



Number Recognition

패턴인식개론 Module 1
2017.04.19

마지막이조
제작 : 유태우(ldaytw@gmail.com)
유태우 기고운 김종현 김현정 최철순



Index

목차 |

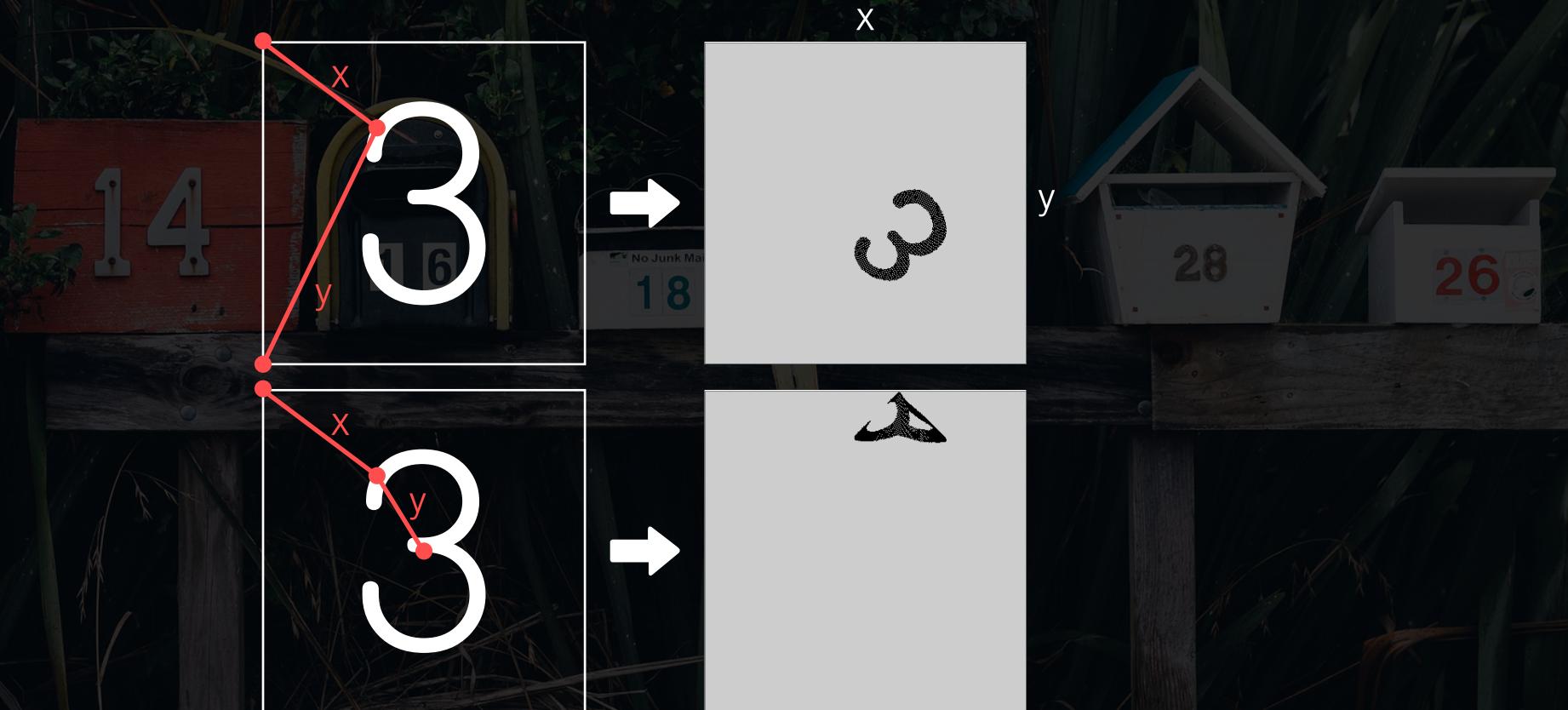
- 01 알고있는 사실
- 02 논의한 방법
- 03 찾아낸 특징
- 04 사용한 특징
- 05 구현 전제조건
- 06 특징별 구현 방법
- 07 숫자별 검출 특징 종합
- 08 결과물

● 폰트가 3가지로 정해져있다.

● 이미지가 정형화되어 있다.

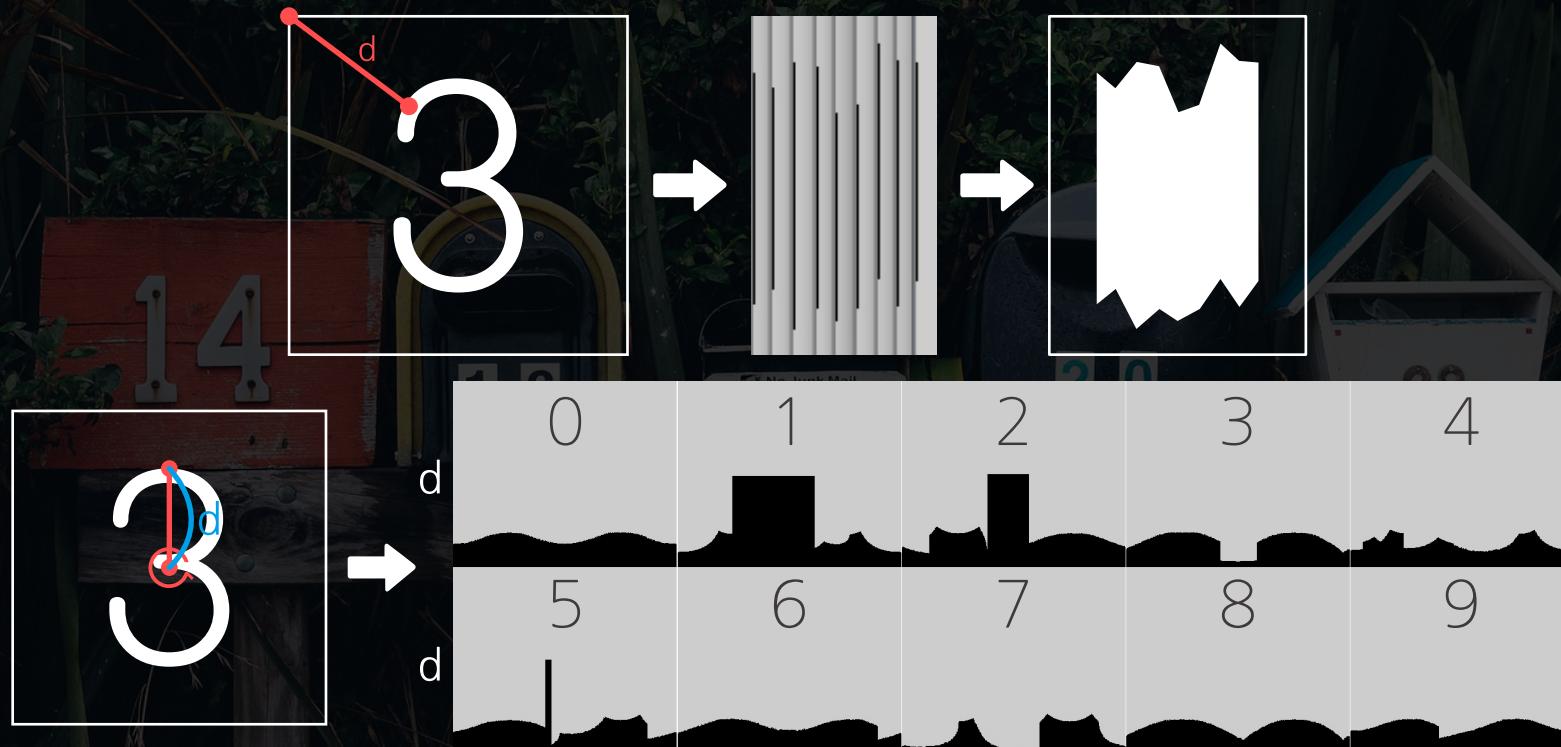
■ 특정 지점과의 거리를 이용(1)

- 중심과 각 픽셀간의 거리의 합
- 특정 두 지점과 각 픽셀간의 거리의 분포를 이용해 tensorflow를 통한 딥러닝



■ 특정 지점과의 거리를 이용(2)

- $(0,0)$ 과 특정 지점과의 그래프
- 360도 회전하며 중심과 가장자리와의 거리 그래프



■ 숫자마다 다른 특징을 이용

상단에 수평선이 있는지

상하 대칭

직선이 몇개 있는지

수평선이 있는지

좌우 대칭

가장 긴변의 각도

원이 존재하는지

수직선의 개수

수직선이 있는지

중심선과 만나는 부분이 몇개인지

수평선의 개수

하단에 수평선이 있는지

수직선의 위치

곡선이 있는지

오른쪽 위가 뚫여있는지

막힌곳의 위치가 어디인지

막혀있는 곳이 있는지

상단에 수평선이 있는지

상하 대칭

수평선이 있는지

직선이 몇개 있는지

수직선의 개수

원이 존재하는지

좌우 대칭

가장 긴변의 각도

수직선이 있는지

중심선과 만나는 부분이 몇개인지

수평선의 개수

1 6

18

2 0

28

26

하단에 수평선이 있는지

수직선의 위치

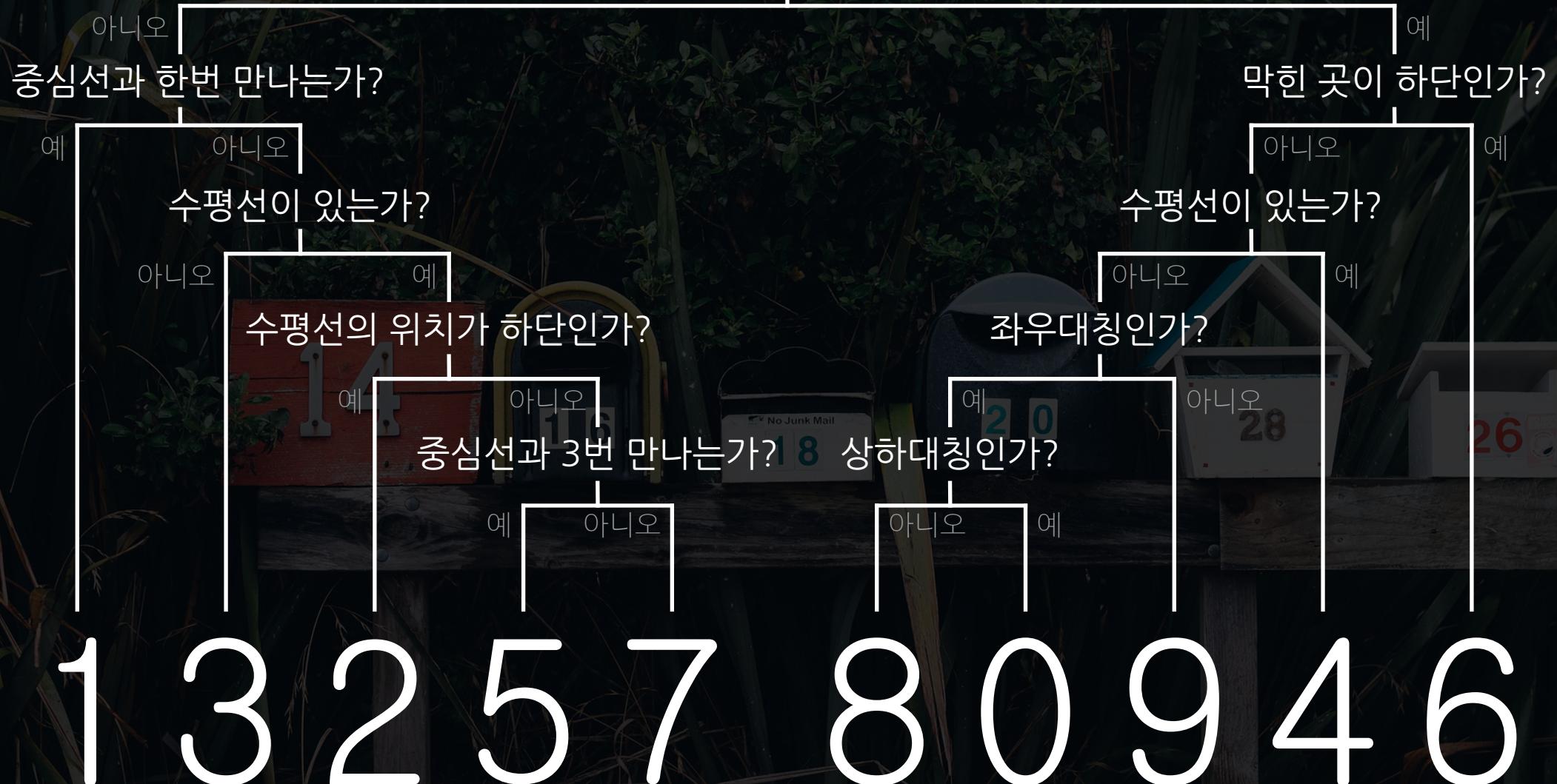
곡선이 있는지

오른쪽 위가 뚫여있는지

막힌곳의 위치가 어디인지

막혀있는 곳이 있는지

막힌 곳이 있는가?

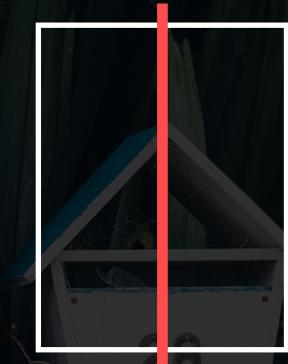




OpenCV



2진화된 이미지

실제영역
및 중심선







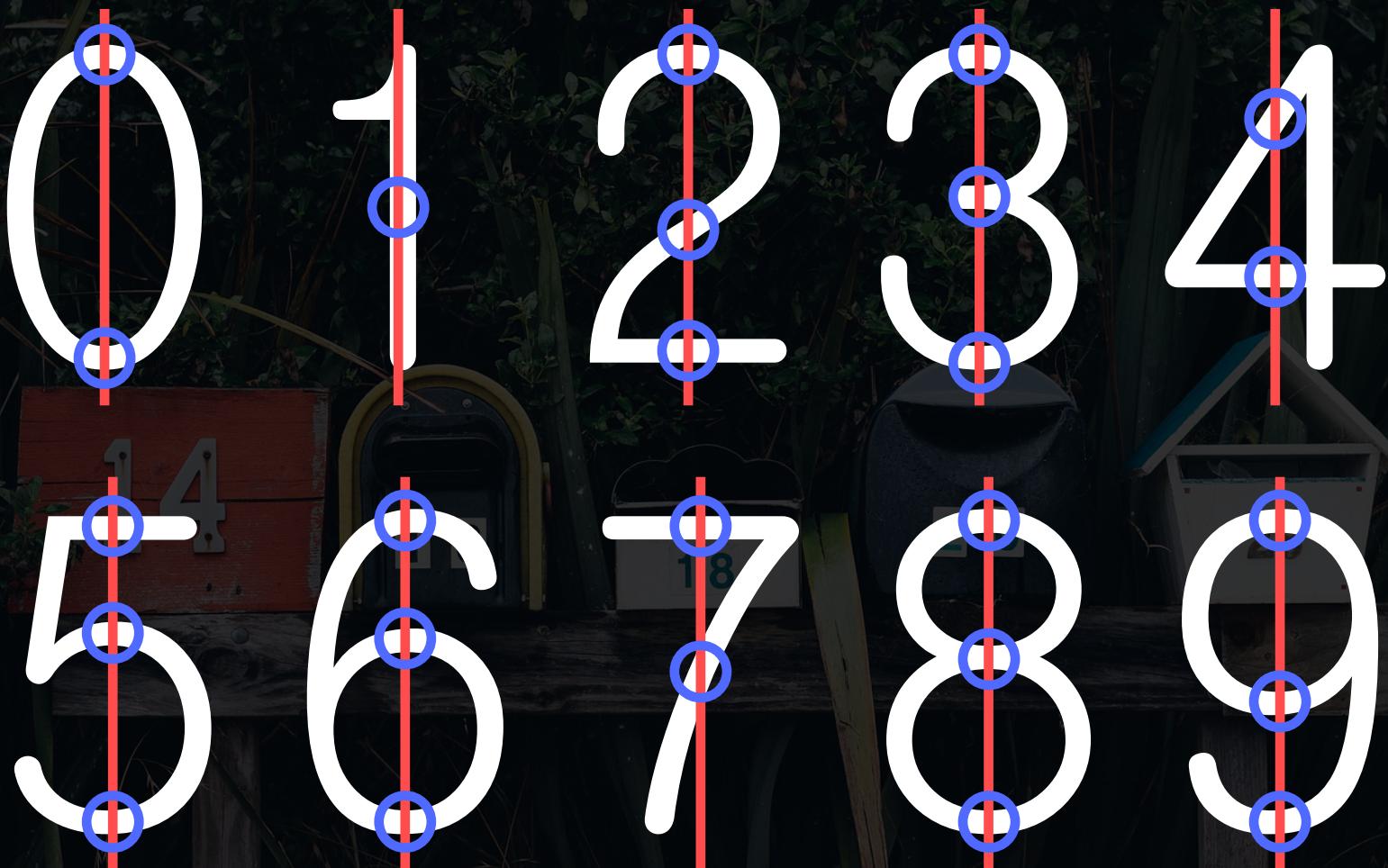
$$\text{threshold} = \text{MAX}(30, L * 0.7)$$

(pixel)

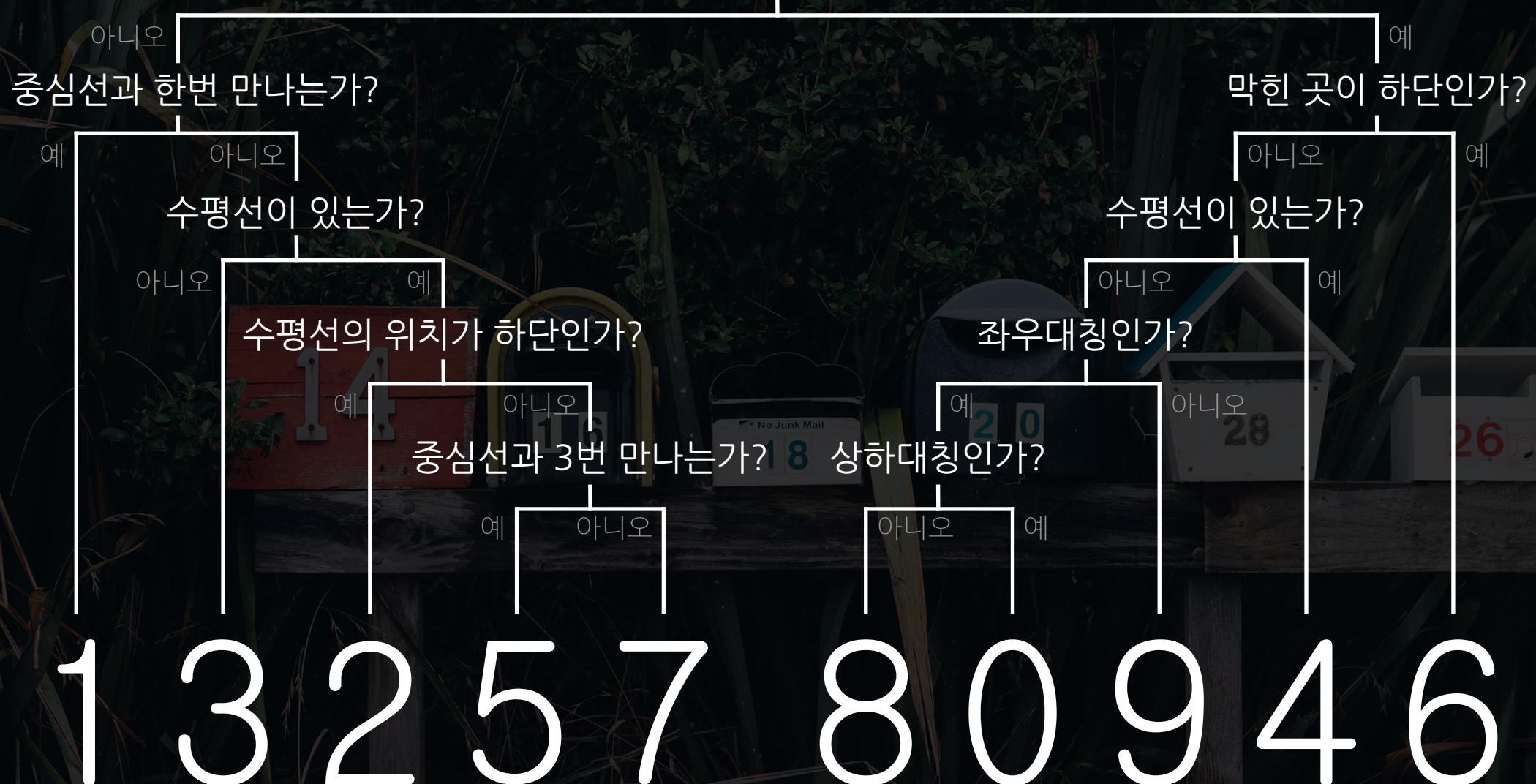


013689


$$\frac{\text{일치한 픽셀 수}}{\text{비교한 픽셀 수}} > 90\%$$



막힌 곳이 있는가?



굴림

```
=====
```

```
input : 굴림₩₩0.png
```

```
예상 숫자 : 0
```

```
=====
```

```
input : 굴림₩₩1.png
```

```
예상 숫자 : 1
```

```
=====
```

```
input : 굴림₩₩2.png
```

```
예상 숫자 : 2
```

```
=====
```

```
input : 굴림₩₩3.png
```

```
예상 숫자 : 3
```

```
=====
```

```
input : 굴림₩₩4.png
```

```
예상 숫자 : 4
```

```
=====
```

```
input : 굴림₩₩5.png
```

```
예상 숫자 : 5
```

```
=====
```

```
input : 굴림₩₩6.png
```

```
예상 숫자 : 6
```

```
=====
```

```
input : 굴림₩₩7.png
```

```
예상 숫자 : 7
```

```
=====
```

```
input : 굴림₩₩8.png
```

```
예상 숫자 : 8
```

```
=====
```

```
input : 굴림₩₩9.png
```

```
예상 숫자 : 9
```

견고딕

```
=====
```

```
input : 견고딕₩₩0.png
```

```
예상 숫자 : 0
```

```
=====
```

```
input : 견고딕₩₩1.png
```

```
예상 숫자 : 1
```

```
=====
```

```
input : 견고딕₩₩2.png
```

```
예상 숫자 : 2
```

```
=====
```

```
input : 견고딕₩₩3.png
```

```
예상 숫자 : 3
```

```
=====
```

```
input : 견고딕₩₩4.png
```

```
예상 숫자 : 4
```

```
=====
```

```
input : 견고딕₩₩5.png
```

```
예상 숫자 : 5
```

```
=====
```

```
input : 견고딕₩₩6.png
```

```
예상 숫자 : 6
```

```
=====
```

```
input : 견고딕₩₩7.png
```

```
예상 숫자 : 7
```

```
=====
```

```
input : 견고딕₩₩8.png
```

```
예상 숫자 : 8
```

```
=====
```

```
input : 견고딕₩₩9.png
```

```
예상 숫자 : 9
```

한컴백제M

```
=====
```

```
input : 한컴백제₩₩0.png
```

```
예상 숫자 : 0
```

```
=====
```

```
input : 한컴백제₩₩1.png
```

```
예상 숫자 : 1
```

```
=====
```

```
input : 한컴백제₩₩2.png
```

```
예상 숫자 : 2
```

```
=====
```

```
input : 한컴백제₩₩3.png
```

```
예상 숫자 : 3
```

```
=====
```

```
input : 한컴백제₩₩4.png
```

```
예상 숫자 : 4
```

```
=====
```

```
input : 한컴백제₩₩5.png
```

```
예상 숫자 : 5
```

```
=====
```

```
input : 한컴백제₩₩6.png
```

```
예상 숫자 : 6
```

```
=====
```

```
input : 한컴백제₩₩7.png
```

```
예상 숫자 : 7
```

```
=====
```

```
input : 한컴백제₩₩8.png
```

```
예상 숫자 : 8
```

```
=====
```

```
input : 한컴백제₩₩9.png
```

```
예상 숫자 : 9
```

This repository contains 6 commits, 1 branch, 0 releases, and 1 contributor. The license is Apache-2.0.

NumberRecognition

미리 정해진 숫자를 인식하는 코드입니다.
숫자는 글꼴, 색상, 한글꺠개M 폰트 3가지 종류로 이루어져있고, 정형화된 크기와 위치에 숫자가 위치합니다.
이 프로그램은 위 3가지 폰트 및 정형화된 이미지에만 적용이 가능하므로 포함된 이미지가 아닌 다른 숫자가 적힌 이미지를 인식하고자 할 때에는 정확한 인식이 불가능할 수 있습니다.

Feature

- 숫자를 검출하기 위해 다음 8가지의 특징을 활용합니다.
- 좌우대칭
- 상대대칭
- 수정선이 상단에 위치하는지
- 수정선이 하단에 위치하는지
- 수정선이 존재하지 않는지
- 막혀있는 곳이 있는지
- 막혀있는 곳이 있다면 상단인지 하단인지
- 중심선이 유통되는 부분이 몇 개인지

Usage

이 프로그램을 사용하기 위해서는 opencv가 필요합니다.
<http://www.opencv.org/releases.html>
참고 : <http://webnautes.tistory.com/716>

```
#include<opencv2/opencv.hpp>
#include<opencv2/imgproc/imgproc.hpp>
#include<opencv2/highgui/highgui.hpp>

int main()
{
    IplImage *src = cvLoadImage(fileName);
}
```

Image

Github

<https://git.io/vSVAI>



Q & A

1 | 6

패턴인식개론 Module 1
2017.04.19

No Junk Mail
18

2 | 0

28

26