

# Digit recognition tool

Dastan Iyembergen

# Content

- Problem
- My solution
  - Preprocessing
  - Recognizing a single digit
- Demo

# Problem

The problem is to recognize digits from a digital image.

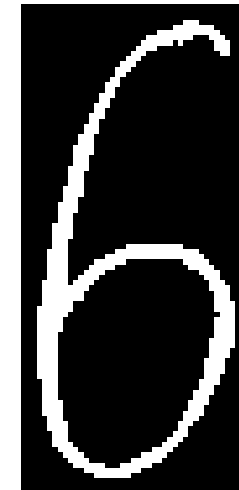
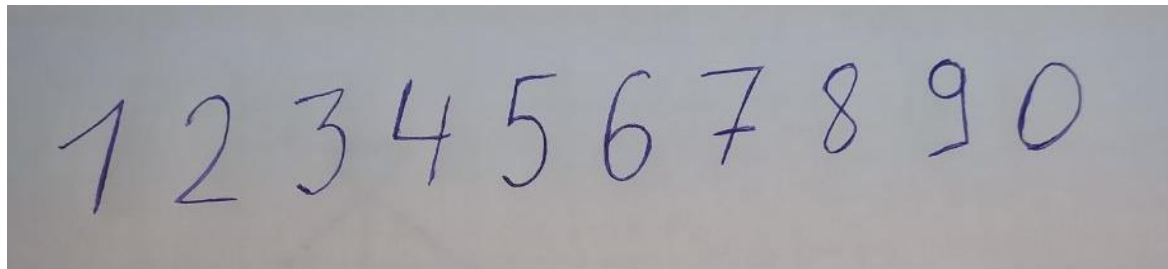
- Input:
  - Digital image of a list of digits on a white paper
- Output:
  - List of numbers (numerical)

# My solution

- Python 3
- CV2 (OpenCV): to read image
- matplotlib.pyplot (plt): to show, save images
- Math, Numpy, Queue
- My approach: 'Human Learning'. ~~Machine Learning~~

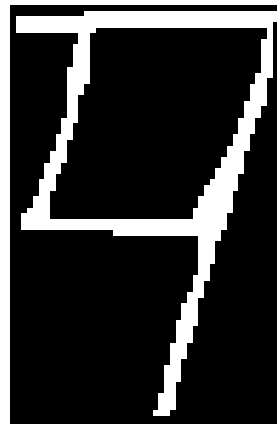
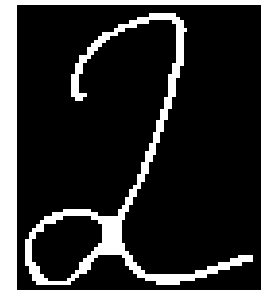
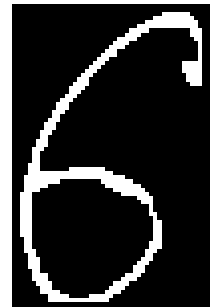
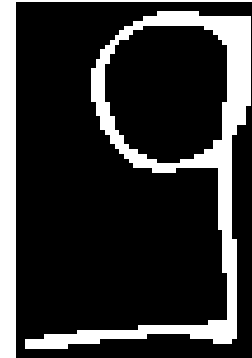
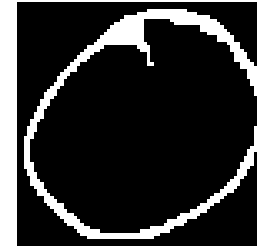
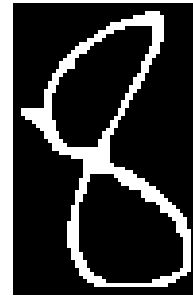
# Preprocessing

- Convert to Gray
- Get Negative
- Thresholding (dynamic)
- Extracting Component with BFS. ~~Region Filling (dilation)~~
- Resize



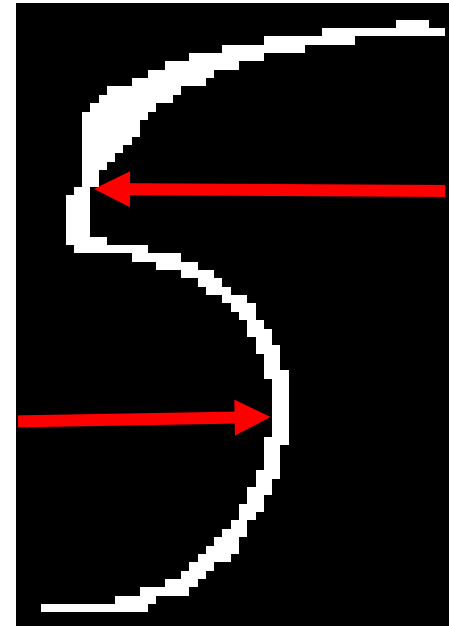
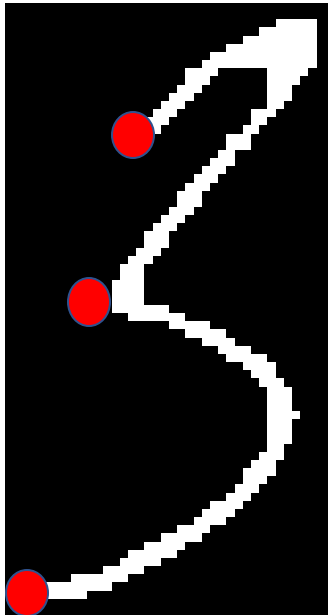
# Recognizing a single digit

- Use holes (extract with BFS):
  - 2 holes: 8
  - 1 big hole: 0
  - Hole at high: 9
  - Hole at low:
    - Only hole: 6
    - Hole with something near it: 2
  - Draw upper line, get 1 hole: 4



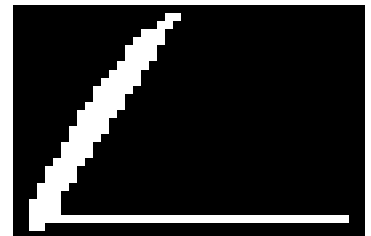
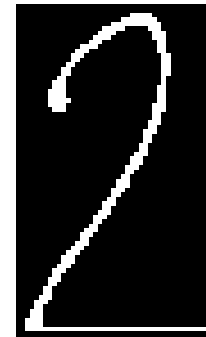
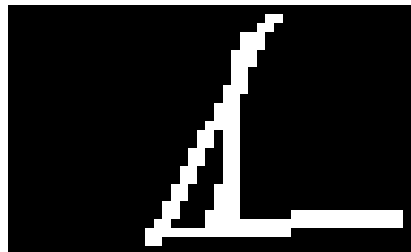
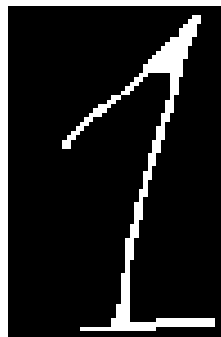
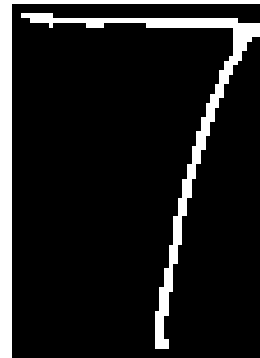
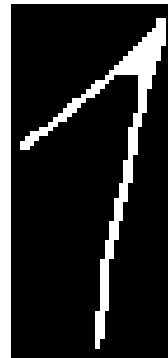
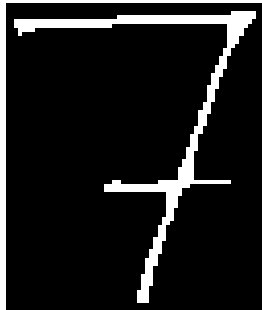
# Recognizing a single digit

- 3 vertices from left: 3
- Distance from right to upper + distance from left to lower  $\gg$  width: 5



# Recognizing a single digit

- - at the center: 7
- Thin lower part:
  - Perfectly thin: 1
  - Mostly thin: 7
  - Else: 1
- Cut lower part, draw line:
  - If hole: 1
  - Else: 2
- Default: 2

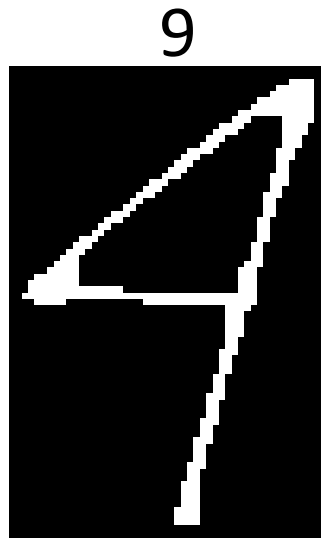




# Results

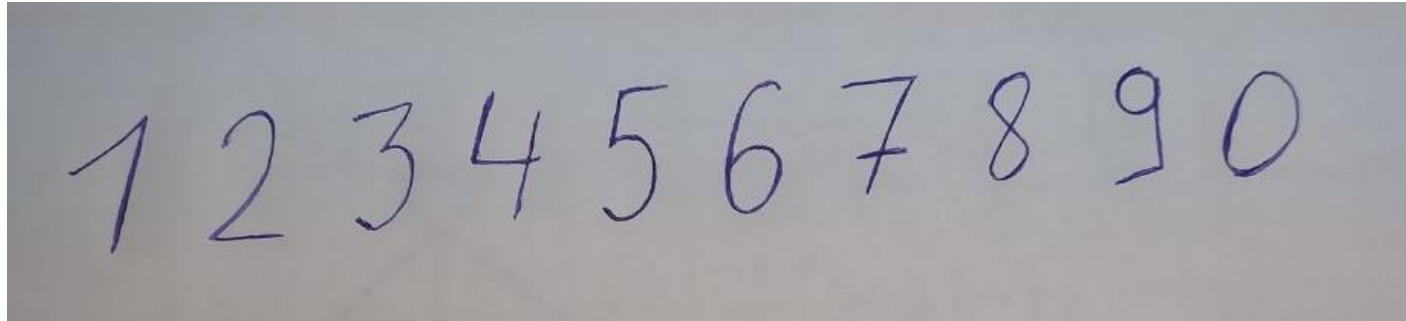
- Test dataset:  $4 \times 10 = 40$
- Correctly found: 37
- Mistakes:

- 



# Testing Multiple Digits

- Input:



- Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]

# Demo

Thank you for your attention