KENTECH
Korea Institute of Energy Technology

Advanced Computer Vision
Week 01

Sep. 2, 2022
Seokju Lee

KENTECH
Korea Institute of Energy Technology

Human Visual System

Learning Human Brain from "Baby": The Purest Nautural Intelligence

1개월 아기의 시선,
아기의 눈은 대체로, 빛과 어둠의
경계가 되는 윤곽선에 반응을 보인다.
2개월 아기의 시선
아기들은 좀 더 선명하게 사람의
얼굴을 쳐다볼 수 있게 된다.
스스로 공간을 이동해 다니면서,
아기들은 차츰 자신이 어디에 있는지,
이번에는 경계선 앞에서
주저앉아 버렸다.

아기성장보고서, EBS

✓ **Visual capabilities:**

- “*What-path*”: object recognition
- “*Where-path*”: localization

Learning Human Brain from "Baby"

그만큼 걷는다는 것은, 아기들에게
새롭고도 대단한 경험이다.
감각을 동원해 주위에서
일어나는 일들을 관찰하고,
아기들은 세상을 이해해 나가게 된다.
아기는 자기 자신에 대해서도
깨달아가게 된다.

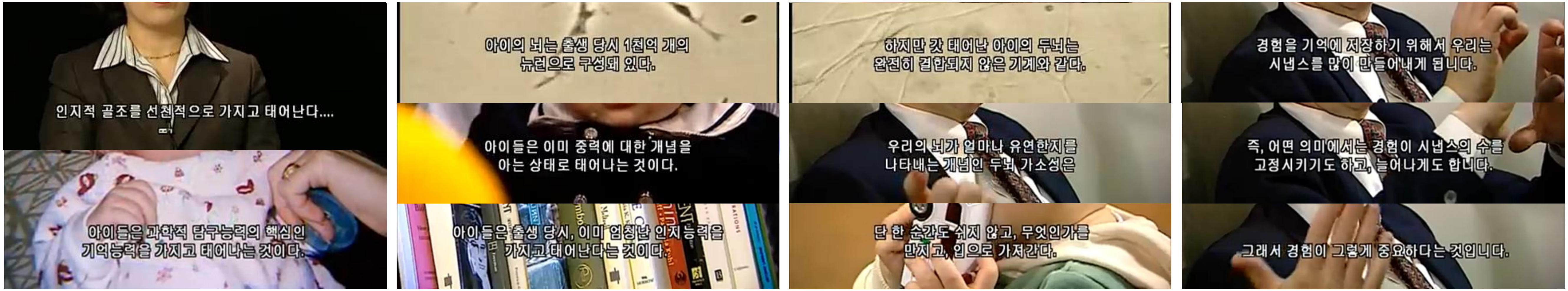
아기성장보고서, EBS

✓ **Interaction & Adaptation:**

- Environmental interaction with *vision*, *motion*, *audio*, …
- *Ego-centric* understanding ⇌ *Environmental* adaptation

Synchronized & **co-embedded**
multimodal sensing knowledge

Learning Human Brain from "Baby"



인지적 골조를 선천적으로 가지고 태어난다....

아이들은 과학적 탐구능력의 핵심인
기억능력을 가지고 태어나는 것이다.

아이의 뇌는 출생 당시 1천억 개의
뉴런으로 구성돼 있다.

아이들은 이미 중력에 대한 개념을
아는 상태로 태어나는 것이다.

아이들은 출생 당시, 이미 엄청난 인지능력을
가지고 태어난다는 것이다.

하지만 갓 태어난 아이의 두뇌는
완전히 결합되지 않은 기계와 같다.

우리의 뇌가 얼마나 유연한지를
나타내는 개념인 두뇌 가소성은

단 한 순간도 쉬지 않고, 무엇인가를
만지고, 입으로 가져간다.

경험을 기억에 저장하기 위해서 우리는
시냅스를 많이 만들어내게 됩니다.

즉, 어떤 의미에서는 경험이 시냅스의 수를
고정시키기도 하고, 늘어나게도 합니다.

그래서 경험이 그렇게 중요하다는 것입니다.

아기성장보고서, EBS

c.f., Pretrained model
& Transfer learning

- ✓ **Powerful memory & Flexible update:**

- *Innate* memory and *well-designed* cognitive process
- Life-long *abstraction* and *dynamically-updating* synapses from *experience*

Learning Human Brain from "Baby"

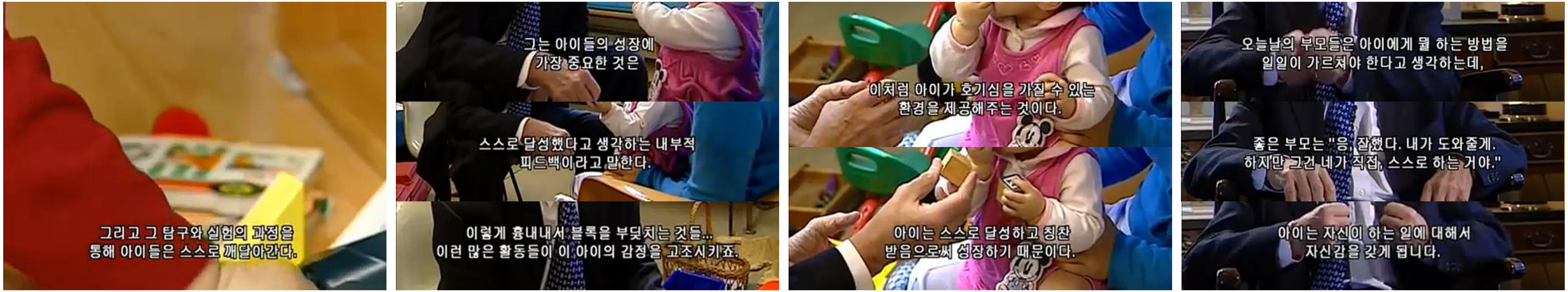
모빌 실험
갓난아이들이 사물의 인과관계를
알고 있다는 사실을 확인할 수가 있다.
그 발달하는 과정에서 아이들
나름대로 가설검증도 하거든요.
수 없는 시행착오와 실험 끝에 터득해낸
방법으로, 명환이는 결국 목표를 달성했다.
굳이 가르쳐주지 않아도 아이들은
스스로 문제를 해결할 능력이 있다.
인형도 함께 딸려온다는 사실을
발견한 것이다.

아기성장보고서, EBS

✓ **Reasoning & Learning:**

- Reasoning based on *uncertain* and *noisy* information
- Learning *functional* and *contextual* information

Learning Human Brain from "Baby"

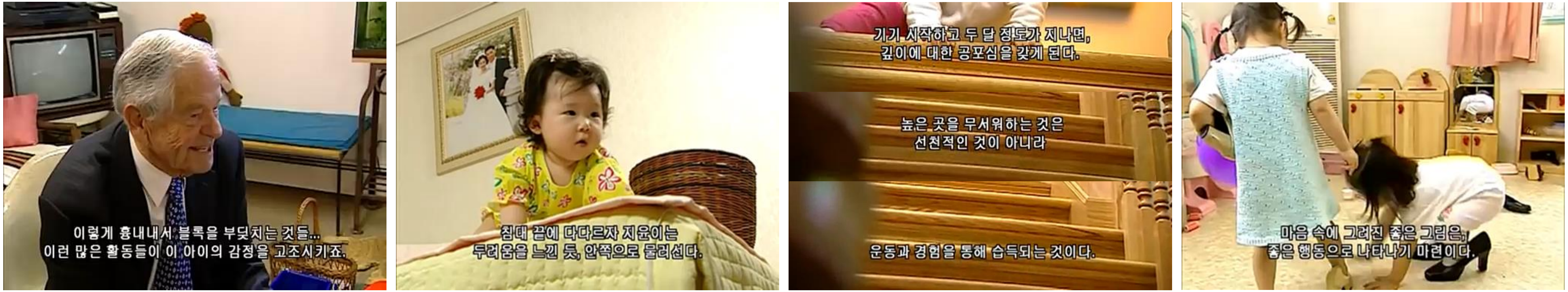

그리고 그 탐구와 실험의 과정을
통해 아이들은 스스로 깨달아간다.
그는 아이들의 성장에
가장 중요한 것은
스스로 달성했다고 생각하는 내부적
피드백이라고 말한다.
이렇게 흉내내서 블록을 부딪치는 것들...
이런 많은 활동들이 이 아이의 감정을 고조시키죠.
이처럼 아이가 호기심을 가질 수 있는
환경을 제공해주는 것이다.
아이는 스스로 달성하고 칭찬
받음으로써 성장하기 때문이다.
오늘날의 부모들은 아이에게 뭘 하는 방법을
일일이 가르쳐야 한다고 생각하는데,
좋은 부모는 "응, 잘했다. 내가 도와줄게.
하지만 그건 네가 직접, 스스로 하는 거야."
아이는 자신이 하는 일에 대해서
자신감을 갖게 됩니다.

아기성장보고서, EBS

✓ **Active & Self-supervised learning:**

- *Curiosity-driven* active learning
- *Reward-driven* self-supervised learning

Learning Human Brain from "Baby"


이렇게 흉내내서 블록을 부딪치는 것들...
이런 많은 활동들이 이 아이의 감정을 고조시키죠.
침대 끝에 다다르자 지윤이는
두려움을 느낀 듯, 안쪽으로 물러선다.
기기 시작하고 두 달 정도가 지나면,
깊이에 대한 공포심을 갖게 된다.
높은 곳을 무서워하는 것은
선천적인 것이 아니라
운동과 경험을 통해 습득되는 것이다.
마음 속에 그려진 좋은 그림은,
좋은 행동으로 나타나기 마련이다.

아기성장보고서, EBS

Could be one of the most difficult challenges for AI

✓ **We Feel, therefore We Learn:**

- "*Emotions* are inherently linked to and influence *cognitive skills* such as attention, memory, executive function, decision-making, critical thinking, problem-solving and regulation, all of which play a key role in *learning*." [1]

[1] M. H. Immordino-Yang, et al., "We Feel, Therefore We Learn: The Relevance of Affective and Social Neuroscience to Education" Mind, Brain, and Education 2007.

KENTECH
Korea Institute of Energy Technology

How about Robotic Intelligence so far?

Task-Specific Perception

✓ **Computer vision:**

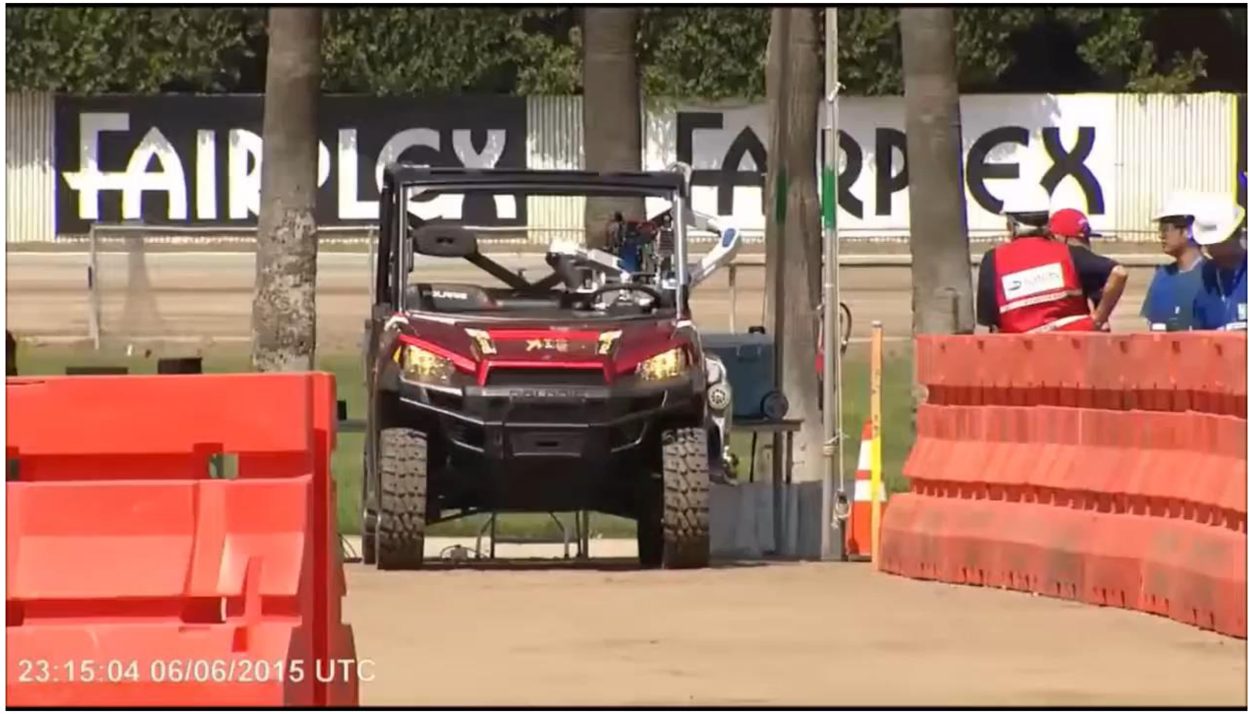
- Traditional approaches for 3D pose estimation and segmentation

✓ **Human operator:**

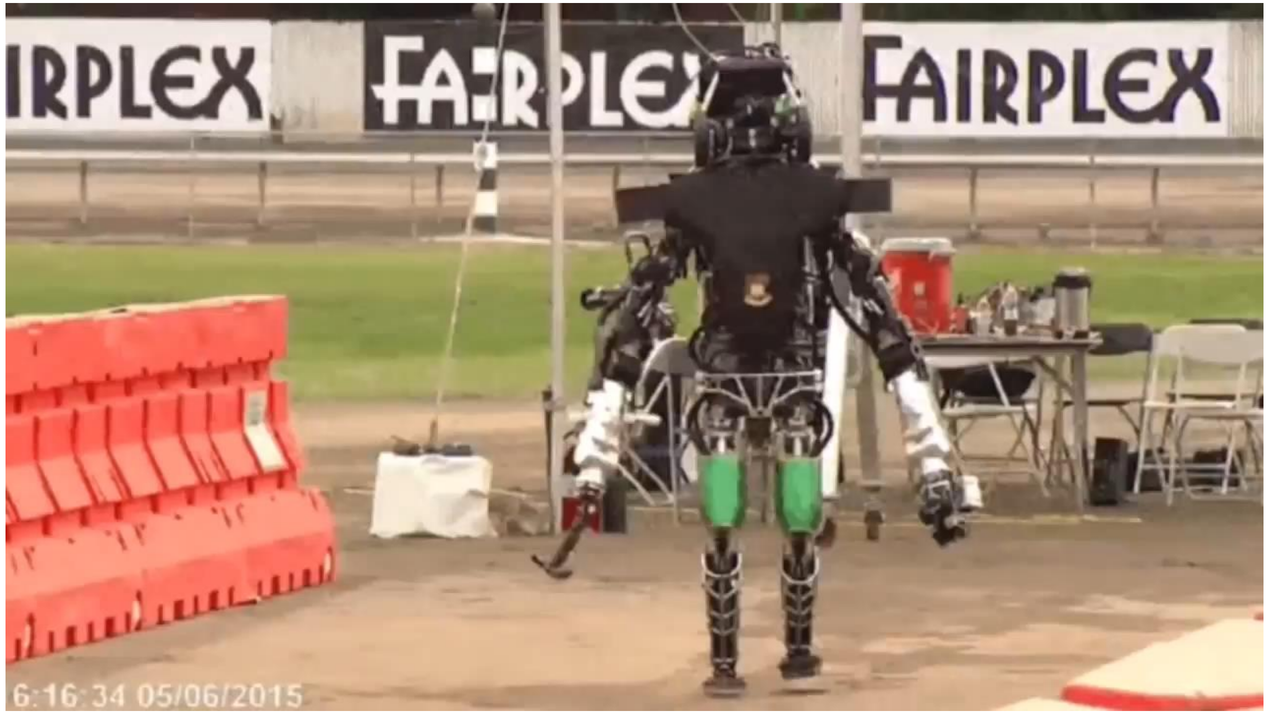
- Planning & Decision making

Lack of Reasoning!

Please focus on the **moment** when the robot **fails**! → **high uncertainty**



KAIST Wins DARPA Robotics Challenge (2015)



Fails compilation

Toward Human-like Intelligence

✓ **Reinforcement learning for recognition and manipulation:**

- Different manipulation strategies for object shapes

✓ **Two “Deep Nets” by 800,000 training:**

- One net for predicting the grasping
- Another net for estimating the effectiveness of grasping (*meta-learning* approach)

“Learning to learn”

처음엔 손바닥 전체로 덮치듯 잡다가
10-11개월이 되면,
엄지와 집게손가락만을 이용해
물체를 집게되는 것이다.

기계와의 대결
KBS1
보통은 로봇에게 규칙을 주지만
이번에는 스스로 학습하게 했습니다

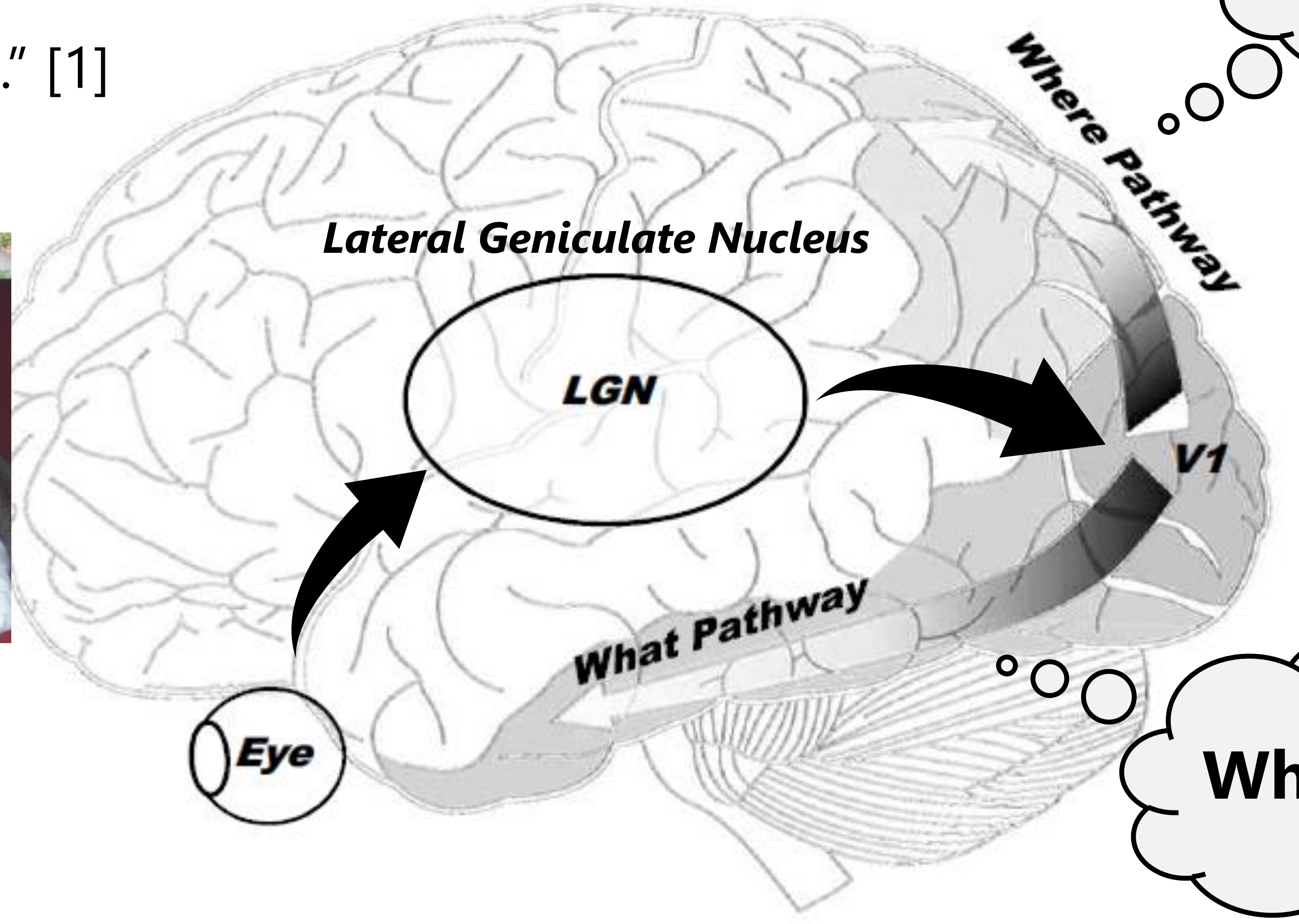
기계와의 대결, KBS (14:10~, Google AI Blog, 2016)

KENTECH
Korea Institute of Energy Technology

Back to Human Visual System

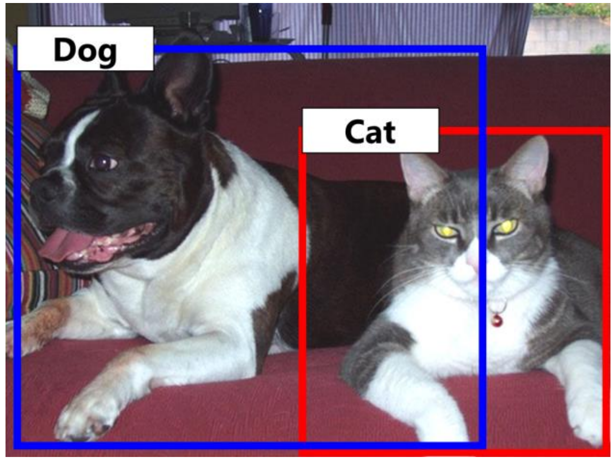
Human Visual System (HVS)

"More than **half** of neocortex in humans is devoted to **vision**." [1]



Where is it?

What is it?



[1] Barton, Robert A. "Visual specialization and brain evolution in primates." *Proceedings of the Royal Society of London* (1998).
[2] M. A. Goodale, et al., "Separate visual pathways for perception and action." *Trends in Neurosciences* (1992).

HVS: Very Robust to Any Changes

Scale change

3D viewpoint variation

Illumination change

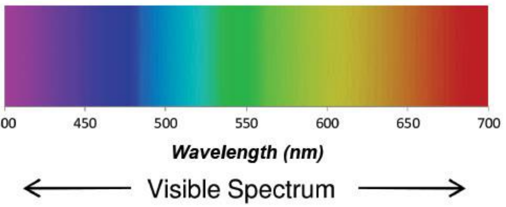
Environmental change

Image blur, occlusion, clutters, …

Characteristics of Human Eye

✓ **The human eye is sensitive to only a small set of wavelengths:**

- Rods (간상세포, B/W) and Cones (원추세포, Color)

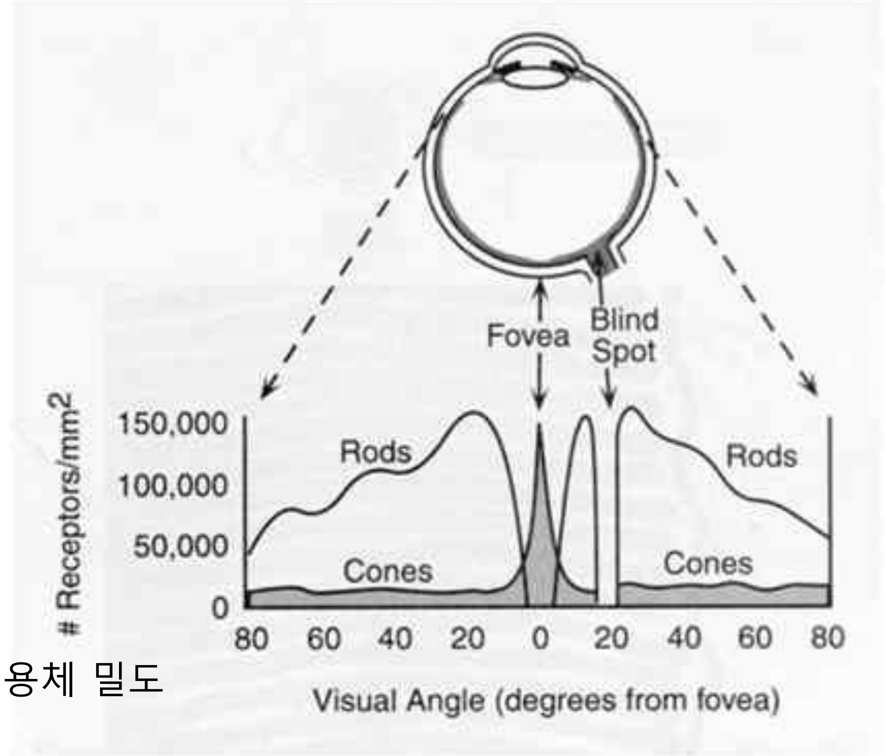


✓ **The pigments in the cones have peak sensitives at:**

- Red (650nm): L-cone
- Green (530nm): M-cone
- Blue (425nm): S-cone
- The sensitivities are independent.

✓ **The human eye can distinguish about 380,000 colors:**

- 128 Hues (색상, color)
- 130 Tints (채도, the addition of white, saturation)
- 16-23 Shades (명도, the addition of black)



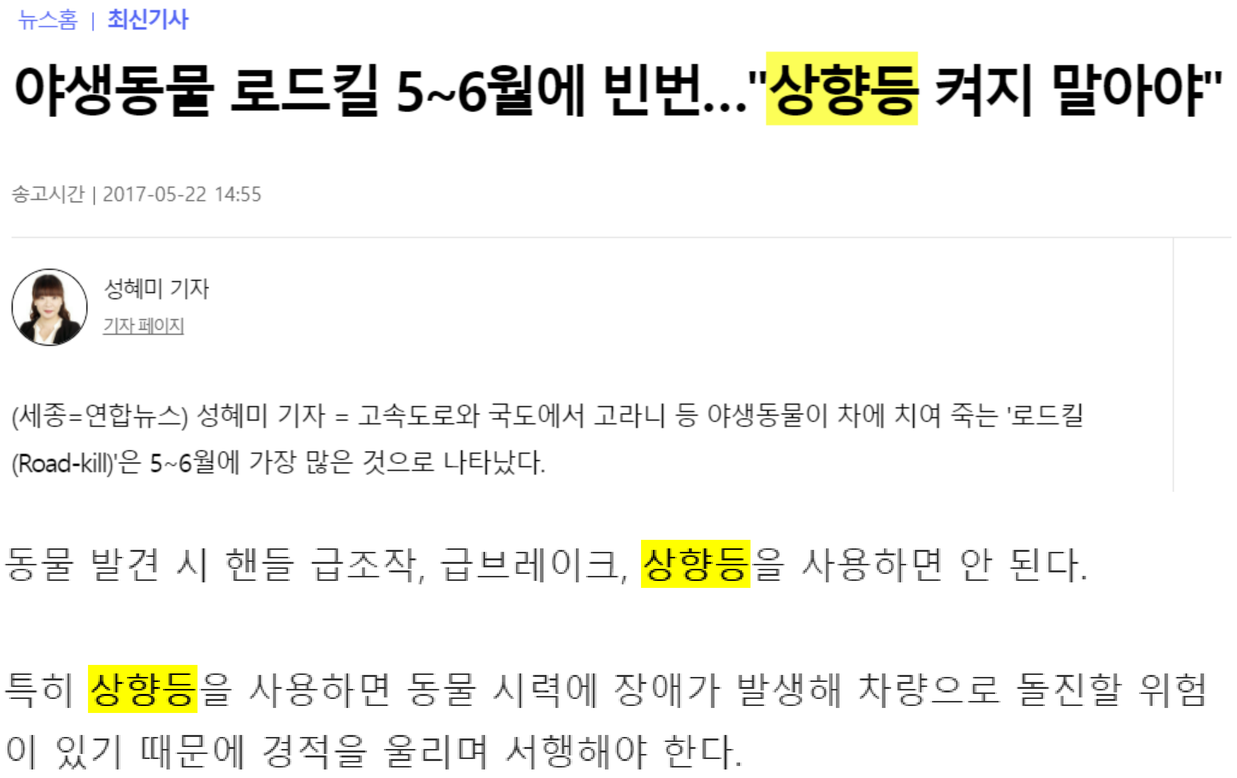
수용체 밀도

→ **Rods are more sensitive to light than cones**

Mechanism of Evolution: Natural Selection

Rods are more **sensitive to light** than cones

→ Easier to **adapt to dark** environments!



뉴스홈 | 최신기사

야생동물 로드킬 5~6월에 빈번..."상향등 켜지 말아야"

송고시간 | 2017-05-22 14:55

성혜미 기자
기자 페이지

(세종=연합뉴스) 성혜미 기자 = 고속도로와 국도에서 고라니 등 야생동물이 차에 치여 죽는 '로드킬(Road-kill)'은 5~6월에 가장 많은 것으로 나타났다.

동물 발견 시 핸들 급조작, 급브레이크, 상향등을 사용하면 안 된다.

특히 상향등을 사용하면 동물 시력에 장애가 발생해 차량으로 돌진할 위험이 있기 때문에 경적을 울리며 서행해야 한다.

Mechanism of Evolution: Natural Selection

Rods are more **sensitive to light** than cones

→ Easier to **adapt to dark** environments!

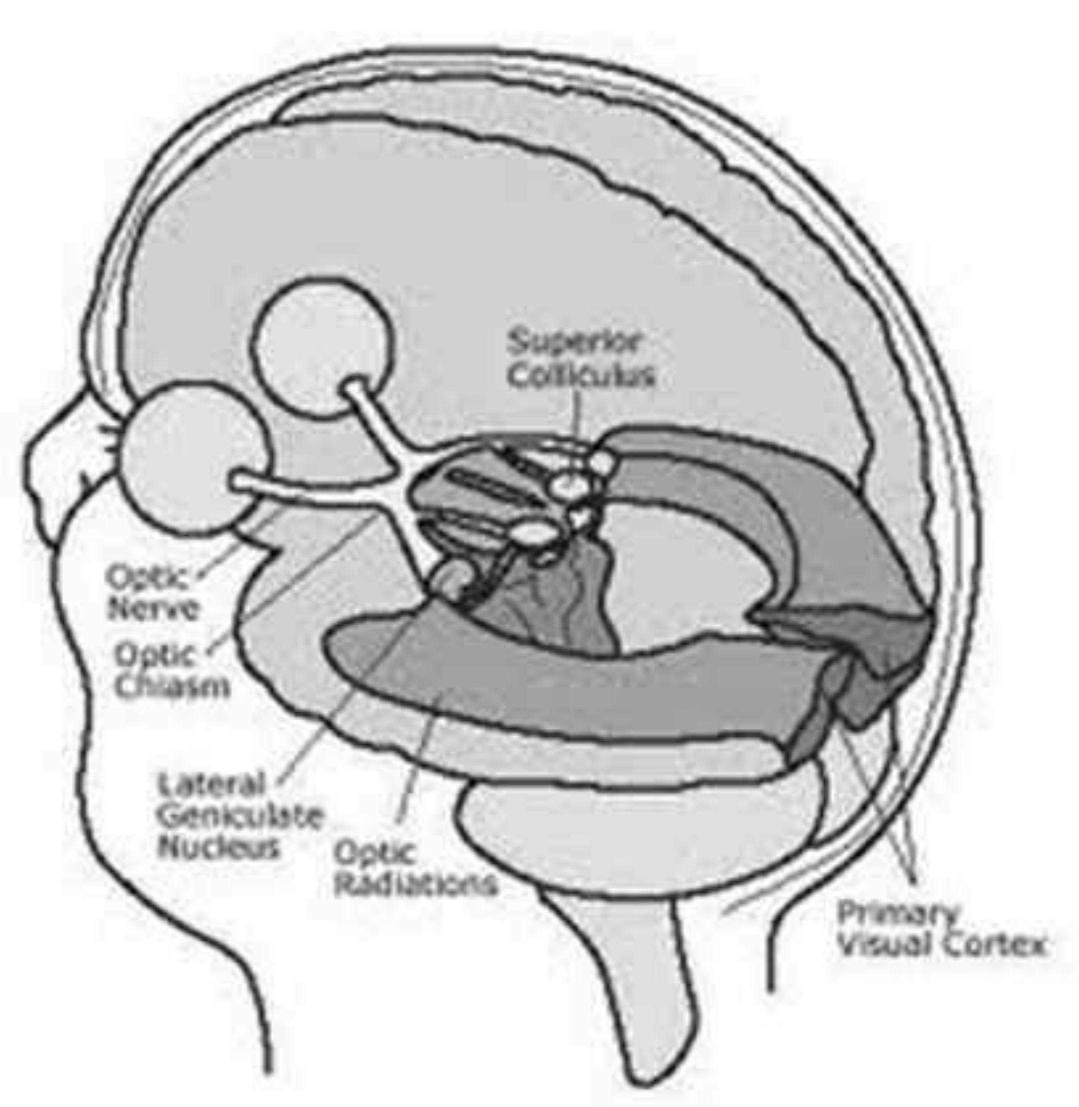
Dual camera approach:
One for cones, and the other one for rods

iPhone 11 - Night Mode (2020)

HVS: Where-Path

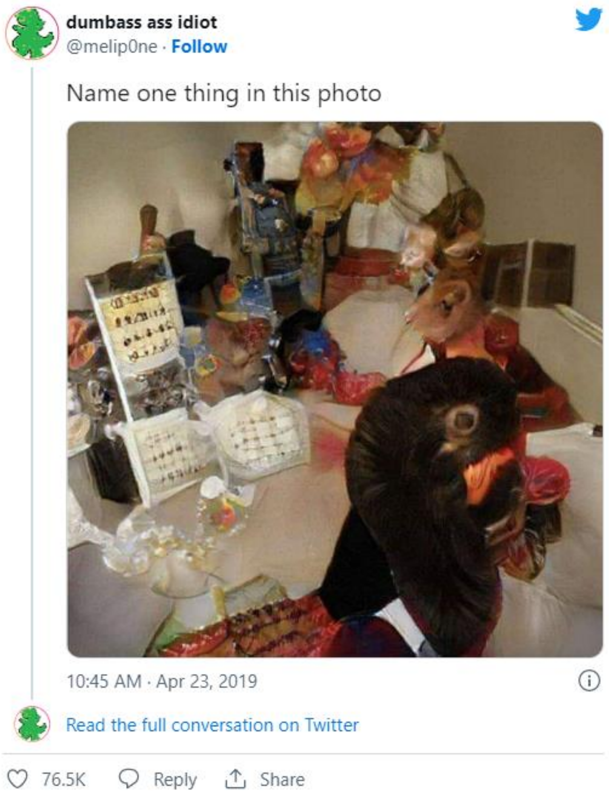
✓ **Integration of multiple cues**

1. Stereo: depth by two eyes
2. Motion
3. Shading
4. Perspective projection
5. Focus
6. Shadow
7. Position relative to the horizon
8. Relative size
9. Familiar size
10. Texture gradients
11. Aerial perspective
12. Superposition



HVS: What-Path

Q. Can you recognize anything in this picture?



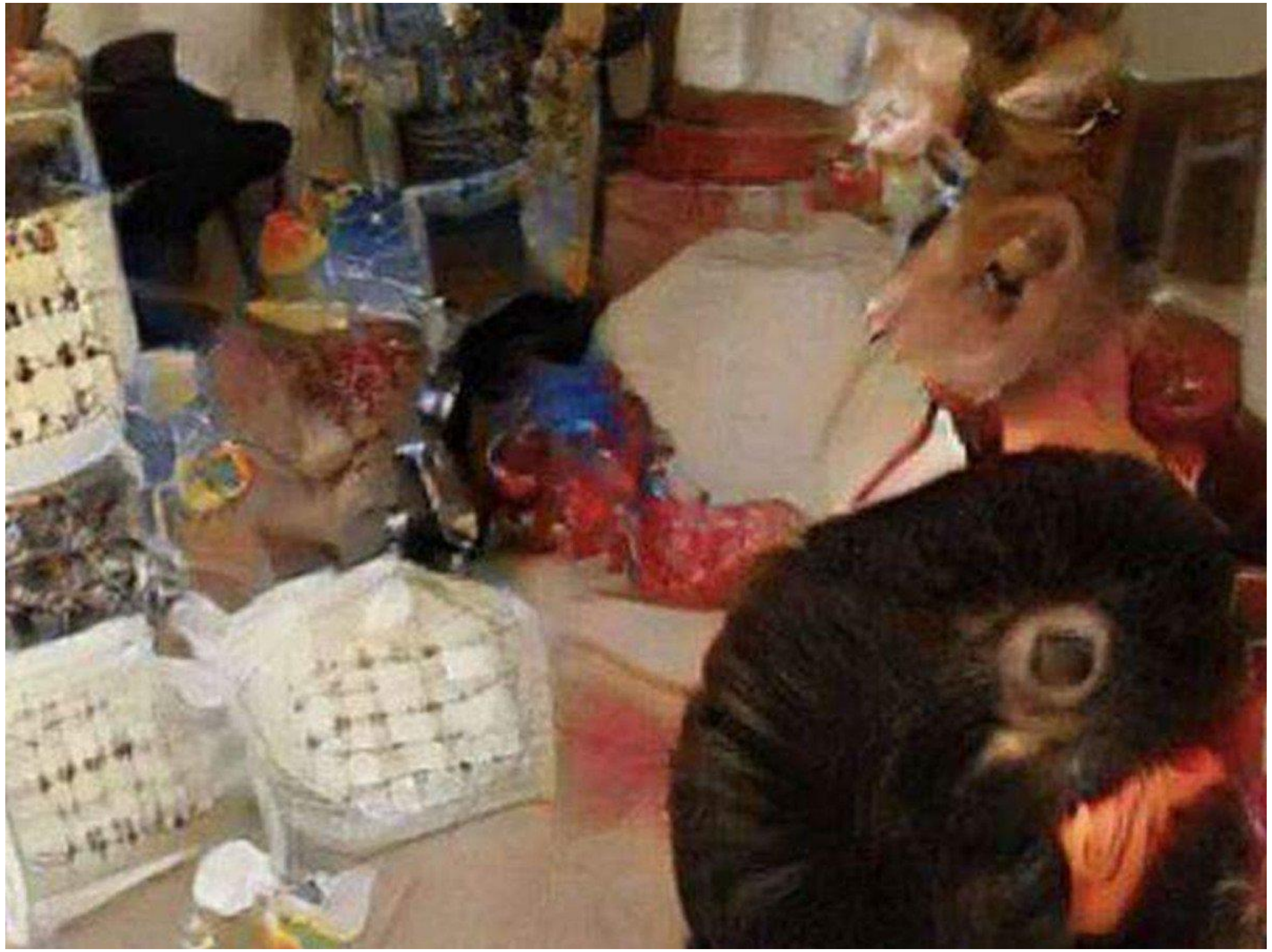
dumbass ass idiot
@melip0ne · Follow

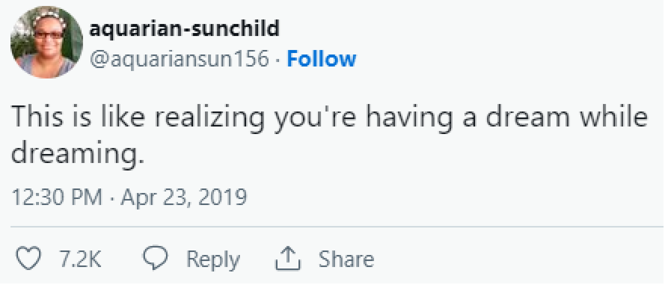
Name one thing in this photo

10:45 AM · Apr 23, 2019

Read the full conversation on Twitter

♡ 76.5K Reply Share



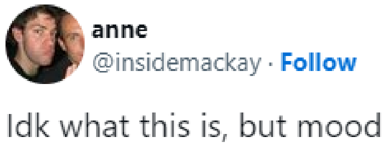


aquarian-sunchild
@aquariansun156 · Follow

This is like realizing you're having a dream while dreaming.

12:30 PM · Apr 23, 2019

♡ 7.2K Reply Share



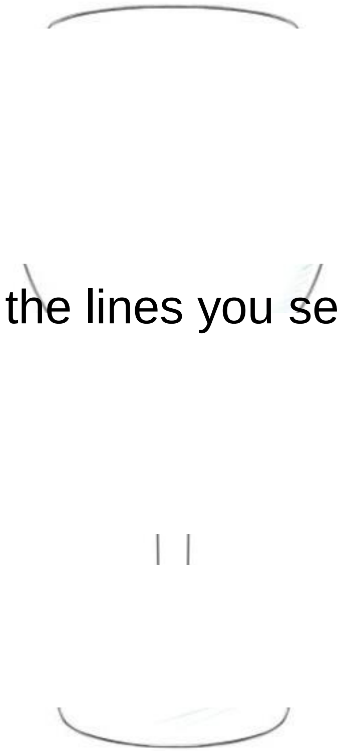
anne
@insidemackay · Follow

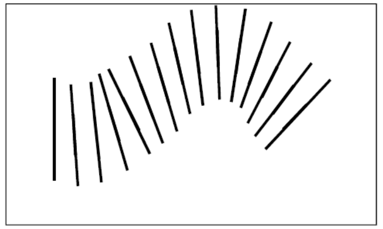
Idk what this is, but mood

→ People can't figure out what is happening in this photo.

HVS: What-Path

- ✓ Object is defined by the **presence** or **absence** of **local** visual properties.



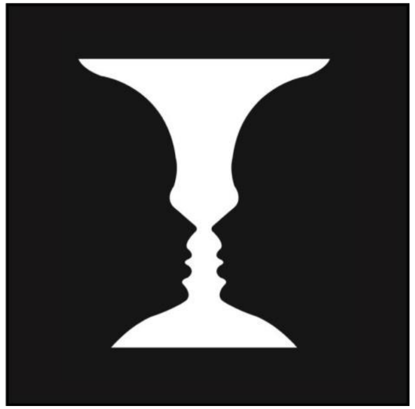


Your HVS constructs all the lines you see and **something more**!

AI Draws Images from Sketch

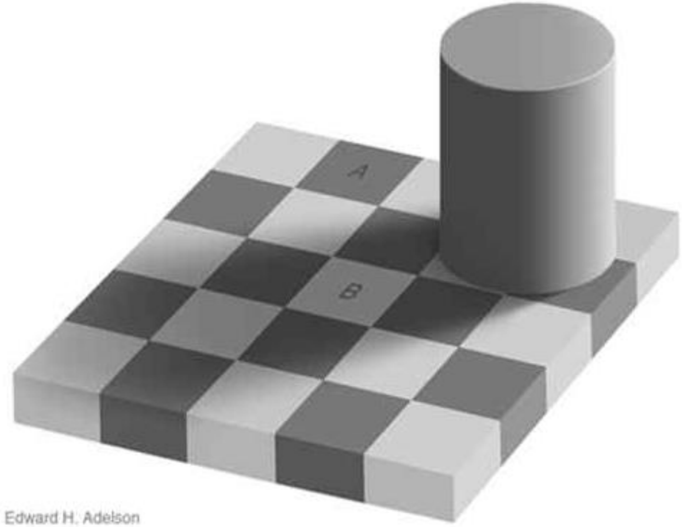
- ✓ Category is described by **object parts** and their **contextual relations**.

- ✓ Object is linked with **visual attention** and **figure-ground** (형태-배경 조직화).

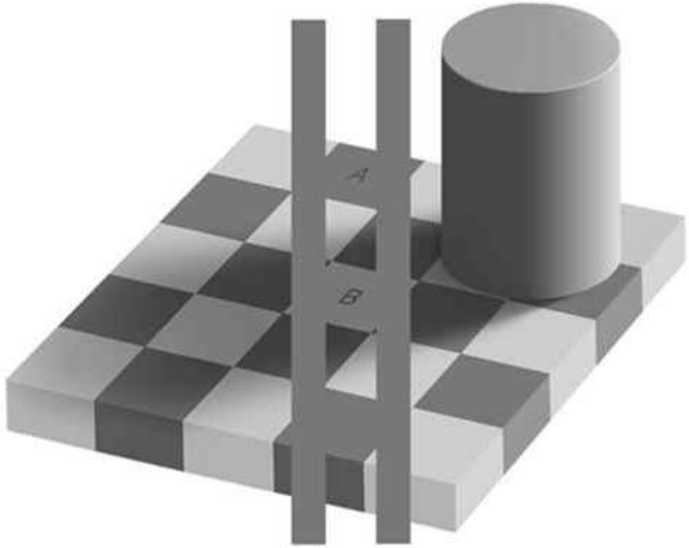


Human Visual System (HVS): Summary

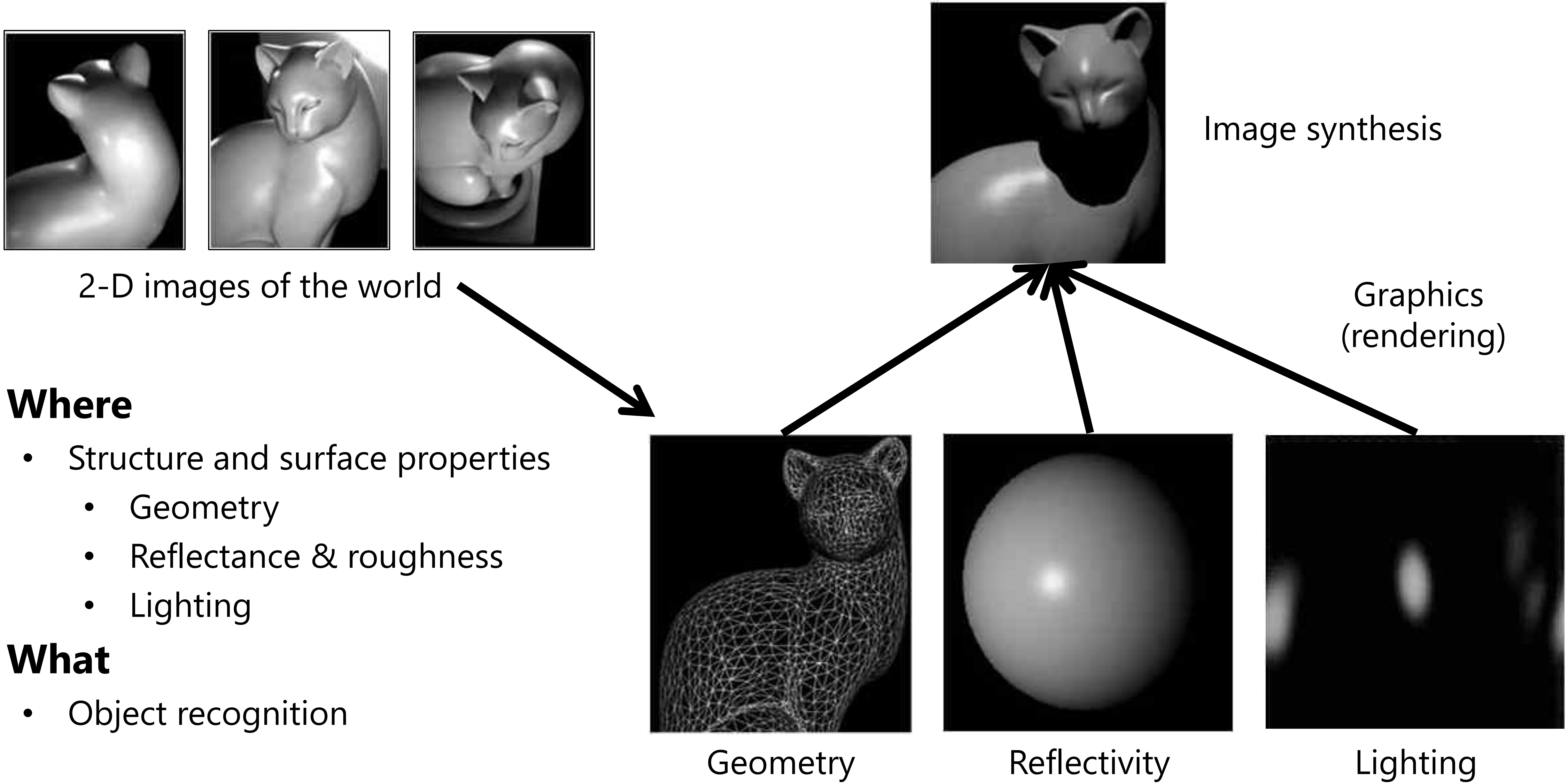
- **We see effortlessly**
  - 70% of our brains for visual perception
  - Perceiving "the visual world as it is"?
  - A reconstruction of the visual world within our brains
  - Relatively large Field of View (FoV) (200 x 135°)
- **Some limitations of HVS**
  - Subject to illusions
  - Quantitatively imprecise
  - Limited to a narrow range of frequencies of radiation
    - Less than 0.1% of the energy that reaches our eyes.
  - Passive
    - Some active systems: bat acoustic imaging systems



Edward H. Adelson



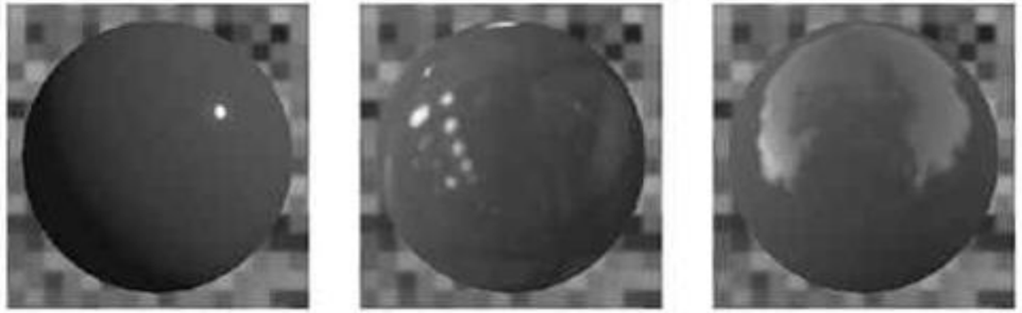
What is "Computer Vision"?



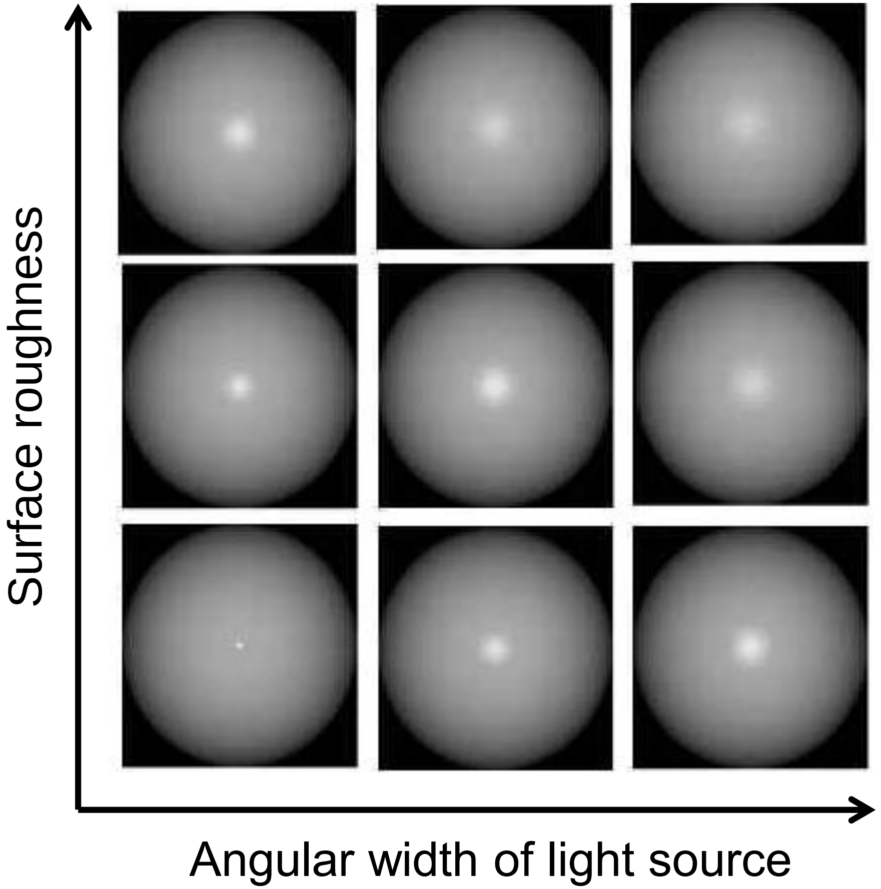
- **Where**
  - Structure and surface properties
    - Geometry
    - Reflectance & roughness
    - Lighting
- **What**
  - Object recognition

Why is Computer Vision is Difficult?

- ✓ **Many factors influencing the image intensity:**
  - Lighting, geometry, reflectivity, sensor, surface roughness, and etc.



- **Ambiguities:**
  - Photometric
  - Geometric: 3-D to 2-D projection



- **Intractability:**
  - Huge amounts of data
  - High computational complexity

KENTECH
Korea Institute of Energy Technology

Image Processing with Python

Python Functions

How to define functions?

→ Plot image (show_image) & close the plot (close_image)

```
def show_image(i, img):
        plt.figure(i)
        plt.imshow(img)
        plt.xticks([]); plt.yticks([])
        plt.ion(); plt.show()

def close_image(i):
        if i == 0:
                plt.close('all’)
        else:
                plt.close(i)
```

```
def do_sum(a, b):
        output = a + b
        return output

def do_subtract(a, b):
        output = a - b
        return output

def do_multiply(a, b):
        output = a * b
        return output
```

Basic Python Library: NumPy

Operations

→ There are many strong operation functions for “**multi-dimensional arrays**”

```
import numpy as np

a = np.array([1,2,3])
print(a)          # [2 3 4]
print(a.dtype)    # int64
b = np.array([1.2, 3.5, 5.1])
print(b.dtype)    # float64
print(a**2)
print(a.sum())
print(a.mean())
print(a.min())
print(a.max())
```

Please try basic math operations.

→ Matrix operation, linear algebra

```
a = np.arange(8)
print(a)          # [0 1 2 3 4 5 6 7]
b = a.reshape(2, 4)
print(b)
c = a.T
print(b)

print(a.shape)
print(b.shape)
print(c.shape)
print(b.sum(axis=0))
print(b.sum(axis=1))
```

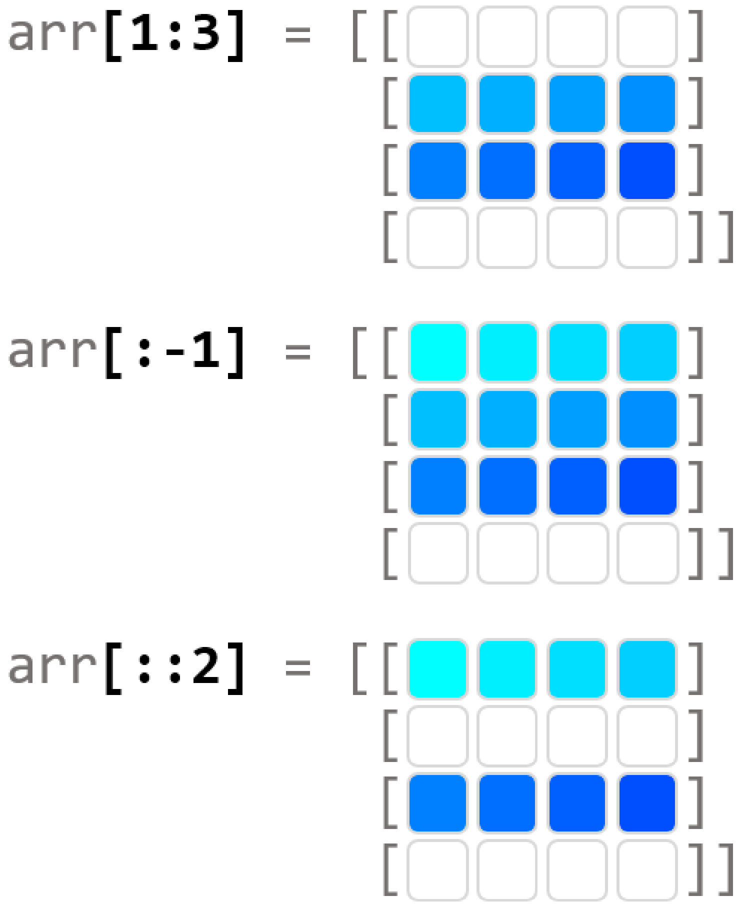
Please discuss each print line.

Basic Python Library: NumPy

Array slicing

→ Along the first axis

```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12],
                [13, 14, 15, 16]])

print(arr)
print(arr[1:3])
print(arr[:-1])
print(arr[::2])
```



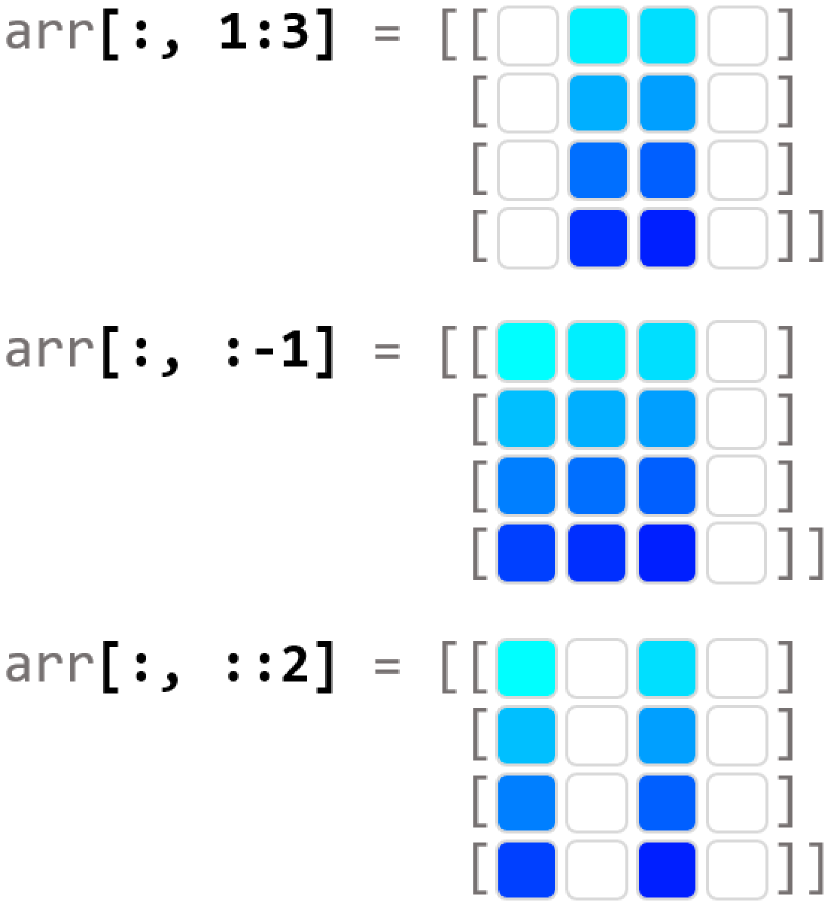
[1] codetorial.net

Basic Python Library: NumPy

Array slicing

→ Along the second axis

```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12],
                [13, 14, 15, 16]])

print(arr)
print(arr[:, 1:3])
print(arr[:, :-1])
print(arr[:, ::2])
```



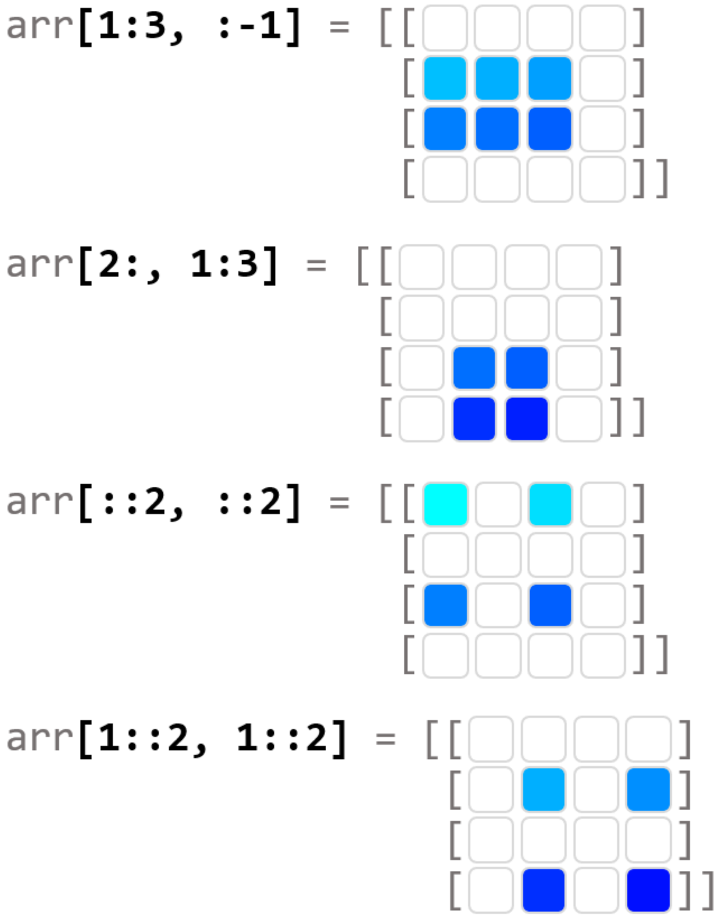
[1] codetorial.net

Basic Python Library: NumPy

Array slicing

→ Along the fist and second axis

```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12],
                [13, 14, 15, 16]])

print(arr)
print(arr[1:3, :-1])
print(arr[2:, 1:3])
print(arr[::2, ::2])
print(arr[1::2, 1::2])
```



[1] codetorial.net

Basic Python Library: NumPy

Changing specific values

→ Can be applied with a condition

```
import numpy as np

a = np.array([[1, 2], [3, 1]])
b = np.where(a == 1, 10, a)

print(a)
print(b)
```

```
import numpy as np

a = np.array([[0.5, 1.2, 0.9], [1.1, 0.8, 1.4]])
b = np.where(a < 1.0, 0.0, a)

print(a)
print(b)
```

[1] codetorial.net

Basic Python Library: NumPy

Concatenate array

→ Along different axis

```
import numpy as np

a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
c = np.array([7, 8, 9])

ab = np.concatenate((a, b))
abc = np.concatenate((a, b, c))

print(ab)
print(abc)
```

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])

ab_0 = np.concatenate((a, b), axis=0)   # Default
ab_1 = np.concatenate((a, b), axis=1)

print(ab_0)
print(ab_1)
```

[1] codetorial.net

Basic Python Library: NumPy

Stack array

→ Along different axis

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
c = np.hstack([a, b])
d = np.vstack([a, b])

print(a)
print(b)
print(c)
print(d)
```

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
c = np.stack([a, b], axis=0)
d = np.stack([a, b], axis=1)

print(a)
print(b)
print(c)
print(d)
```

Difference between
concatenate and stack?

Basic Python Library: NumPy

Broadcasting

→ Flexible operations

```
import numpy as np

array1 = np.array([1, 2, 3, 4]).reshape(2, 2)
array2 = np.array([1.5, 2.5])

add = array1 + array2

print(add)
```

What are the rules to enable broadcasting?

Basic Python Library: NumPy

Append array

→ Along different axis

```
import numpy as np

arr = np.array([[ [1, 1], [2, 2] ],
                [ [3, 3], [4, 4] ]])
item = np.array([ [5, 5], [6, 6] ])

print(arr.shape)
print(item.shape)

append = np.append(arr, item.reshape(1, 2, 2), axis=0)

print(append)
```

KENTECH
Korea Institute of Energy Technology

Image Processing Puzzle

Summary of Previous Lesson

Basics of Python functions

→ How to define and use

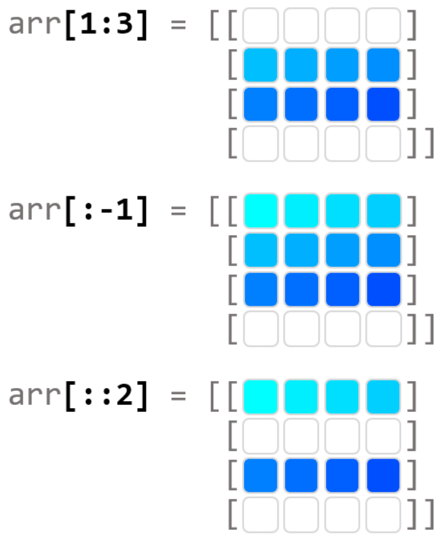
Basics of NumPy arrays

→ Many useful operations for multi-dimensional arrays

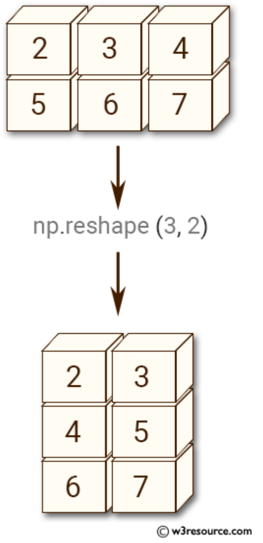
→ Please always check the current shape of arrays using the **shape** method!

→ Basic math operations: add, subtract, min, max, mean, etc. along different axis

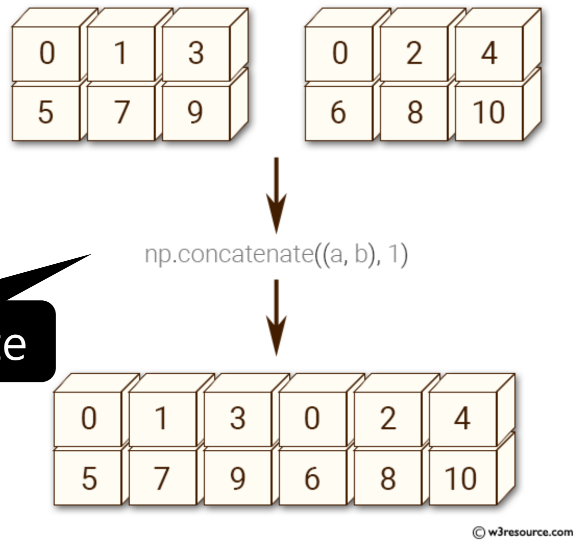
→ Basic transformations: slicing, reshape, concatenate, stack, append, etc.



slice



reshape



concatenate

[1] w3resource.com

Now, Let's Play with Images!

Solve image processing puzzles.

→ Multiple image processing missions to make specific images.

→ If you need, please refer below pages.

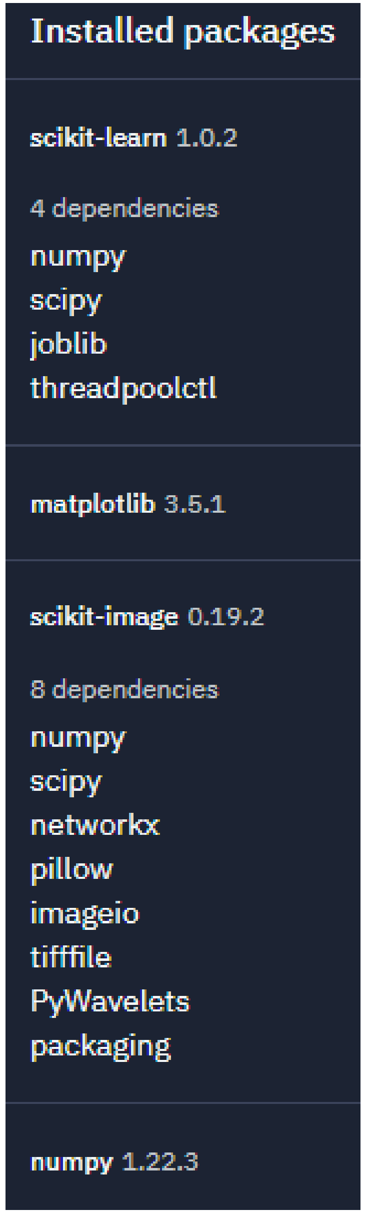
Numpy: https://numpy.org/doc/stable/reference/

Scikit-learn image: https://scikit-image.org/

PIL image: https://pillow.readthedocs.io/en/stable/reference/Image.html

Codes are available at:

https://view.kentech.ac.kr/a6fac1a2-7304-409e-85f0-78b1d527034f



Installed packages

scikit-learn 1.0.2

4 dependencies

numpy
scipy
joblib
threadpoolctl

matplotlib 3.5.1

scikit-image 0.19.2

8 dependencies

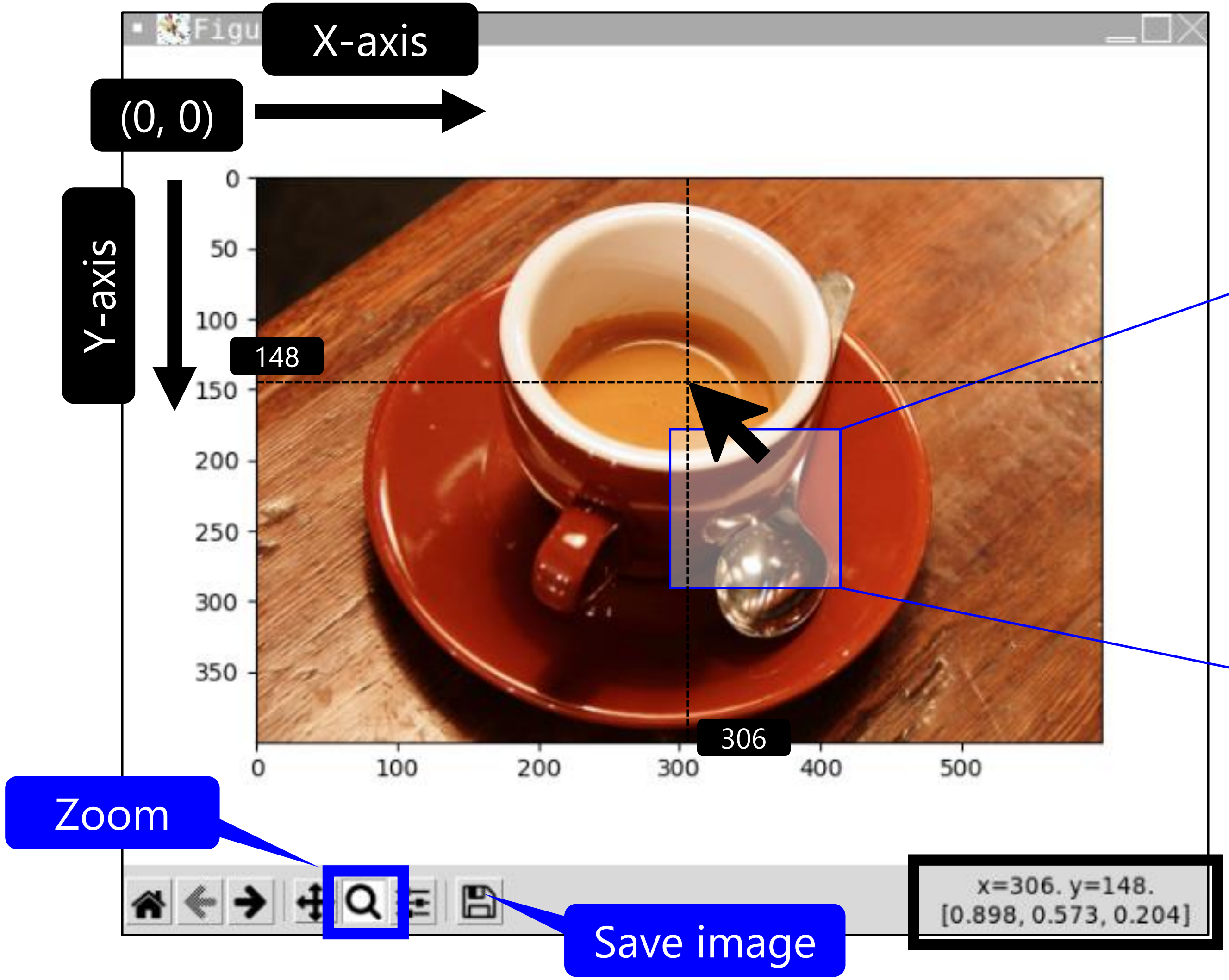
numpy
scipy
networkx
pillow
imageio
tifffile
PyWavelets
packaging

numpy 1.22.3

Image Processing Puzzle 1

Now we will try visualization

→ Enables image debugging



```
x=306. y=148.
[0.898, 0.573, 0.204]
```

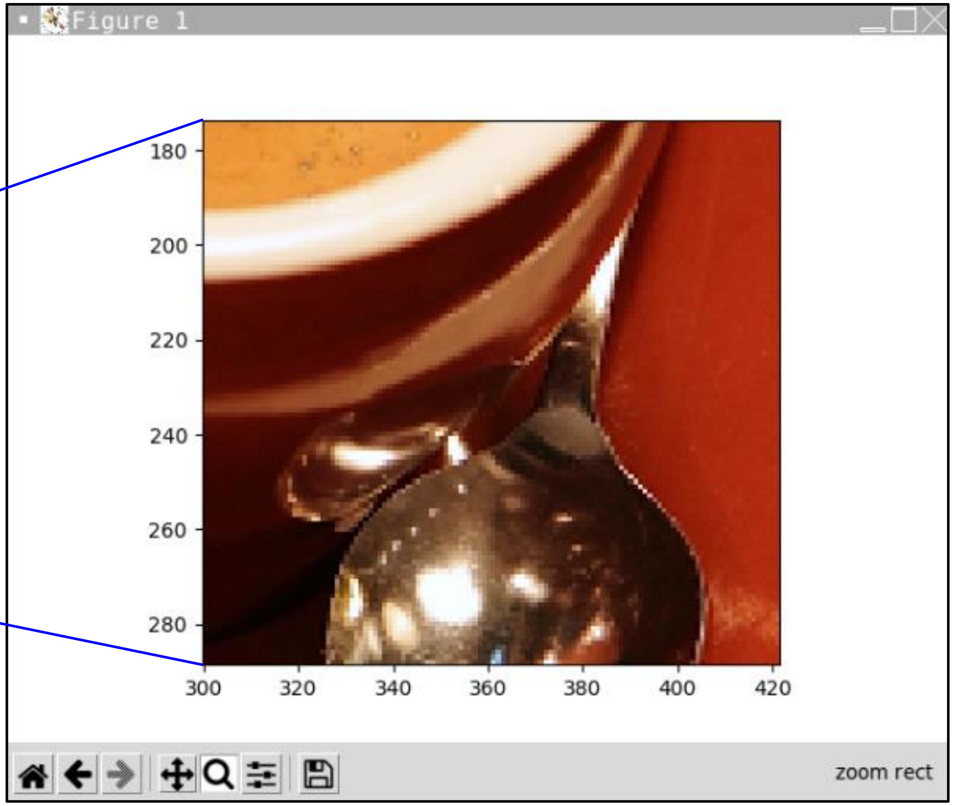


Image Processing Puzzle 1

Discussion 2: shifting image over x-axis

→ Use loop to generate the target image

```
hh, ww, ch = im.shape

for i in range(ww):
        print('index:', i)
        im1 = np.zeros(im.shape)
        im1[:, i:, :] = im[:, :ww-i, :]
        if (im1 == gt1).all():
                break;

check_same(gt1, im1)
```

Check the shape

Search space: maximum width

Initialize

Assign

Break the loop if the target is matched

Check if the output is correct

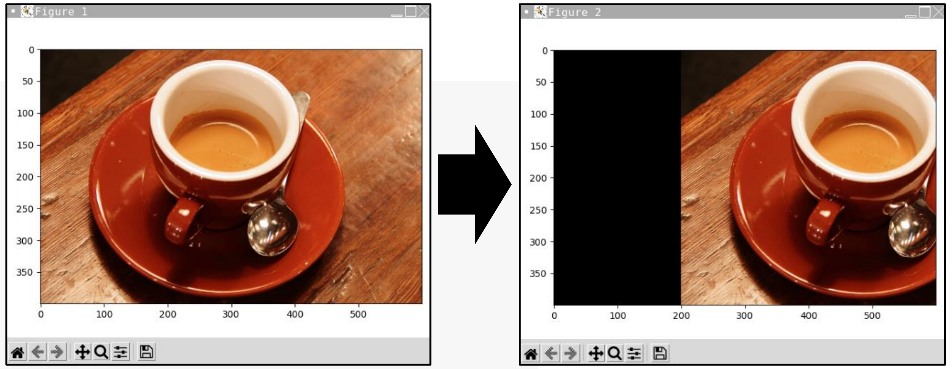


Image Processing Puzzle 1

Discussion 2-2: shifting over both x- and y-axis

→ Use loop to generate the target image → Too slow!

```
flag = False
hh, ww, ch = im.shape
for i in range(hh):
  for j in range(ww):
    print('index:', i, j)
    im1 = np.zeros(im.shape)
    im1[i:, j:, :] = im[:hh-i, :ww-j, :]
    error = np.abs(im1 - gt1).mean()
    if error == 0:
      flag = True
      break;
  if flag:
    break;
check_same(gt1, im1)
```

Search space: both maximum height + width
Use "**nested loop**"

You can check the processing time using the below code lines:

```
import time
start = time.time()
…
print(time.time() - start)
```

Image Processing Puzzle 1

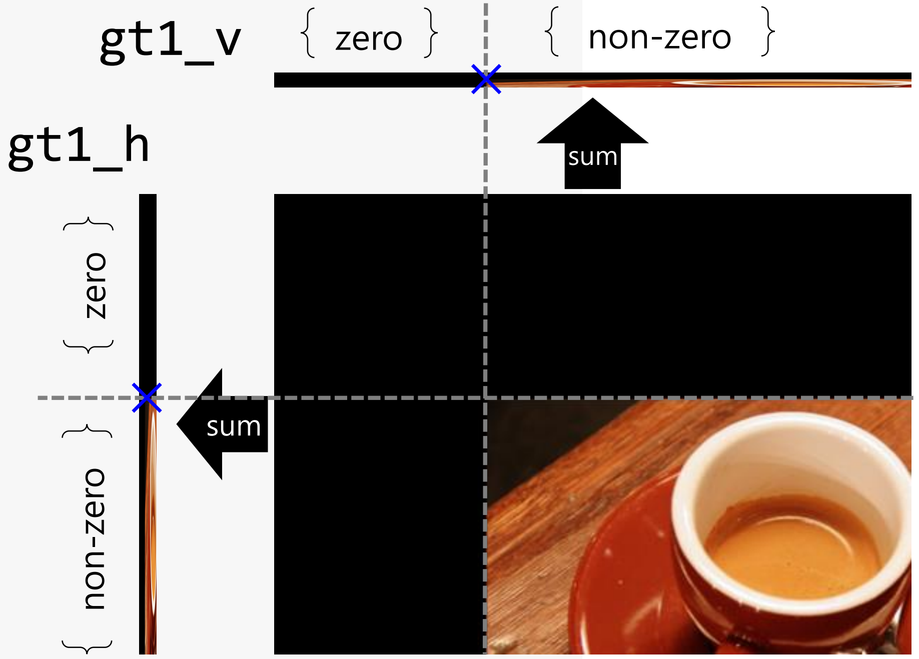
Discussion 2-2: shifting over both x- and y-axis

→ Dimension reduction. Please discuss this code in your report!

```
hh, ww, ch = im.shape

gt1_h = gt1.sum(axis=(0,2))
gt1_v = gt1.sum(axis=(1,2))

jj = (gt1_h == 0).sum()
ii = (gt1_v == 0).sum()

im1[ii:, jj:, :] = im[:hh-ii, :ww-jj, :]

check_same(gt1, im1)
```

Dimension reduction

Automatic binary sum
(= count the number of zeros)



→ Once you are familiar with the “`for-loop`” statement,

Try to implement your code **efficiently** through **dimension reduction!**

Image Processing Puzzle 2

Discussion 4: Image multiplication

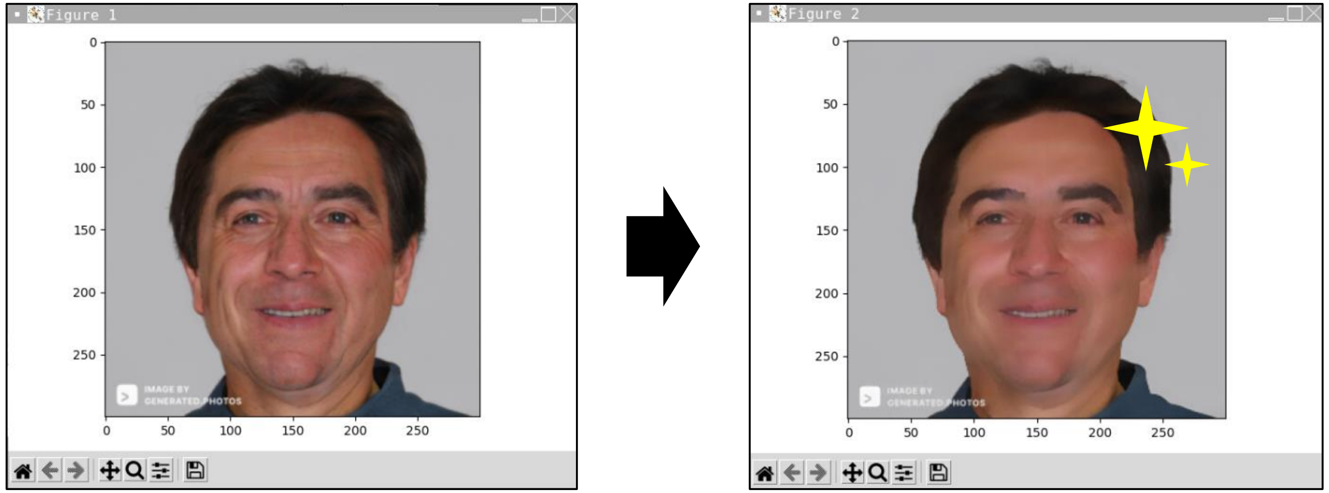
→ Multiply a binary mask to black out unnecessary parts.

Discussion 5: Image cropping and resizing

→ Crop a specific region with slicing, and resize the image.

Discussion 6: Photoshop - bilateral filter

→ Let's see how to do “photoshop” with a basic filtering algorithm.



Randomly generated face image (AI model) from https://generated.photos/face-generator/new

Discussion

<Image Processing Puzzle 2>

Please paste your result images (screenshot or save) in this report if needed.

### Discussion 2-3: Shifting image ###

2.7. Try yourself! Please try to use “dimension reduction”.

2.8. Please visualize your output image.

### Discussion 4 - Image multiplication ###

4.1. Please load a ground truth (GT) array from 'samples/puzzle2/gt1.npy'

4.2. Please convert 'im' to look like 'gt1' by multiplying the binary mask.

4.3. What is broadcasting?

4.4. What are the rules for allowing broadcasting?

### Discussion 5: Image cropping and resizing ###

5.1. Crop the image (cat's eye) by array slicing.

5.2. Increase the size of the "cat's eye" image to 400 x 400. Please discuss the difference between "rescale" and "resize".

### Discussion 6: Photoshop - bilateral filter ###

6.1. Please capture random faces from "https://generated.photos/face-generator/new"

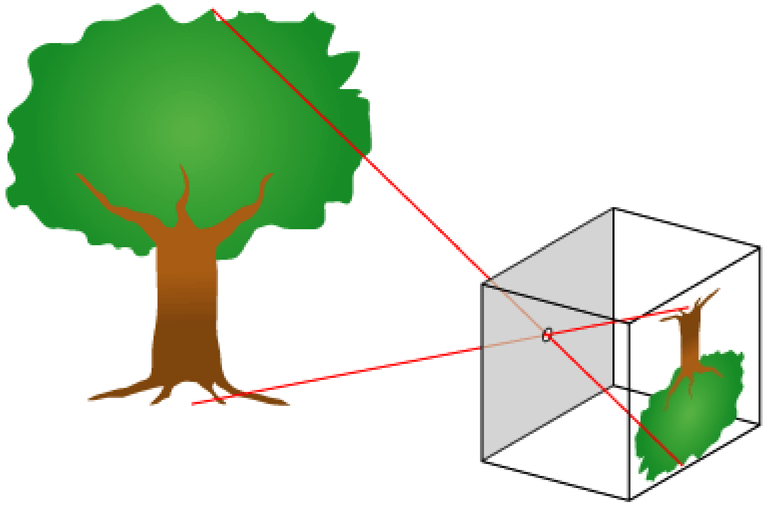
6.2. Please apply a "bilateral filter" on the image.

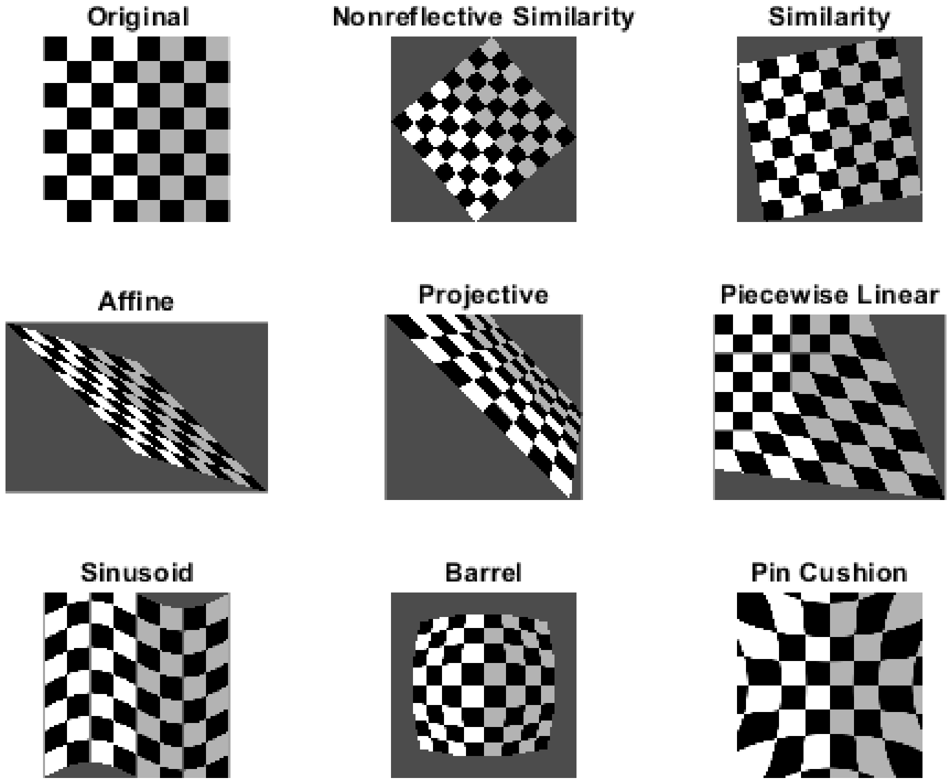
6.3. Please analyze the effect of the bilateral filter by changing the input parameters.

6.4. What are the pros and cons of filtering?

Next Contents

- Basic camera theory
  - Pinhole camera model
- Basic image transformation




Original
Nonreflective Similarity
Similarity
Affine
Projective
Piecewise Linear
Sinusoid
Barrel
Pin Cushion