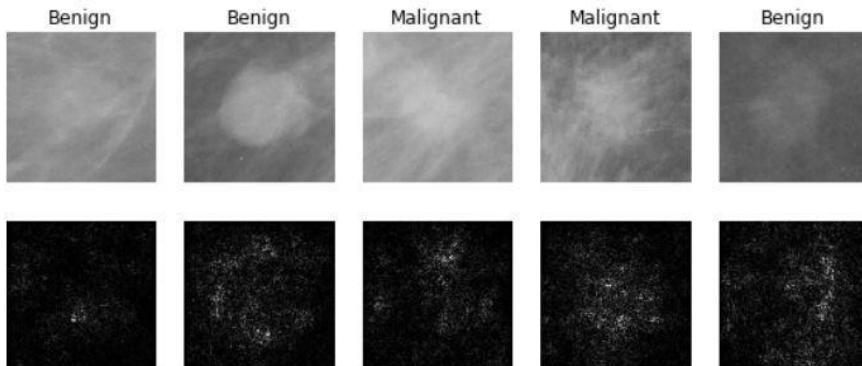


[Guo et al. 2014]



Figures copyright Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard Lewis, and Xiaoshi Wang, 2014. Reproduced with permission.

Fast-forward to today: ConvNets are everywhere



[Levy et al. 2016]

Figure copyright Levy et al. 2016.
Reproduced with permission.



[Dieleman et al. 2014]

From left to right: [public domain by NASA](#), usage [permitted](#) by
ESA/Hubble, [public domain by NASA](#), and [public domain](#).



[Sermanet et al. 2011]

[Ciresan et al.]

Photos by Lane McIntosh.
Copyright CS231n 2017.

Fast-forward to today: ConvNets are everywhere

[This image](#) by Christin Khan is in the public domain and originally came from the U.S. NOAA.



Whale recognition, Kaggle Challenge

Photo and figure by Lane McIntosh; not actual example from Mnih and Hinton, 2010 paper.



Mnih and Hinton, 2010

Fast-forward to today: ConvNets are everywhere

No errors



A white teddy bear sitting in the grass

Minor errors



A man in a baseball uniform throwing a ball

Somewhat related



A woman is holding a cat in her hand

Image Captioning

[Vinyals et al., 2015] [Karpathy and Fei-Fei, 2015]



A man riding a wave on top of a surfboard



A cat sitting on a suitcase on the floor

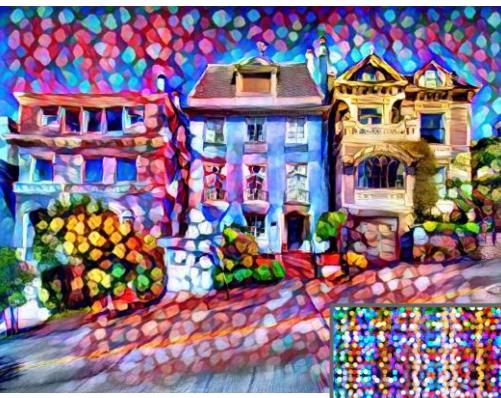
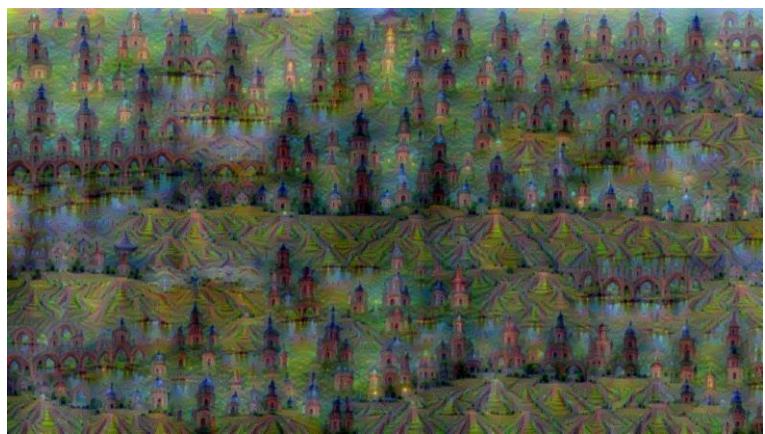
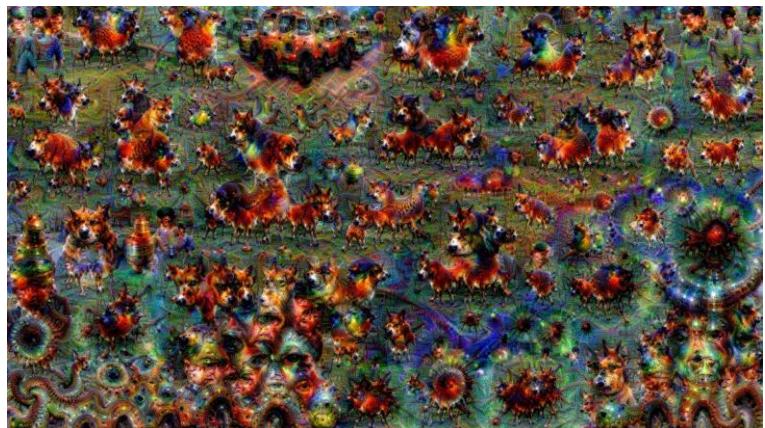


A woman standing on a beach holding a surfboard

All images are CC0 Public domain: <https://pixabay.com/en/luggage-antique-cat-1643010/> <https://pixabay.com/en/teddy-plush-bears-cute-teddy-bear-1623436/> <https://pixabay.com/en/surf-wave-summer-sport-litoral-1668716/> <https://pixabay.com/en/woman-female-model-portrait-adult-983967/> <https://pixabay.com/en/handstand-lake-meditation-496008/> <https://pixabay.com/en/baseball-player-shortstop-infield-1045263/>

Captions generated by Justin Johnson using [Neuraltalk2](#)

Fast-forward to today: ConvNets are everywhere

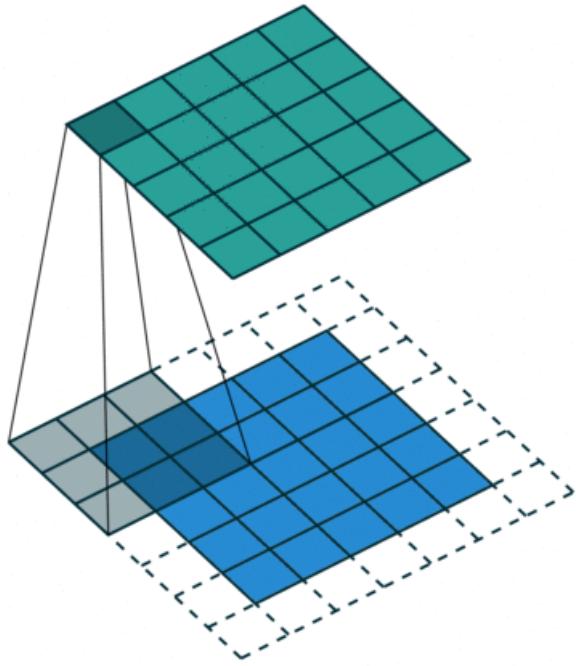


Figures copyright Justin Johnson, 2015. Reproduced with permission. Generated using the Inceptionism approach from a [blog post](#) by Google Research.

Original image is CC0 public domain
[Starry Night](#) and [Tree Roots](#) by Van Gogh are in the public domain
[Bokeh image](#) is in the public domain
Stylized images copyright Justin Johnson, 2017; reproduced with permission

Gatys et al, “Image Style Transfer using Convolutional Neural Networks”, CVPR 2016
Gatys et al, “Controlling Perceptual Factors in Neural Style Transfer”, CVPR 2017

2D Convolution



3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

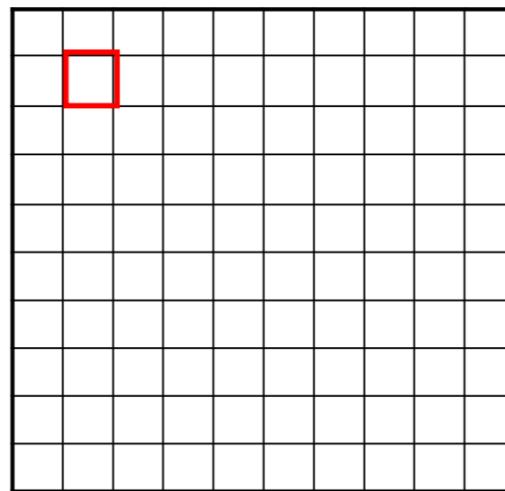
3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

2D Convolution

$$I[.,.]$$

$$h[.,.]$$



$$f[\cdot, \cdot]$$

	1	1	1
1	1	1	1
9	1	1	1

$$h[m,n] = \sum f[k,l] I[m+k, n+l]$$

2D Convolution

$I[.,.]$

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	0	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10									

$f[.,.]$

1	1	1
1	1	1
1	1	1

$$h[m, n] = \sum f[k, l] I[m + k, n + l]$$

2D Convolution

$I[.,.]$

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	0	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

$h[.,.]$

0	10	20	30	30							

$f[.,.]$

1	1	1
1	1	1
1	1	1

$$h[m, n] = \sum f[k, l] I[m+k, n+l]$$

2D Convolution

$I[.,.]$

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	90	0	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30						

$f[.,.]$

1	1	1
1	1	1
1	1	1

$$h[m, n] = \sum f[k, l] I[m + k, n + l]$$

2D Convolution

$$I[\cdot, \cdot]$$

$$h[.,.]$$

$$f[\cdot, \cdot]$$

1
9

1	1	1
1	1	1
1	1	1

$$h[m,n] = \sum f[k,l] I[m+k, n+l]$$

2D Convolution

$I[.,.]$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10		
	0	20	40	60	60	60	40	20		
	0	30	60	90	90	90	60	30		
	0	30	50	80	80	90	60	30		
	0	30	50	80	80	90	60	30		
	0	20	30	50	50	60	40	20		
	10	20	30	30	30	30	20	10		
	10	10	10	0	0	0	0	0		

$f[.,.]$

1	1	1
1	1	1
1	1	1

$$\frac{1}{9}$$

$$h[m, n] = \sum f[k, l] I[m + k, n + l]$$

2D Convolution

What does it do?

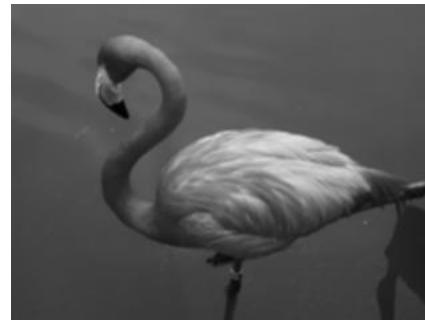
- Replaces each pixel with an **average of its neighborhood**
- Achieve **smoothing** effect (**remove sharp features**)

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2D Convolution으로 할 수 있는 것

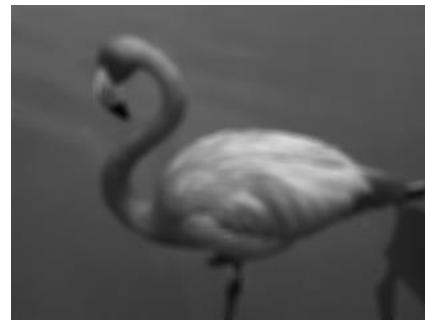
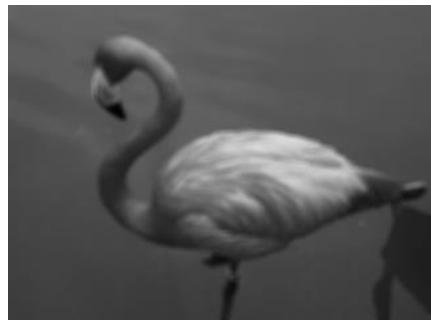
- 이미지 스무딩

Original
image



3×3
Mean filtering

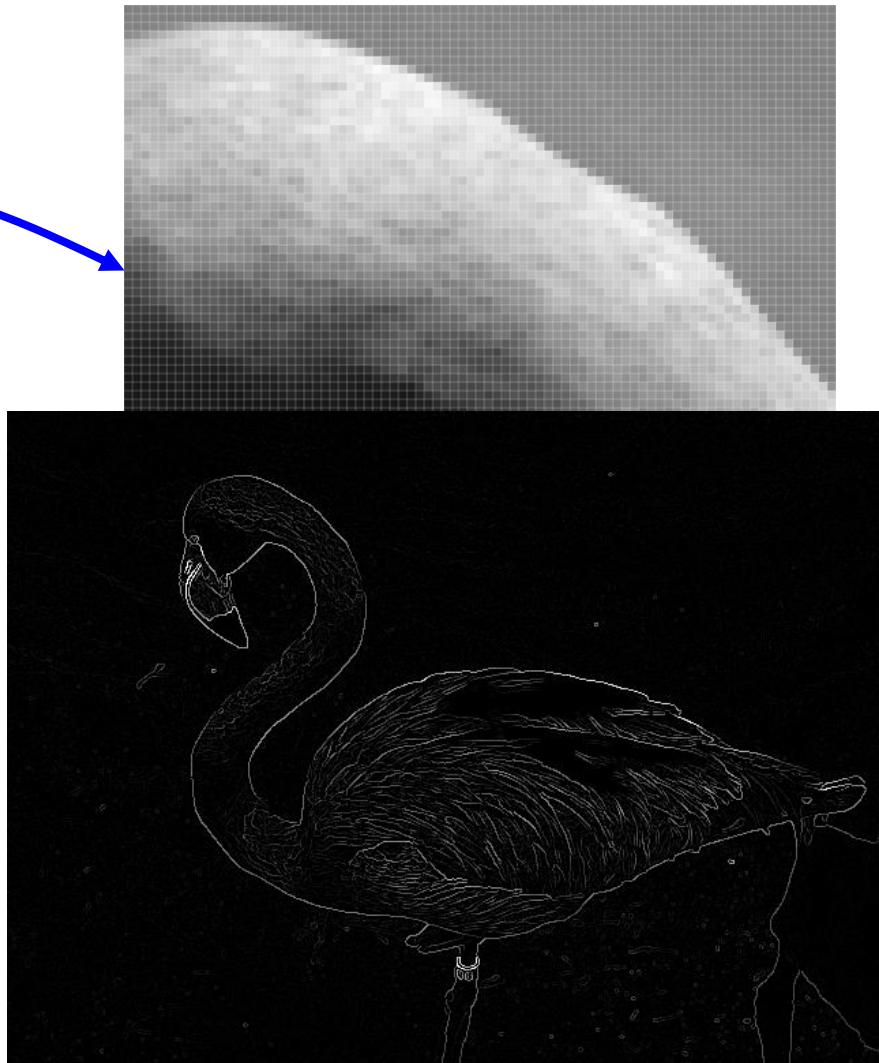
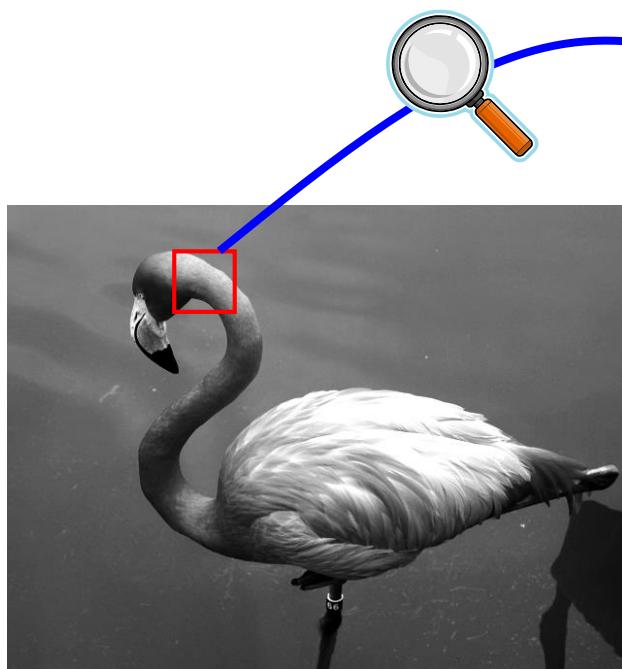
5×5



7×7

2D Convolution으로 할 수 있는 것

- 경계선 및 패턴 검출



2D Convolution으로 할 수 있는 것

- 경계선 및 패턴 검출

Original



2D Convolution으로 할 수 있는 것

- 이미지 선명화



Original image



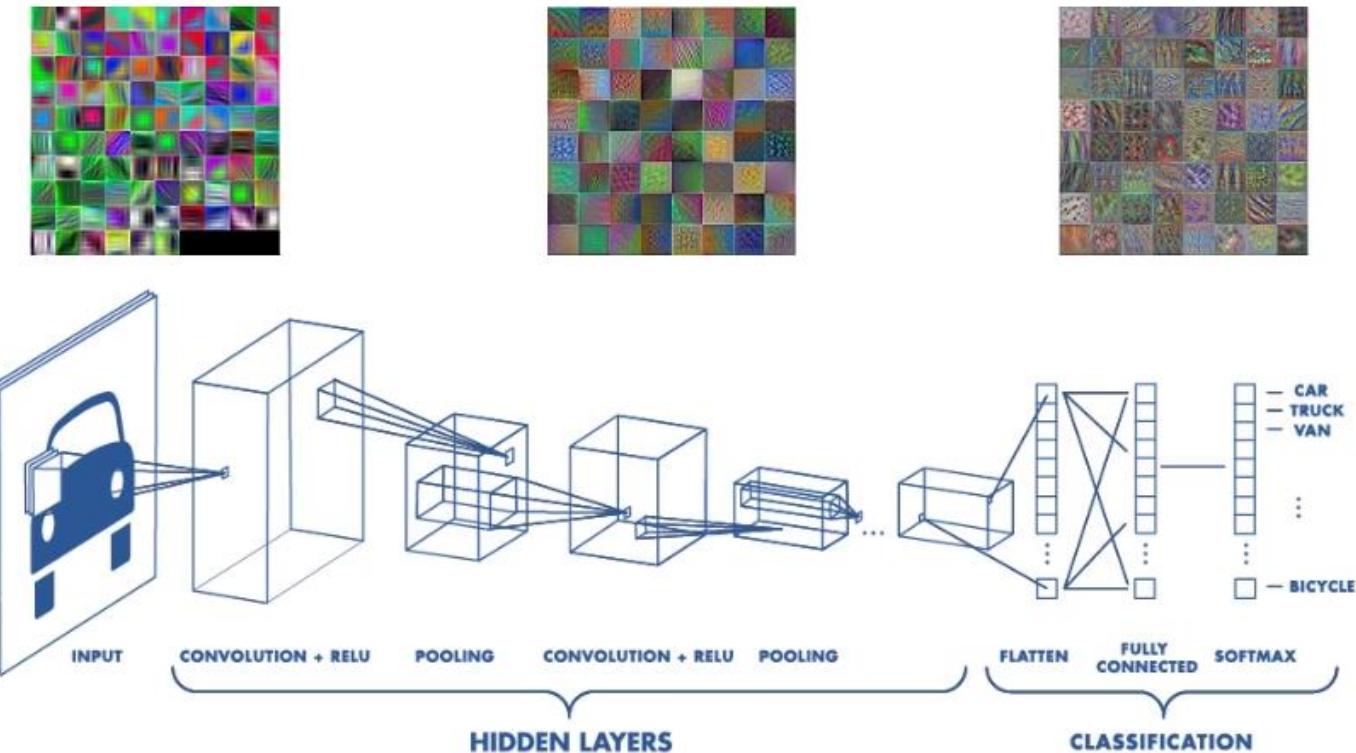
유형1



유형2

이미지 인식을 위한 딥러닝 구조

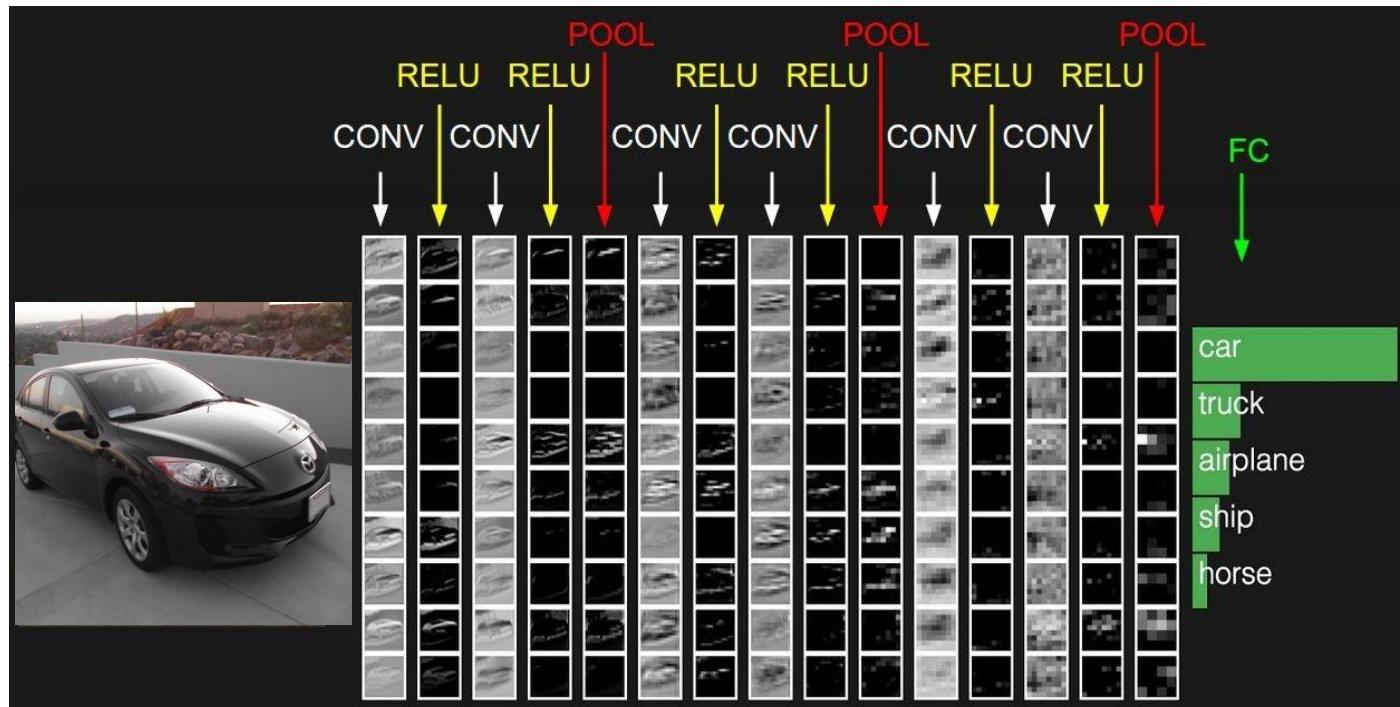
Convolutional Neural Network



Convolutional Neural Networks

■ 컨볼루션과 이미지 축소를 반복하는 구조

- 여기서 컨볼루션 방식은 **학습**으로 결정



자세한 과정은 딥러닝 커리큘럼에서.

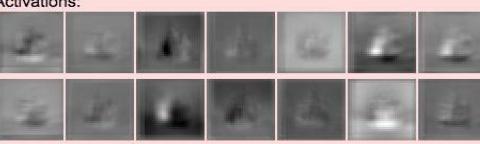
ConvNetJS 데모

Network Visualization

input (32x32x3)
max activation: 0.34313, min: -0.49608
max gradient: 0.04754, min: -0.0368



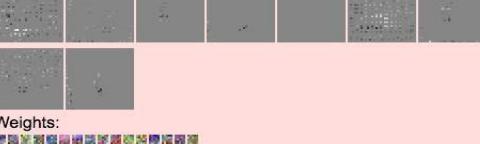
Activations:



conv (32x32x16)
filter size 5x5x3, stride 1
max activation: 1.42613, min: -1.28123
max gradient: 0.03521, min: -0.03962
parameters: $16 \times 5 \times 3 + 16 = 1216$



Activation Gradients:



Weights:



Weight Gradients:



Example predictions on Test set

test accuracy based on last 200 test images: 0.16666666666666666

 horse frog deer	 horse truck cat	 frog dog deer	 frog dog cat
 frog deer horse	 car truck cat	 airplane truck car	 frog horse deer
 deer horse dog	 truck car horse	 car truck airplane	 horse deer bird
 dog frog deer	 truck car ship	 car cat dog	 dog cat frog
 deer dog horse	 frog deer horse	 deer frog horse	 horse deer car
 horse frog deer	 frog deer dog	 frog horse deer	 frog deer horse
 truck airplane ship	 car airplane ship	 dog cat bird	 ship airplane truck
 car ship airplane	 dog frog deer	 airplane bird ship	 frog dog deer

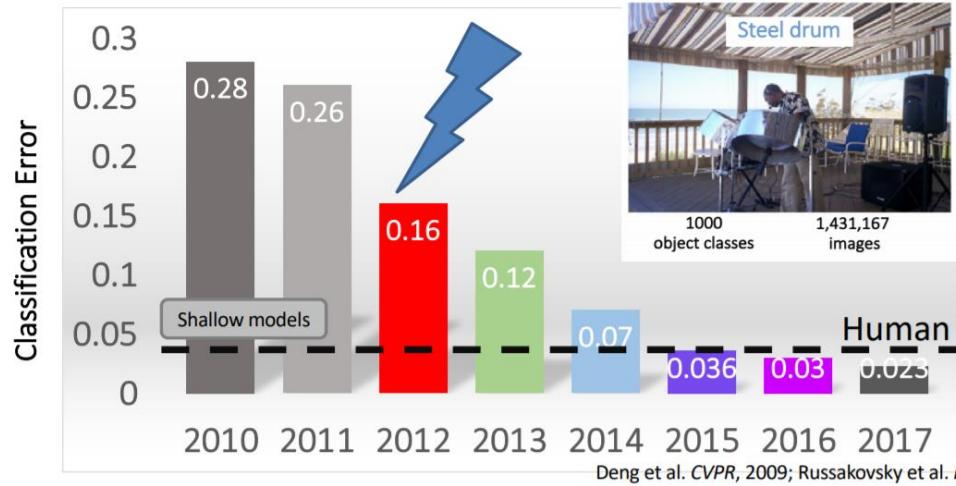
<http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html>

Part 3.

인공지능은 항상 옳은가?

인간을 앞지른 인공지능 기술

IMAGENET Classification Task



딥러닝: 패턴인식의 제왕?

아마존의 채용 인공지능

■ 여성을 차별하는 Amazon의 인공지능

- 남성위주의 데이터, 성과로 학습
- 우수한 여성인재 선발을 가로막은 편향된 인공지능



Photo by [Bryan Angelo](#) on [Unsplash](#)

인종에 따른 차별

■ 백인 위주로 구성된 안면 데이터

- 흑인과 동양인에 대한 인식률 하락
- 백인 남성: 1% 안면인식 오차
- 흑인 여성: 35% 안면인식 오차



Gender was misidentified in up to 1 percent of lighter-skinned males in a set of 385 photos.

■ 범죄 프로파일링

- 범죄 이력을 통한 재범률 예측 인공지능
- 흑인 범죄자에 대한 높은 재범 오탐지율
→ 개과천선한 사람에 대한 억측



Gender was misidentified in 35 percent of darker-skinned females in a set of 271 photos.

편향된 데이터와 알고리즘

■ 편향된 데이터와 알고리즘

- 데이터가 잘못되거나 편향되어서 발생한 이슈
- 인공지능의 편향성을 줄이기 위한 지속적인 노력

■ 편향성이 이슈화 된 이후 많은 문제가 해결되고 있다

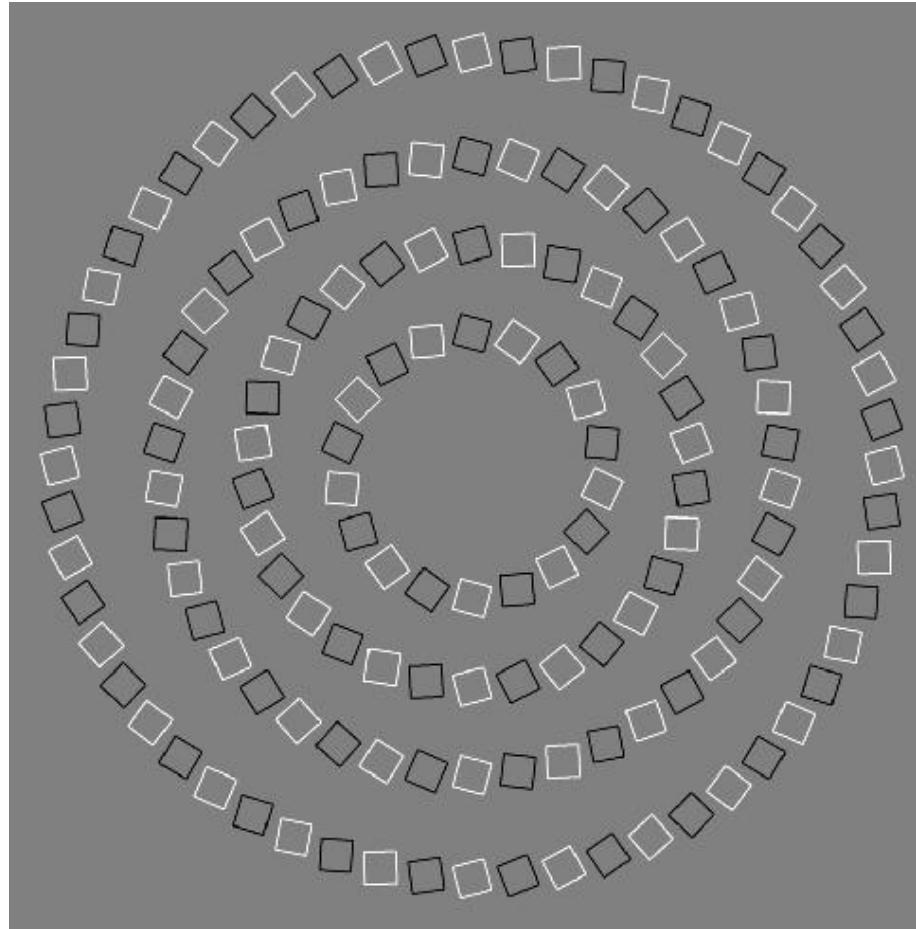
- 기획/개발자가 신경쓰면 상당 수 해결되는 문제

■ 모든 잘못은 데이터에 있었던 것인가?

- 데이터와 알고리즘을 공평하게 만들면 문제가 없는 것일까?
- 편향성만이 문제일까?

사람의 뇌는 잘 속는다!

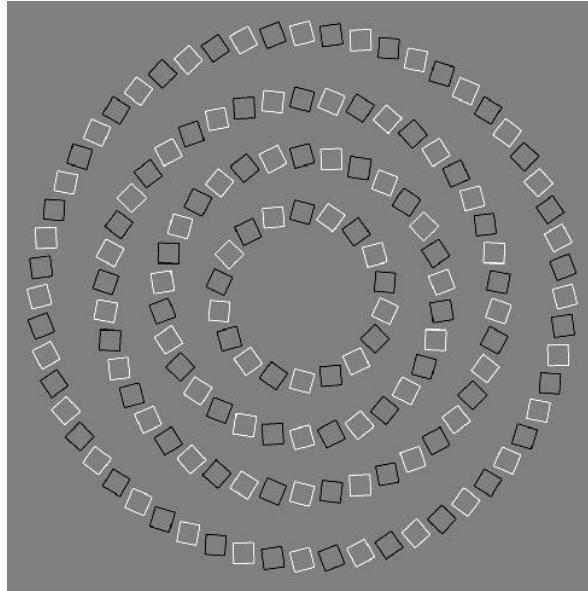
- 그림안에 어떤 형태가 보이나요?



(Pinna and Gregory, 2002)

사람의 뇌는 잘 속는다!

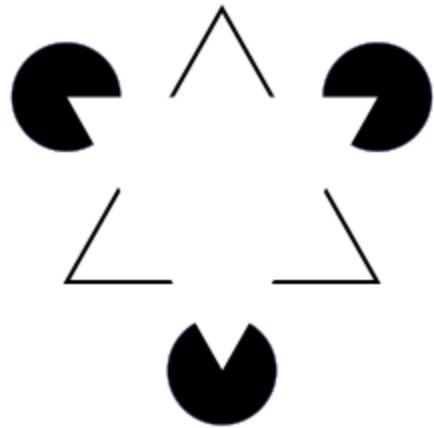
■ 그림안에 무엇이 보이나요?



- 정답: 원 4개 (소용돌이 모양 아님)

사람의 뇌는 잘 속는다!

■ 그 외 예시



인공지능도 속을까?

절대적 샘플의 역사

■ Timeline

- 2004 → 정상같은 스팸메일 만들기
- 2013 → 뉴럴 네트워크 속이기, 이미지 분류기 속이기
- 2014 → 간단한 인공지능 공격 기술
- 2015 →



판다

+ .007 ×



=



코끼리

- 2004, Adversarial classification,' Dalvi et al.
- 2013, 'Evasion attacks Against Machine Learning at Test Time,' Biggio.
- 2013, 'fool ImageNet classifiers imperceptibly,' Szegedy et al.
- 2014, 'cheap cloased form attack', Goodfellow et al.

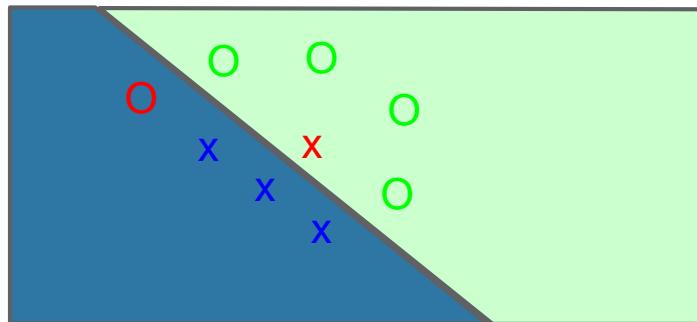
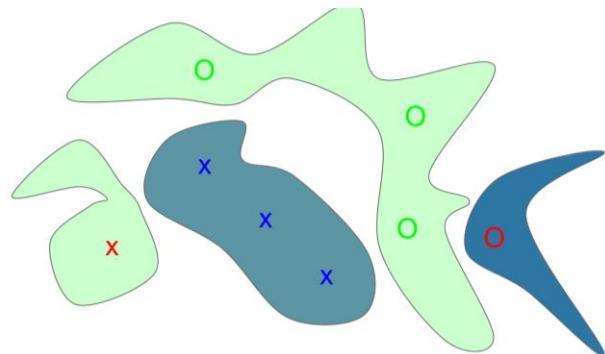
절대적 샘플 발생원인

■ 과적합 (overfitting)

- 데이터의 형상과 분포를 너무 자세하게 학습

■ 과도한 구분선:

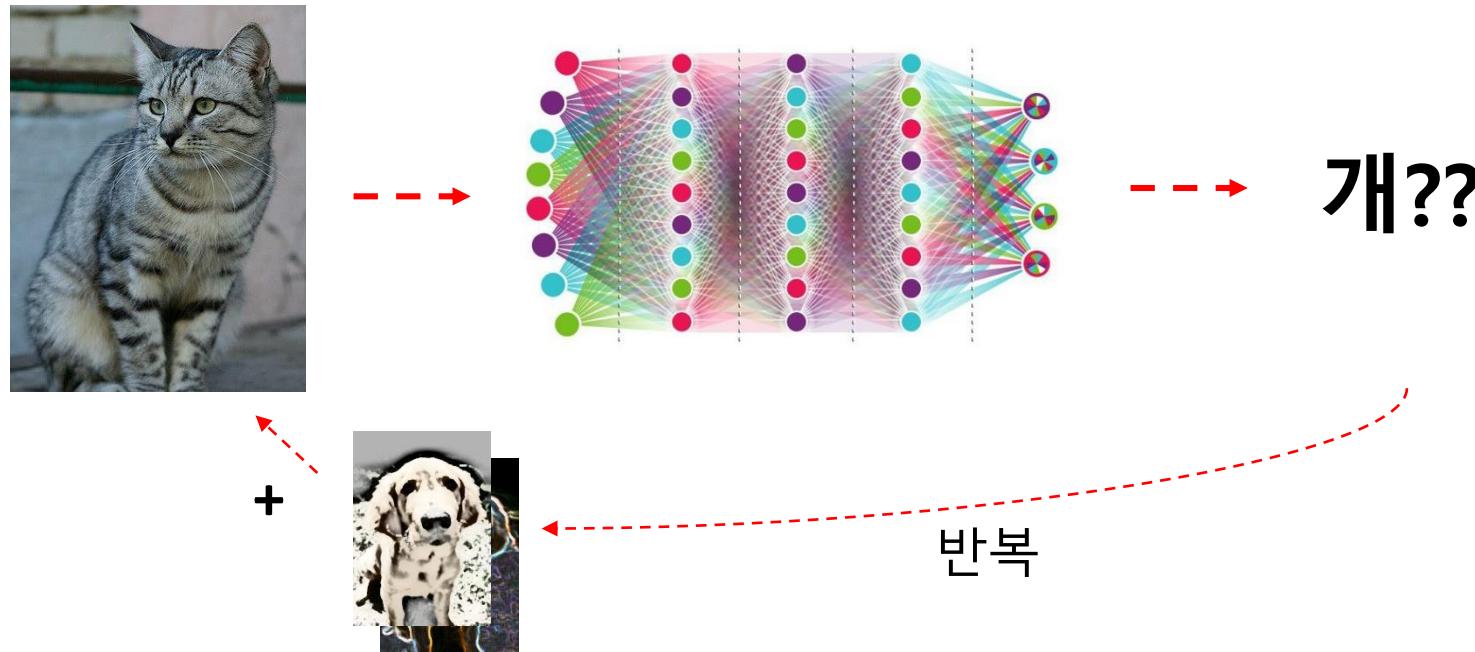
- 선으로 무언가를 o/x로 구분하는것에 대한 한계



(Goodfellow 2016)

생성 방법 1

- 1. 인공지능이 개로 생각하는 노이즈를 더한다
- 2. 뉴럴넷이 무엇으로 판단하는지 본다
- 3. 뉴럴넷이 완전히 속을때까지 1,2번 반복

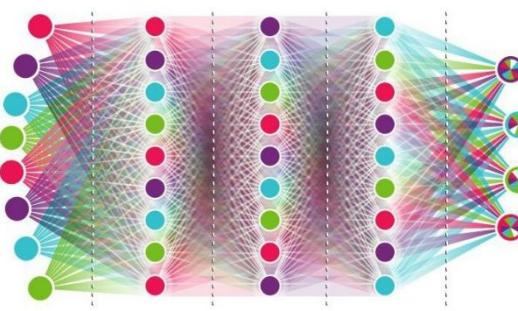


생성 방법 2

- 1. 정답을 일부러 틀리게 준다.
- 2. 뉴럴넷에 가한 피드백을 이미지에도 함께 준다.
- 3. 뉴럴넷이 완전히 속을때까지 1,2번 반복



--->
←---
피드백
+



---> 고양이?
←--- 빼비빅. 답은 자동차다.
피드백

절대적 샘플 예시

■ 절대적 샘플 1

- 인공지능의 오작동 유발하는 1픽셀 공격



Automobile (Dog)



Automobile

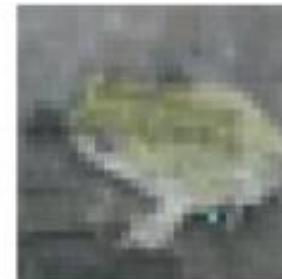


Cat (Dog)

(Airplane)



Frog (Dog)



Frog (Truck)



Dog (Cat)

https://www.researchgate.net/figure/Illustration-of-one-pixel-adversarial-attacks-68-The-correct-label-is-mentioned-with_fig1_322221397

적대적 샘플 예시

■ 적대적 샘플 2 (adversarial example)

- 자율자동차의 오작동 유발

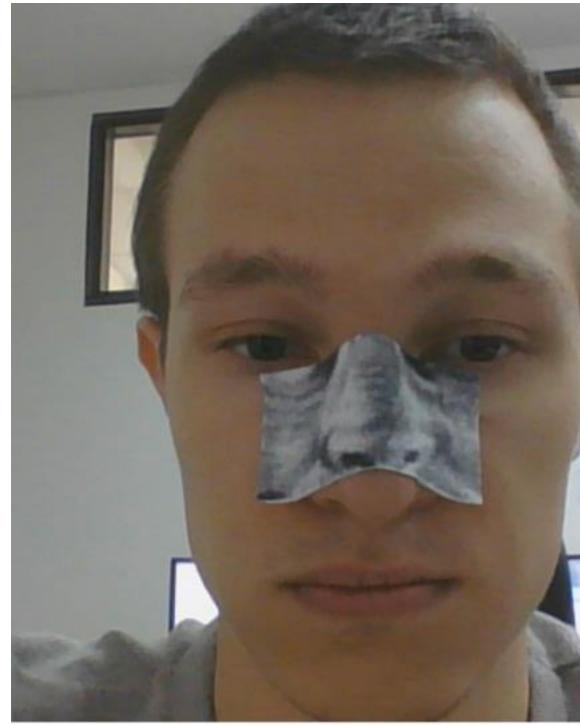


<https://medium.com/self-driving-cars/adversarial-traffic-signs-fd16b7171906>

적대적 샘플 예시

■ 적대적 샘플 3

- 안면인식 시스템의 오작동 유발

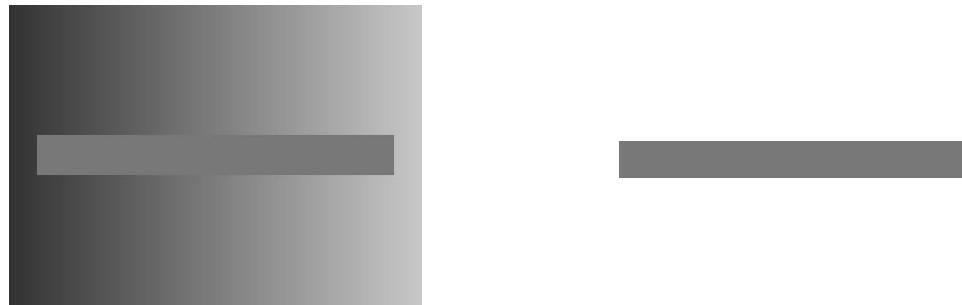


On adversarial patches: real-world attack on ArcFace-100 face recognition system, <https://arxiv.org/pdf/1910.07067.pdf>

어떻게 해결할까?

■ 가장 쉬운 해결책

- 속을 수 있는 데이터를 만들어서 같이 학습한다.
- 인공지능도 이상한 데이터를 경험할 수 있게 한다.
 - (사람의 경우) 아래 이미지 안쪽 막대는 같은색이야!



■ 다양한 해결책이 현재 활발히 연구중

- 현재 진행형 연구
- 근본적으로 창과 방패의 대결
 - 좋은 방패 이후에 더 좋은 창이 나온다

Part 4.

인공지능 경진대회

경진대회 플랫폼

- Kaggle: 구글의 대표적인 인공지능 경진대회 플랫폼



경진대회 플랫폼

DACON 커뮤니티 대회 교육 랭킹 더보기

로그인 회원가입



디스코드에서 기다릴게요!

자유채팅, 문의, 스터디 모집까지 디스코드 채널에서 다이렉트로!

⑤ 10억 8960만원 상금 450,618 제출 141,171 팀 참여 160 개 대회 개최 1,496,871 xp

진행중인 대회 >

오직 데이콘에서만 참여할 수 있어요

대회 이미지	제목	개최일	마감일	태그
	도배 하자 유형 분류 AI 경진대회	2023.04.10	D-25	알고리즘 비전 분류 MLOps Weighted F1 Score
	신한AI, 보다 나은 금융 생활을 위한 AI…	2023.04.10	D-18	아이디어 금융 AI 서비스 플랫폼 알고리즘 정성평가
	제2회 코스포 x 데이콘 도서 추천 알고…	2023.04.17	D-18	체용 알고리즘 정형 추천시스템 RMSE

리더보드

신한AI, 보다 나은 금융 생활을 위한 AI… D-18

순위	이름	이미지
1	신한고양이	
2	위트의 전사들	
3	600마력	
4	STAYC	

인공지능 경진대회란?

■ 주어진 데이터를 토대로 문제를 푸는 대회

■ 예시:

- 학습용: 데이터 사진 10만장 + 정답
 - 평가용: 데이터 사진 10만장

- 각자 정답을 예측해서 서버에 업로드하여 **랭킹** 자동산정

The image displays a 4x8 grid of 32 clinical photographs of skin lesions. Each photograph is labeled with its corresponding diagnosis directly beneath it. The diagnoses are as follows:

- Row 1: seborrheic keratosis, melanoma, melanoma, nevus, nevus, nevus, melanoma, nevus
- Row 2: melanoma, nevus, melanoma, melanoma, melanoma, seborrheic keratosis, seborrheic keratosis, nevus
- Row 3: seborrheic keratosis, seborrheic keratosis, melanoma, seborrheic keratosis, seborrheic keratosis, melanoma, nevus, melanoma
- Row 4: melanoma, seborrheic keratosis, seborrheic keratosis, seborrheic keratosis, nevus, seborrheic keratosis, melanoma, melanoma

The images show a variety of skin textures and pigmentation patterns characteristic of each diagnosis.

?

공개 리더보드 vs 비공개 리더보드

■ 일반적으로 대회는 비공개 리더보드를 운영

- 대회 기간동안 1~50%에 해당하는 정답에 대한 랭킹
- 대회 종료 직후 모든 정답에 대한 점수를 공개

→ 격동의 랭킹판

Public Leaderboard		Private Leaderboard											
This leaderboard is calculated with approximately 20% of the test data. The final results will be based on the other 80%, so the final standings may be different.													
													
 Raw Data  Refresh													
#	Team Name	Notebook	Team Members	Score	Entries	Last							
1	Overfitting More ?			0.948	150	3d							
2	Bandabi			0.946	294	3d							
Your Best Entry ↑ Your submission scored 0.938, which is not an improvement of your best score. Keep trying!													
3	chixy			0.945	180	4d							
4	ABBA McCandless			0.945	296	3d							
5	Avane Srimanarayana			0.944	138	3d							
6	Juneau it's a stego			0.944	192	4d							
7	bestfitting			0.943	92	3d							
8	DavidChen			0.943	76	3d							

Public Leaderboard		Private Leaderboard											
The private leaderboard is calculated with approximately 80% of the test data.													
 Refresh													
#	△pub	Team Name	Notebook	Team Members	Score	Entries	Last						
1	▲ 12	Guanshuo Xu			0.936	50	4d						
2	▲ 2	ABBA McCandless			0.932	296	3d						
3	▲ 29	KaizaburoChubachi			0.931	21	3d						
4	▼ 3	Overfitting More ?			0.930	150	3d						
5	▲ 4	Roman Vlasov			0.929	108	4d						
6	▲ 22	Victor Durnov			0.929	76	4d						
7	—	bestfitting			0.929	92	3d						
8	▼ 6	Bandabi			0.929	294	3d						

경진대회 플랫폼 장점

■ 장점 1. 경쟁을 통한 동기부여

- 알아서 잠에서 일어나고, 알아서 코딩을 하게됩니다.

#	△pub	Team Name	Notebook	Team Members	Score ⓘ	Entries	Last
1	▲ 880	All Data Are Ext			0.9490	116	3mo
2	▲ 55	aloe			0.9485	61	3mo
3	▲ 262	Deloitte Analytics Spain			0.9484	118	3mo
4	▲ 210	Atagi Yuya			0.9476	23	3mo
5	▲ 723	Wenlu			0.9475	19	3mo
6	▲ 155	<^.^>			0.9468	168	3mo
7	▲ 502	James Sebastian			0.9466	75	3mo
8	▲ 218	Charlie			0.9463	58	3mo
9	▲ 243	ria			0.9462	90	3mo
10	▲ 263	thakurudit			0.9461	67	3mo
11	▲ 21	DSRGN			0.9459	387	3mo
12	▲ 397	Carcinogens			0.9458	44	3mo
13	▲ 450	Asharam Meena			0.9457	52	3mo
14	▲ 433	Kaz&Kun			0.9457	111	3mo

경진대회 플랫폼 장점

- 장점 2. 상금 + 명예 + 이력
 - 관련분야 취업 시 강력한 이력



Chris Deotte
Data Scientist & Researcher at Nvidia
San Diego, California, United States
Joined 3 years ago · last seen in the past day
<http://chrисdeotte.com/>

Followers 6236 Competitions Grandmaster

[Home](#) [Competitions \(22\)](#) [Datasets \(65\)](#) [Notebooks \(58\)](#) [Discussion \(5,466\)](#) ... [Contact User](#) [Follow User](#)

Competitions Grandmaster	Datasets Grandmaster	Notebooks Grandmaster	Discussion Grandmaster
Current Rank 40 of 149,234	Highest Rank 26	Rank 1 of 28,380	Rank 1 of 170,176
 5  8  4	 8  8  22	 44  11  3	 210  264  2712
IEEE-CIS Fraud ...  a year ago Top 1%	1st of 6381	Data Without Drift  8 months ago	315 votes
University of Liv...  6 months ago Top 1%	7th of 2618	RAPIDS  a month ago	226 votes
Instant Gratifica...  a year ago Top 1%	7th of 1832	Melanoma TFR...  4 months ago	134 votes
Triple Stratified ...  4 months ago	545 votes	25 Million Image...  2 years ago	520 votes
Feature Enginee...  a year ago	766 votes	One Feature Mo...  9 months ago	482 votes
Thank you Kagg...  3 months ago	672 votes	1st Place Solutio...  a year ago	539 votes

경진대회 플랫폼 장점

■ 장점 3. 고수들의 코드가 다양하게 올라온다

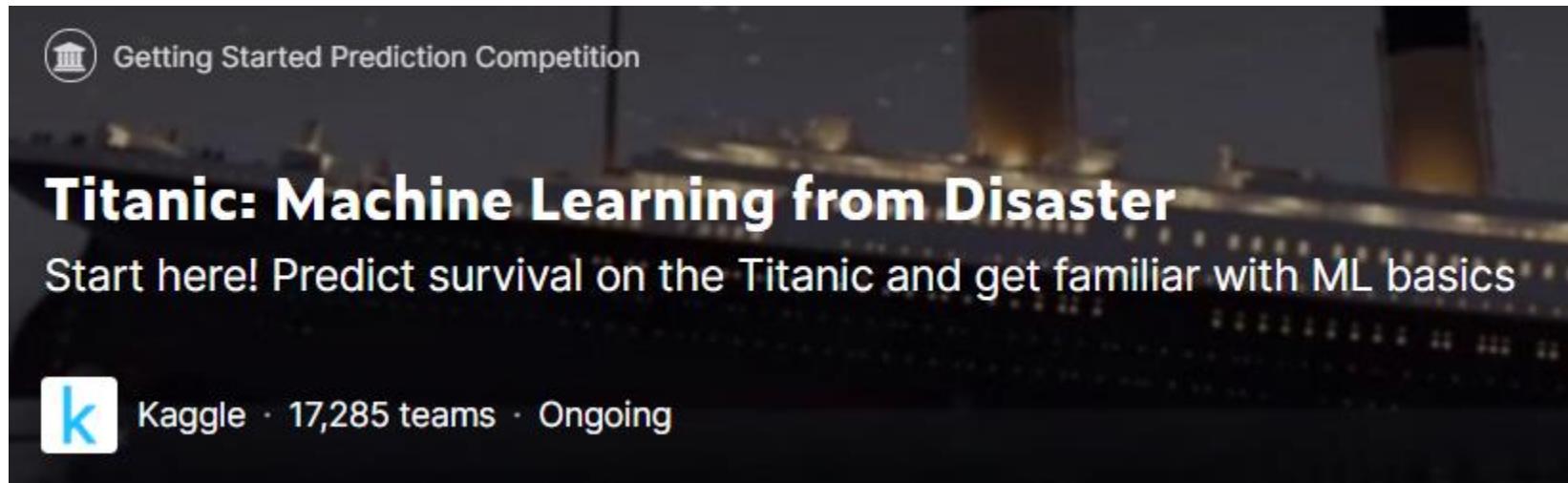
- 꾸준한 인공지능 동향에 대해 공부하는 능력
- 선 사용, 후 이해

All Your Work Shared With You Favorites Hotness ▾

Rank	User Profile	Title	Updated	Score	Tags	Hotness	Action
1		SIIM: d3 EDA, Augmentations and ResNeXt	Updated 4mo ago	0.422	medicine, exploratory data analysis, gpu	586	Gold ⚙️
2		Triple Stratified KFold with TFRecords	Notebook copied with edits from · Updated 4mo ago	0.945	tpu	545	Gold ⚙️
3		Melanoma. Pytorch starter. EfficientNet	Updated 4mo ago	0.925	gpu	403	Gold ⚙️
4		Analysis of Melanoma Metadata and EffNet Ensemble	Updated 4mo ago	0.9513	exploratory data analysis, classification, deep learning, image data, tpu	386	Gold ⚙️
5		Incredible TPUs - finetune EffNetB0-B6 at once	Updated 5mo ago	0.941	tpu	328	Gold ⚙️

어떻게 시작해야 하나요?

- 1. 경진대회 사이트 가입
- 2. Titanic 검색 (연습대회)
- 3. 대회 참가 신청



어떻게 시작해야 하나요?

■ 4. 데이터 구경해보기

- 성별, 나이, 승객등급 등을 조합하여 생존 여부를 예측해보자
- 혼자서 한번 생각해보기
 - 가정1. 영화에서 보니깐 여자와 어린이를 우선으로 구출하는 장면이 나오더라
 - 가정2. 높은 등급의 승객들이 더 많이 살아남을 것 같다.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050



어떻게 시작해야 하나요?

- 5. Notebooks 탭으로 이동 후 추천을 많이 받은 글 클릭
- 6. 솔루션 Copy and Edit 클릭

The screenshot shows the Kaggle homepage with the 'Notebooks' tab highlighted. Below the navigation bar, there's a search bar and a list of recommended notebooks. The first notebook in the list is titled 'Top 3%. Efficient ensembling in few lines of code' by a user with a yellow profile picture. It was updated 13h ago and has a score of 0.80861. The second notebook is 'Titanic EDA' by a user with a green profile picture, updated 4h ago, with a score of 0.80861. The third is 'Getting Started with Titanic Dataset' by a user with a green profile picture, updated 10h ago, with a score of 0.78947. The fourth is 'Hyperparameters tuning techniques' by a user with a yellow profile picture, updated 13h ago, with a score of 0.78708.

The screenshot shows a specific notebook titled 'Titanic Data Science Solutions' by a user with a blue profile picture. The notebook has 6567 views and 27356 copies. A red dashed arrow points from the 'Copy and Edit' button to the 'Copy and Edit' button on the left side of the slide. The notebook description states it is a Python notebook using data from 'Titanic: Machine Learning from Disaster'. It features 'feature engineering, model comparison' and is described as a companion to the book 'Data Science Solutions'.

Titanic Data Science Solutions

This notebook is a companion to the book [Data Science Solutions](#).

The notebook walks us through a typical workflow for solving data science competitions at sites like Kaggle.

There are several excellent notebooks to study data science competition entries. However many will skip some of the explanation on how the solution is developed as these notebooks are developed by experts for experts. The objective of this notebook is to follow a step-by-step workflow, explaining each step and rationale for every decision we take during solution development.

Workflow stages

The competition solution workflow goes through seven stages described in the Data Science Solutions book.

1. Question or problem definition.
2. Acquire training and testing data.
3. Wrangle, prepare, cleanse the data.
4. Analyze, identify patterns, and explore the data.
5. Model, predict and solve the problem.
6. Visualize, report, and present the problem solving steps and final solution.
7. Supply or submit the results.

어떻게 시작해야 하나요?

■ 7. 실행 후 설명 읽으며 공부

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Titanic Data Science Solutions, Draft saved.
- Toolbar:** Share, Save Version (0), Run All, Draft Session (0m), H D C P U R A M.
- Data Panel:** Data, Add data, input (90.9 KB), titanic, output, /kaggle/working.
- Code Cells:**
 - Acquire data:** The cell contains code to read CSV files into Pandas DataFrames:

```
[1]: train_df = pd.read_csv('../input/train.csv')
test_df = pd.read_csv('../input/test.csv')
combine = [train_df, test_df]
```
 - Analyze by describing data:** The cell contains code to print column values:

```
[1]: print(train_df.columns.values)
```
- Settings Panel:** Language (Python), Environment (Preferences), Accelerator (None), Internet (Off).
- Code Help Panel:** Find Code Help, Search for examples of how to do things.

어떻게 시작해야 하나요?

■ 무슨말인지 하나도 모르겠어요

- 영어가 문제라면: 번역기 + 영어공부
- 코드가 문제라면:
 - Python 혹은 R 관련 공부, 학교의 관련 커리큘럼 수강

■ 따라 해봤는데, 이해가 조금씩 되는것 같아요

- solution을 토대로 자신만의 코드를 개발
- 새로운 경진대회를 검색 후 참가

■ 어렵지만 재미있는 것 같아요!

- 함께 할만한 친구를 찾아서 같이하면 좋습니다!