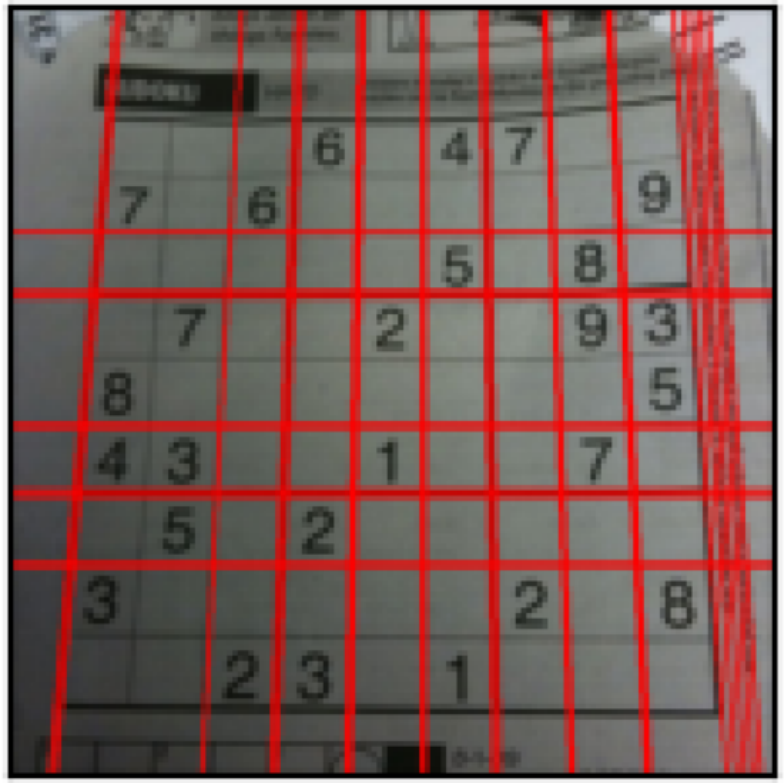
Template Matching

김수환

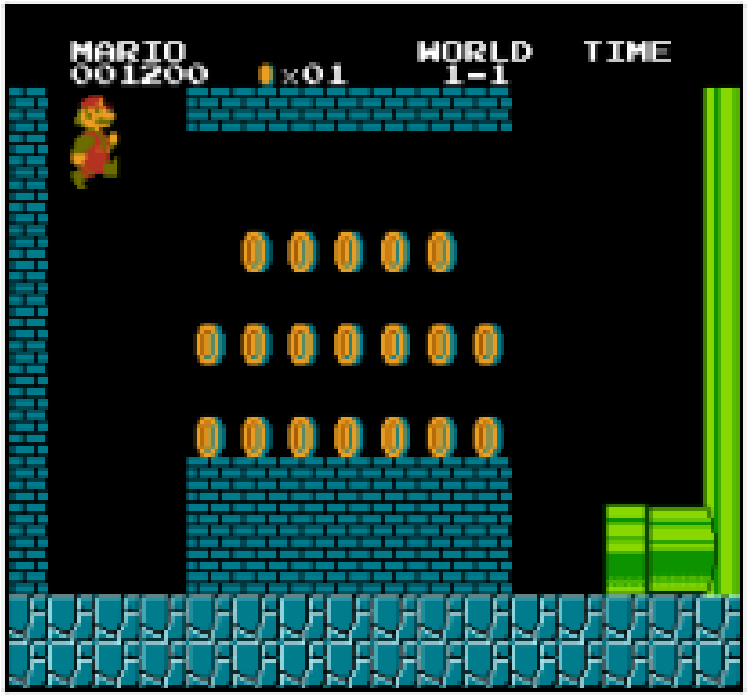
https://www.soohwan.kim

Line/Circle Detection

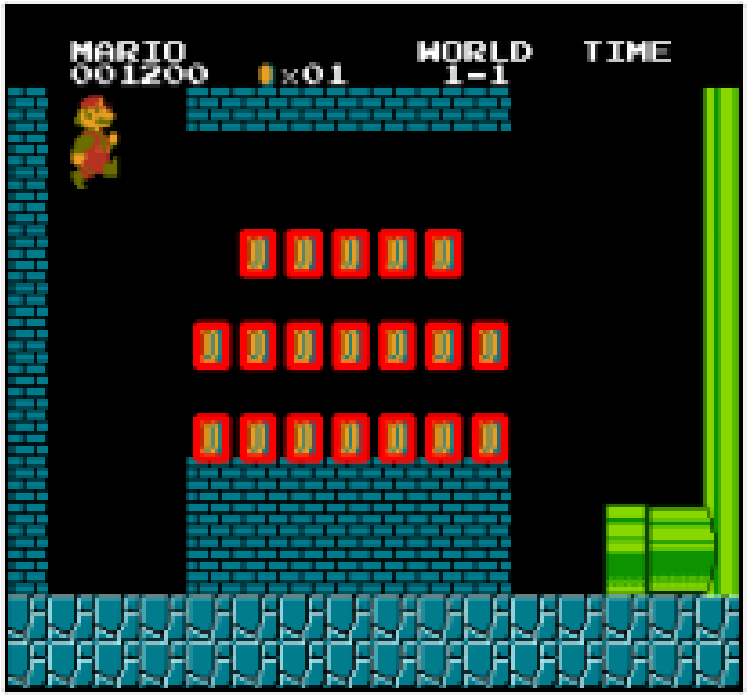


Object Detection

Original



Template Matching




학습목표

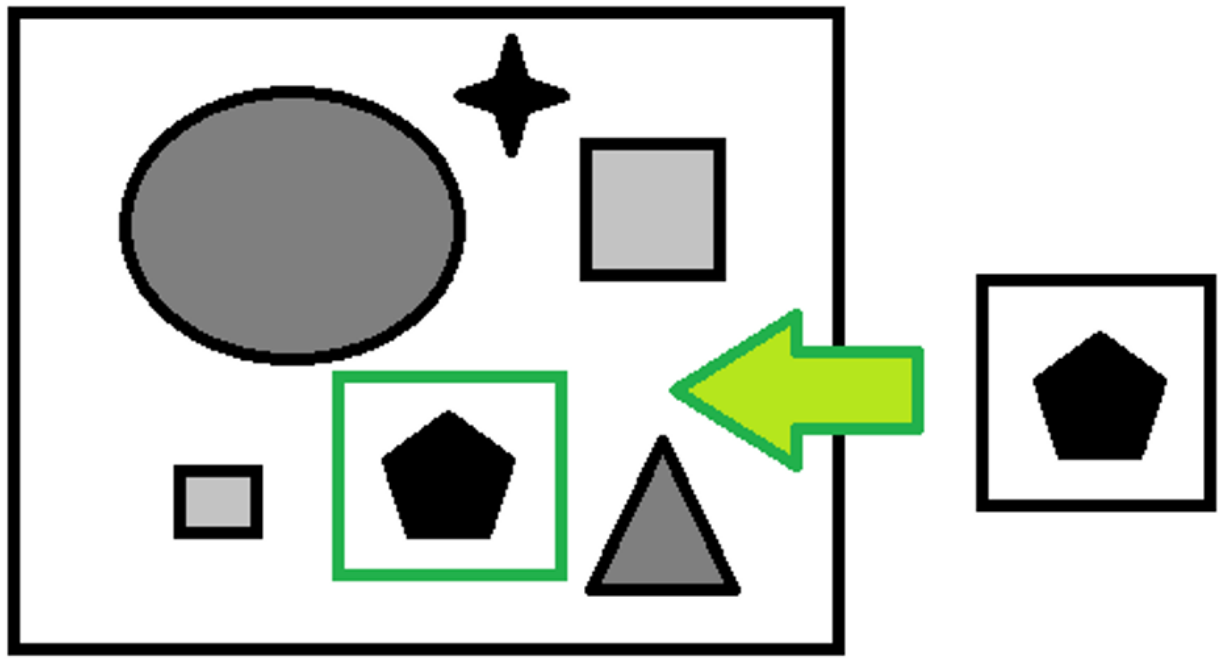
학습목표

1. Image Processing
    1. Image Thresholding
    2. Image Blending
    3. Image Filtering
    4. Morphological Transformations
    5. Image Gradients
    6. Hough Transforms
    7. Template matching

Template Matching

Template Matching

- 목적
  - Image에서 특정한 이미지/물체(template image)를 찾고 싶을 때



https://www.opto-e.com/basics/template-matching

Template Matching

- 방법
  1. Template Image를 Reference Image 위에 sliding 시킨다.
  2. Template Image와 Reference Image가 겹치는 부분의 pixel 값을 비교한다.
  3. 가장 비슷한 곳이 바로 해당 물체가 있는 곳!

Template

7	7	5
4	3	2
3	8	2

Reference

1	1	1	1	1	1
1	5	7	7	5	1
1	2	4	3	2	1
1	2	3	8	2	1
1	2	2	2	1	1
1	1	1	1	1	1

https://www.researchgate.net/figure/Template-and-Reference-Matrix_fig1_301443589

Matching Criteria (Image Difference)

1. SSD (Sum of Squared Differences)

$$SSD = \sum_{x',y'} \left(T(x',y') - I(x+x', y+y')\right)^2 = ||\mathbf{t} - \mathbf{r}||^2$$

2. SAD (Sum of Absolute Differences)

$$SAD = \sum_{x',y'} \left|T(x',y') - I(x+x', y+y')\right| = ||\mathbf{t} - \mathbf{r}||_1$$

3. NSSD (Normalized SSD)

$$NSSD = \frac{\sum_{x',y'} \left(T(x',y') - I(x+x', y+y')\right)^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x', y+y')^2}} = \frac{||\mathbf{t} - \mathbf{r}||^2}{\|\mathbf{t}\|\|\mathbf{r}\|}$$

Matching Criteria (Image Difference)

1. ZSSD (Zero Mean Sum of Squared Differences)

$$ZSSD = \sum_{x',y'} \left(\left(T(x',y') - \bar{T} \right) - \left(I(x+x', y+y') - \bar{I}\,(x,y) \right) \right)^2$$

2. ZSAD (Zero Mean Sum of Absolute Differences)

$$ZSAD = \sum_{x',y'} \left| \left(T(x',y') - \bar{T} \right) - \left(I(x+x', y+y') - \bar{I}\,(x,y) \right) \right|$$

3. NZSSD (Normalized Zero Mean Sum of Squared Differences)

$$NZSSD = \frac{\sum_{x',y'} \left(\left(T(x',y') - \bar{T} \right) - \left(I(x+x', y+y') - \bar{I}\,(x,y) \right) \right)^2}{\sqrt{\sum_{x',y'} \left(T(x',y') - \bar{T} \right)^2 \cdot \sum_{x',y'} \left(I(x+x', y+y') - \bar{I}\,(x,y) \right)^2}}$$

Matching Criteria (Image Difference)

1. SCC, (Simple Cross-Correlation)

$$SCC = \sum_{x',y'} T(x',y') I(x+x', y+y') = \mathbf{t} \cdot \mathbf{r}$$

2. NCC (Normalized Cross-Correlation)

$$NCC = \frac{\sum_{x',y'} T(x',y') I(x+x', y+y')}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x', y+y')^2}} = \frac{\mathbf{t} \cdot \mathbf{r}}{\|\mathbf{t}\| \|\mathbf{r}\|} = \cos\theta$$

Matching Criteria (Image Difference)

1. ZCC (Zero Mean Cross-Correlation)

$$ZCC = \sum_{x',y'} \left(T(x',y') - \bar{T}\right) \left(I(x+x',y+y') - \bar{I}\,(x,y)\right)$$

2. ZNCC (Zero Mean Normalized Cross-Correlation)

$$ZNCC = \frac{\sum_{x',y'} \left(T(x',y') - \bar{T}\right) \left(I(x+x',y+y') - \bar{I}\,(x,y)\right)}{\sqrt{\sum_{x',y'} \left(T(x',y') - \bar{T}\right)^2 \cdot \sum_{x',y'} \left(I(x+x',y+y') - \bar{I}\,(x,y)\right)^2}}$$

Cross-Correlation

- Definition: a weighted average of a function $f(\tau)$ at the moment t where the weighting is given by $g(\tau)$ simply shifted by amount t

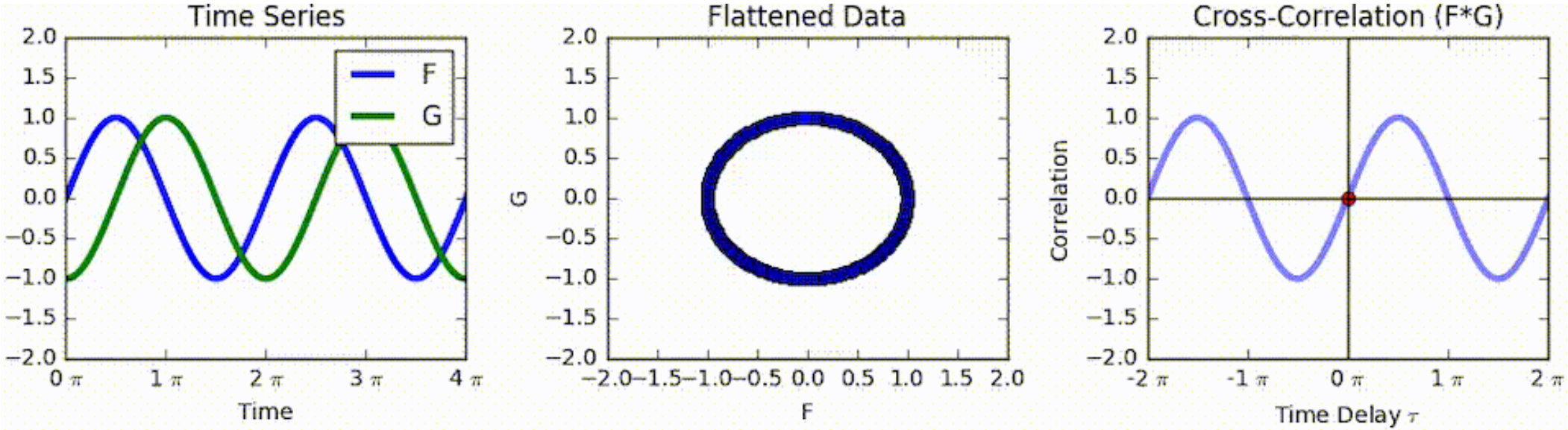
$$f(t) \star g(t) = (f \star g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t+\tau) d\tau$$

- Non-Commutative

$$g(t) \star f(t) = \int_{-\infty}^{\infty} g(\tau) f(t+\tau) d\tau = \int_{-\infty}^{\infty} -g(s-t) f(s) ds \neq f(t) \star g(t)$$

https://en.wikipedia.org/wiki/Convolution

Cross-Correlation



https://en.wikipedia.org/wiki/Cross-correlation

Convolution vs. Cross-Correlation

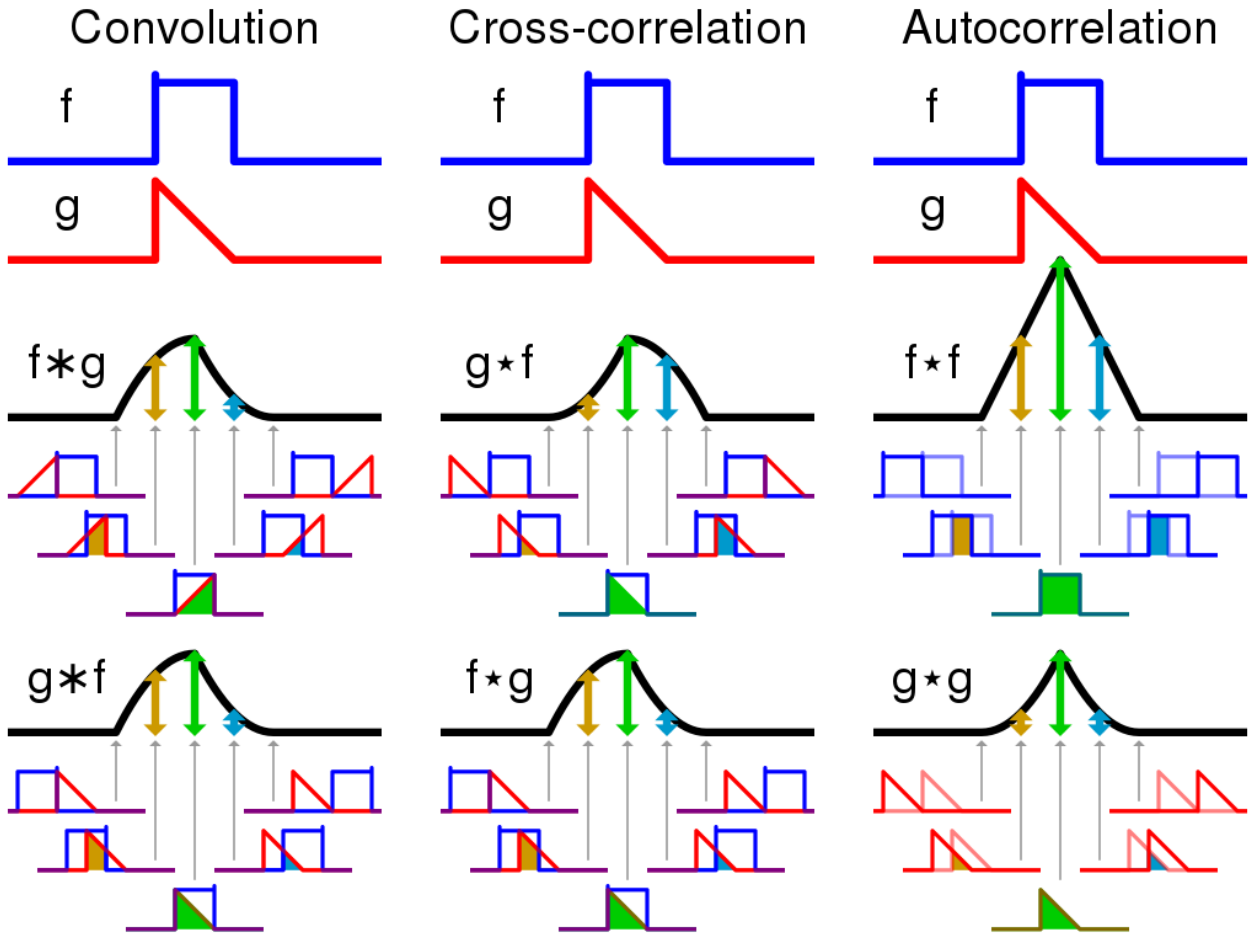
- Convolution

$$f(t) * g(t) = (f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau, \quad f(t) * g(t) = g(t) * f(t)$$

- Cross Correlation

$$f(t) \star g(t) = (f \star g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t + \tau) d\tau \quad f(t) \star g(t) \neq g(t) \star f(t)$$

Convolution vs. Cross-Correlation



https://en.wikipedia.org/wiki/Cross-correlation

`cv.matchTemplate()`[1]

```
result = cv.matchTemplate(image, templ, method[, result[, mask]])
```

- Image와 template이 겹치는 부분을 비교한다.
  - `image`: Image where the search is running. It must be 8-bit or 32-bit floating-point.
  - `templ`: Searched template. It must be not greater than the source image and have the same data type.
  - `result`: Map of comparison results. It must be single-channel 32-bit floating-point. If image is W×H and templ is w×h , then result is (W−w+1)×(H−h+1)
  - `method`: Parameter specifying the comparison method.
  - `mask`: Optional mask. It must have the same size as templ.

`method`[2]	Criteria	`method`[2]	Criteria	`method`[2]	Criteria
`cv.TM_SQDIFF`	SSD	`cv.TM_CCORR`	SCC	`cv.TM_CCOEFF`	ZCC
`cv.TM_SQDIFF_NORMED`	NSSD	`cv.TM_CCORR_NORMED`	NCC	`cv.TM_CCOEFF_NORMED`	ZNCC

1. https://docs.opencv.org/4.5.0/df/dfb/group__imgproc__object.html#ga586ebfb0a7fb604b35a23d85391329be
2. https://docs.opencv.org/4.5.0/df/dfb/group__imgproc__object.html#ga3a7850640f1fe1f58fe91a2d7583695d

Template Matching: Code 1

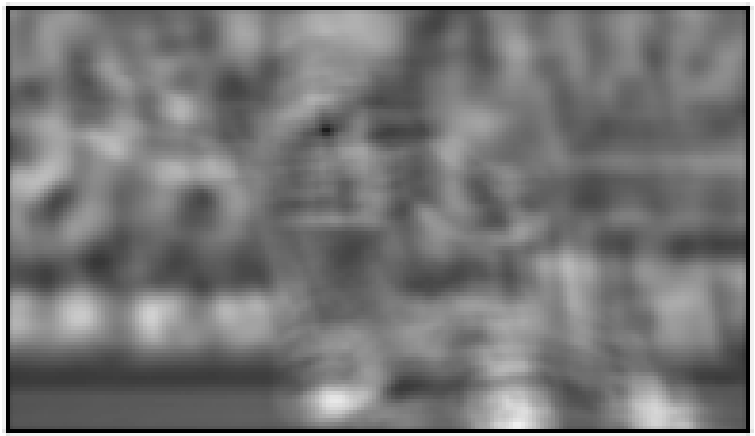
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Load a reference image as grayscale
img = cv2.imread('messi5.jpg', 0)
img2 = img.copy()

# Load a template image as grayscale
template = cv2.imread('messi_face.jpg', 0)
w, h = template.shape[::-1]

# All the 6 methods for comparison in a list
methods = ['cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED',
           'cv2.TM_CCORR',  'cv2.TM_CCORR_NORMED',
           'cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED']

for meth in methods:
    img = img2.copy()
    method = eval(meth)
```

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_template_matching/py_template_matching.html

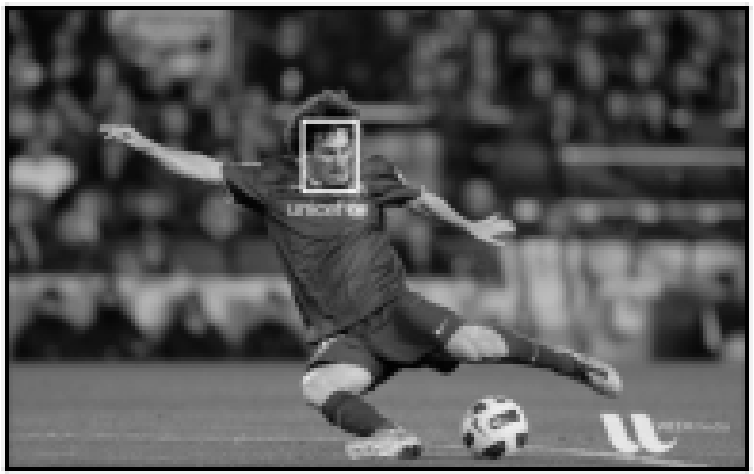
Template Matching: Result 1

cv2.TM_SQDIFF

Matching Result



Detected Point



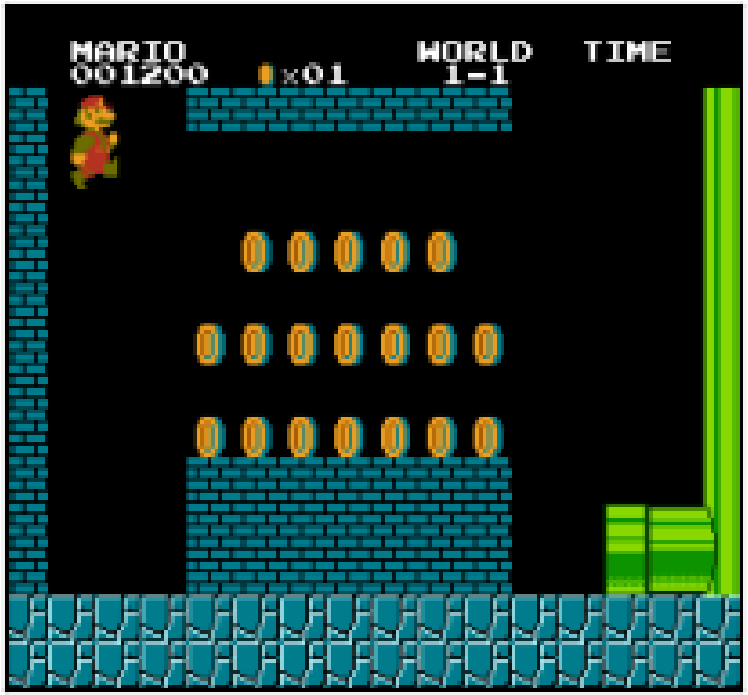
Template Matching: Code 2

```
# Load a reference image
img_rgb = cv2.imread('mario.png')
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)

# Load a template image as grayscale
template = cv2.imread('mario_coin.png', 0)
w, h = template.shape[::-1]

# Apply template matching
res = cv2.matchTemplate(img_gray, template, cv2.TM_CCOEFF_NORMED)

# Thresholding
threshold = 0.8
loc = np.where(res >= threshold)

# Draw a bounding box
img_res = img_rgb.copy()
for pt in zip(*loc[::-1]):
    cv2.rectangle(img_res, pt, (pt[0] + w, pt[1] + h), (0,0,255), 2)

# Display results
titles = ['Original', 'Template Matching']
```

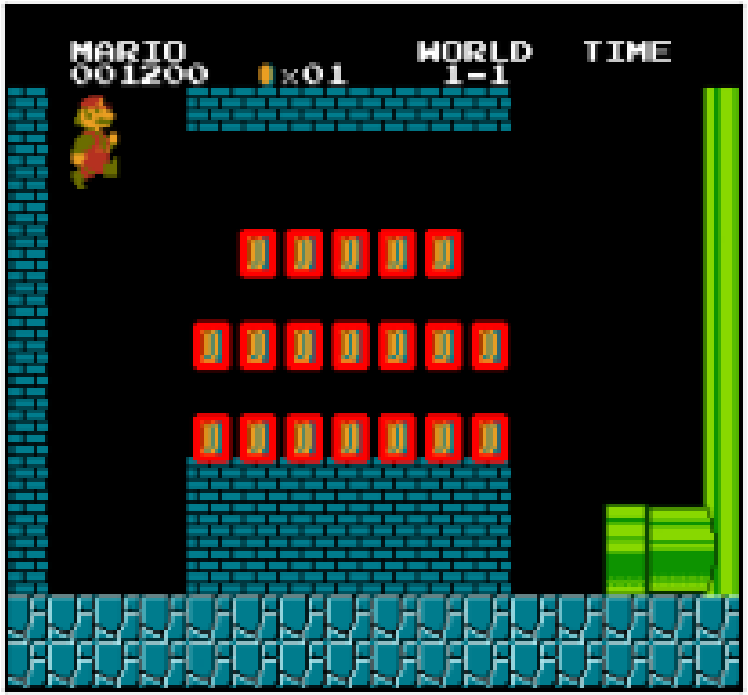
https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_template_matching/py_template_matching.html

Template Matching: Result 2

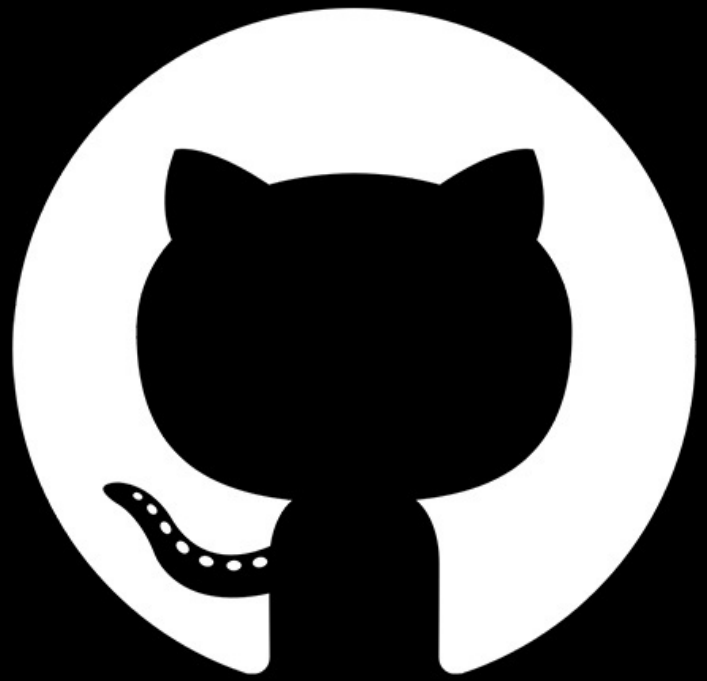
Original



Template Matching



Push Code to GitHub



Summary

- Template Matching
  - 목적: Image에서 Template Image에 해당하는 물체 인식
  - 방법
    1. Sliding Window
    2. Matching Criteria (SSD, SAD, NCC)
  - 단점?
    - Translation!
    - Rotation?
    - Scale?

References

References

- OpenCV Python Tutorials
  - Core Operations
    - Basic Operations on Images
    - Arithmetic Operations on Images
  - Image Processing
    - Image Thresholding
    - Smoothing Images
    - Morphological Transformations
    - Image Gradients
    - Hough Line Transform
    - Hough Circle Transform
    - Template Matching