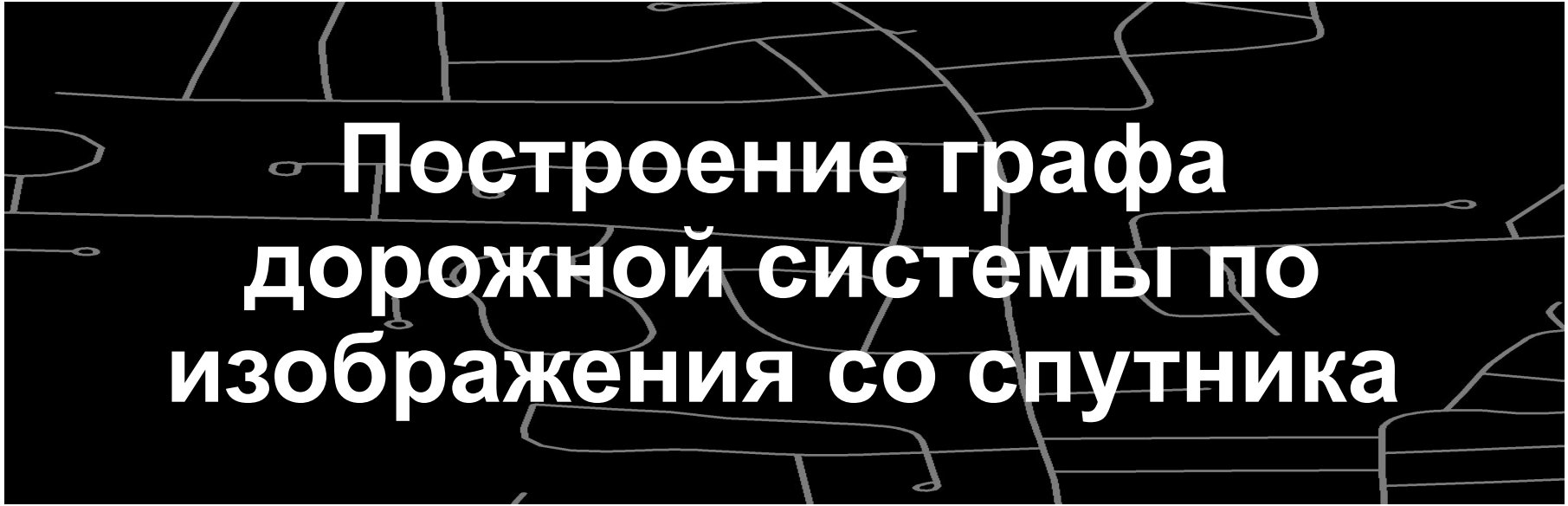


Построение графа дорожной системы по изображения со спутника

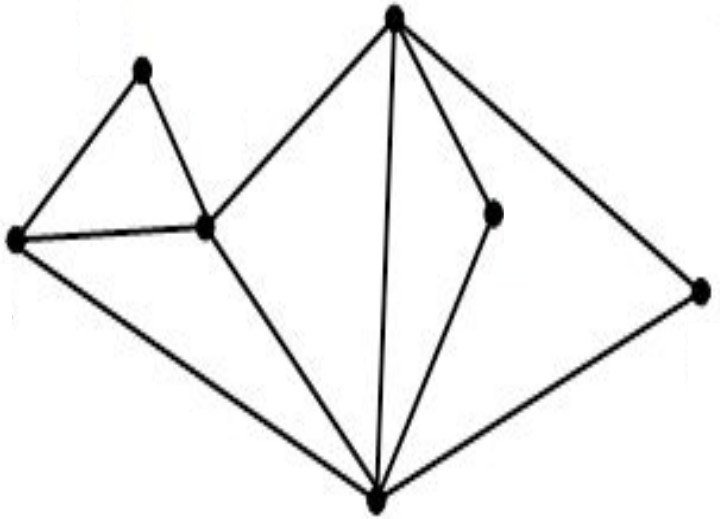
Кондренко Кирилл
21203

Шальнев Тимофей
21204

Новосибирский государственный университет

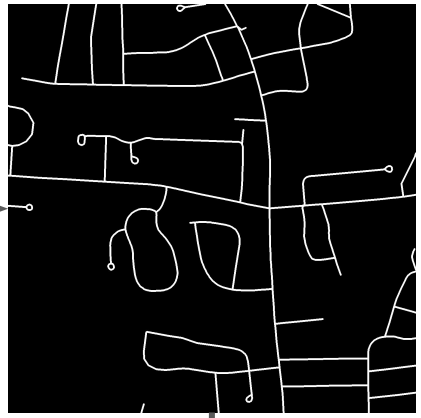
2024

Problem

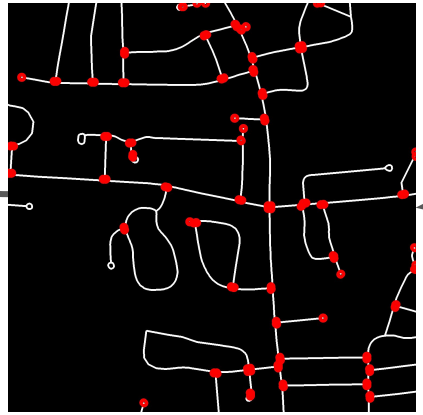


Pipeline

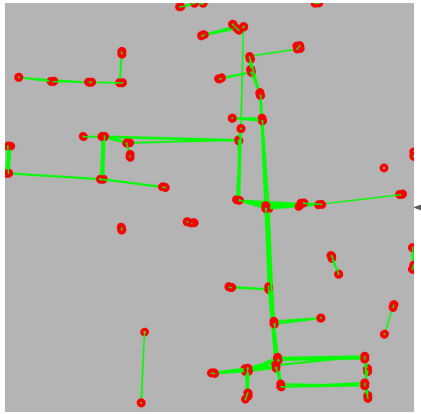
Binary Segmentation



Crossroads detection

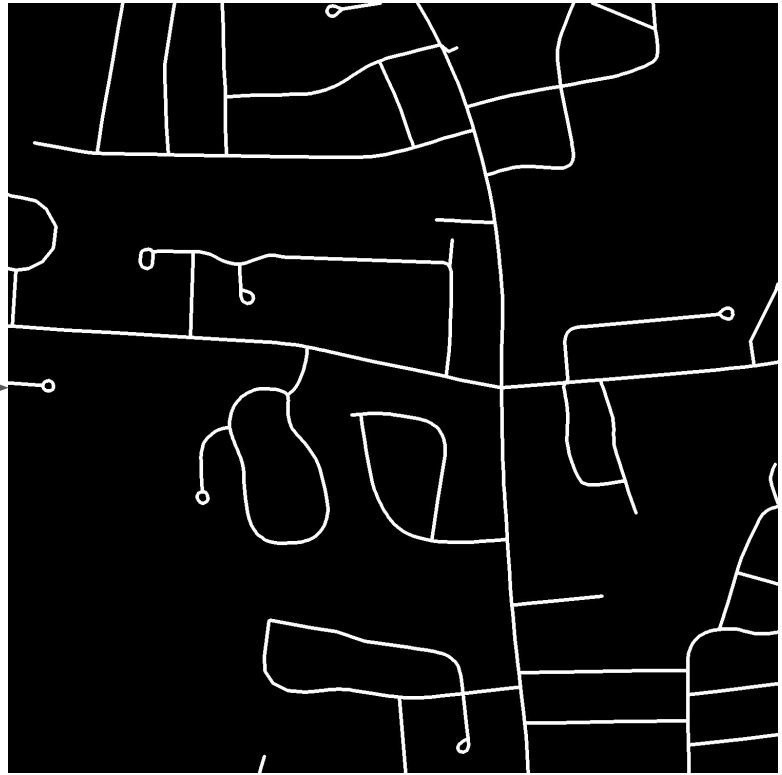


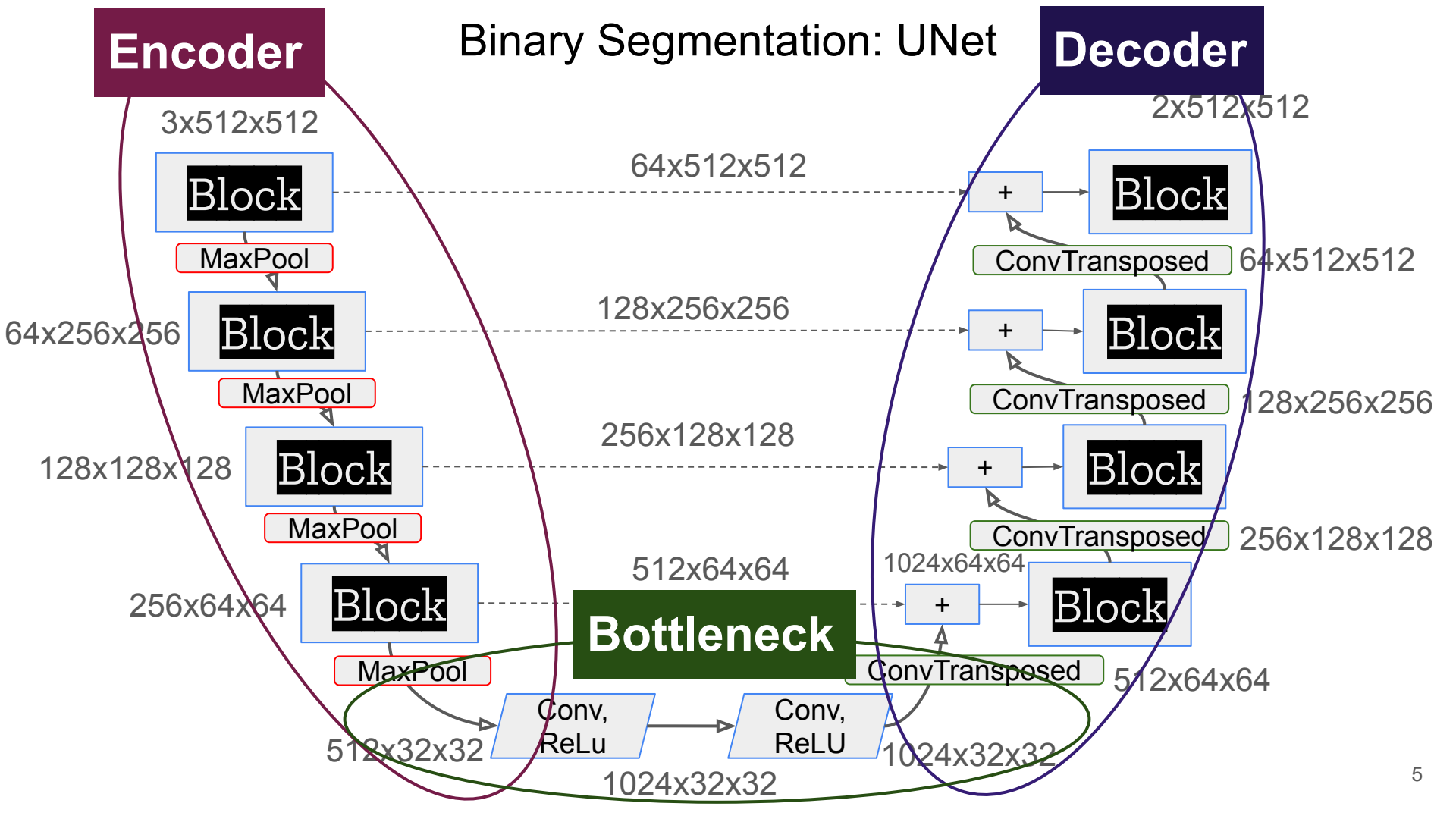
Making a graph

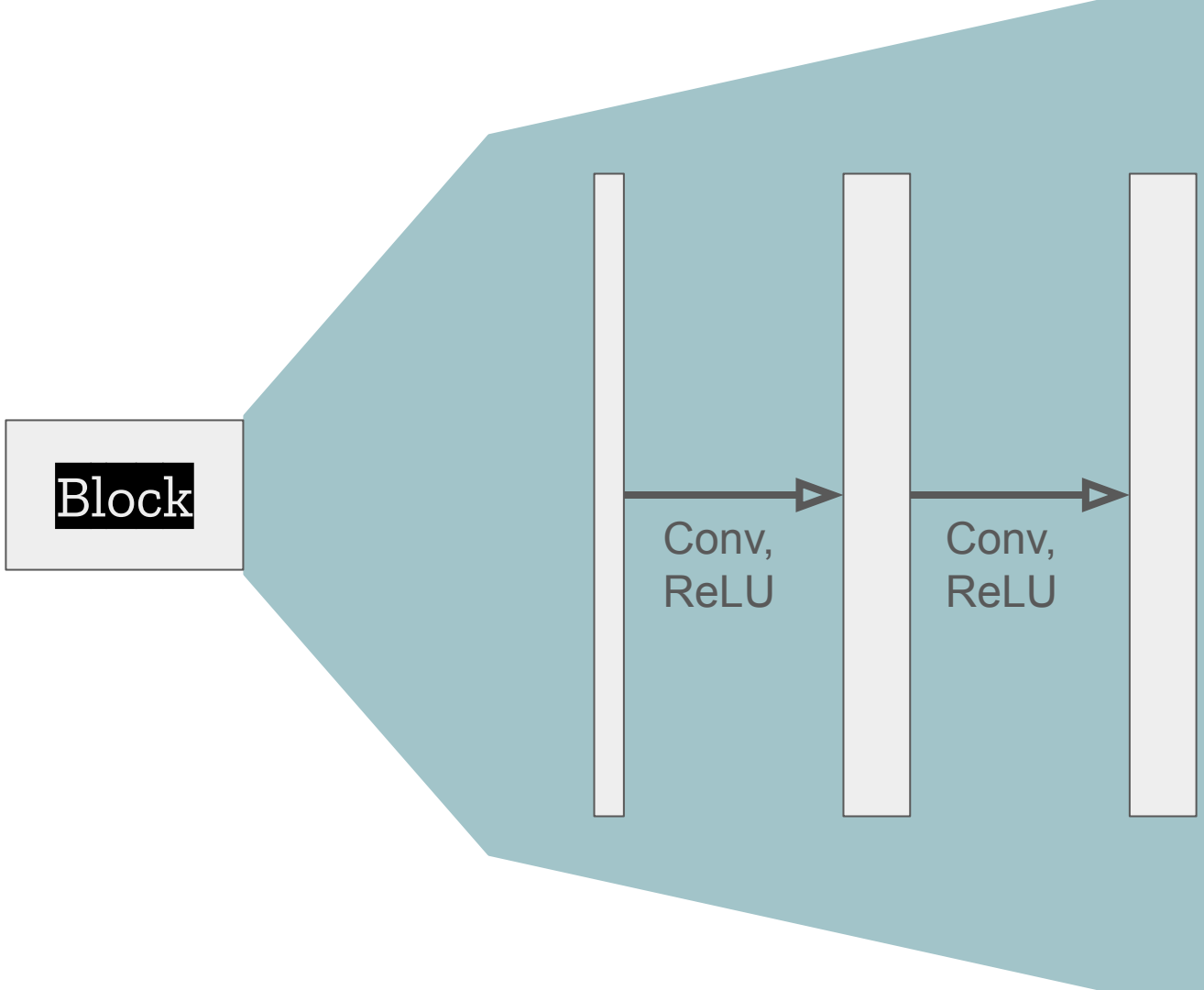


Massachusetts Roads Dataset

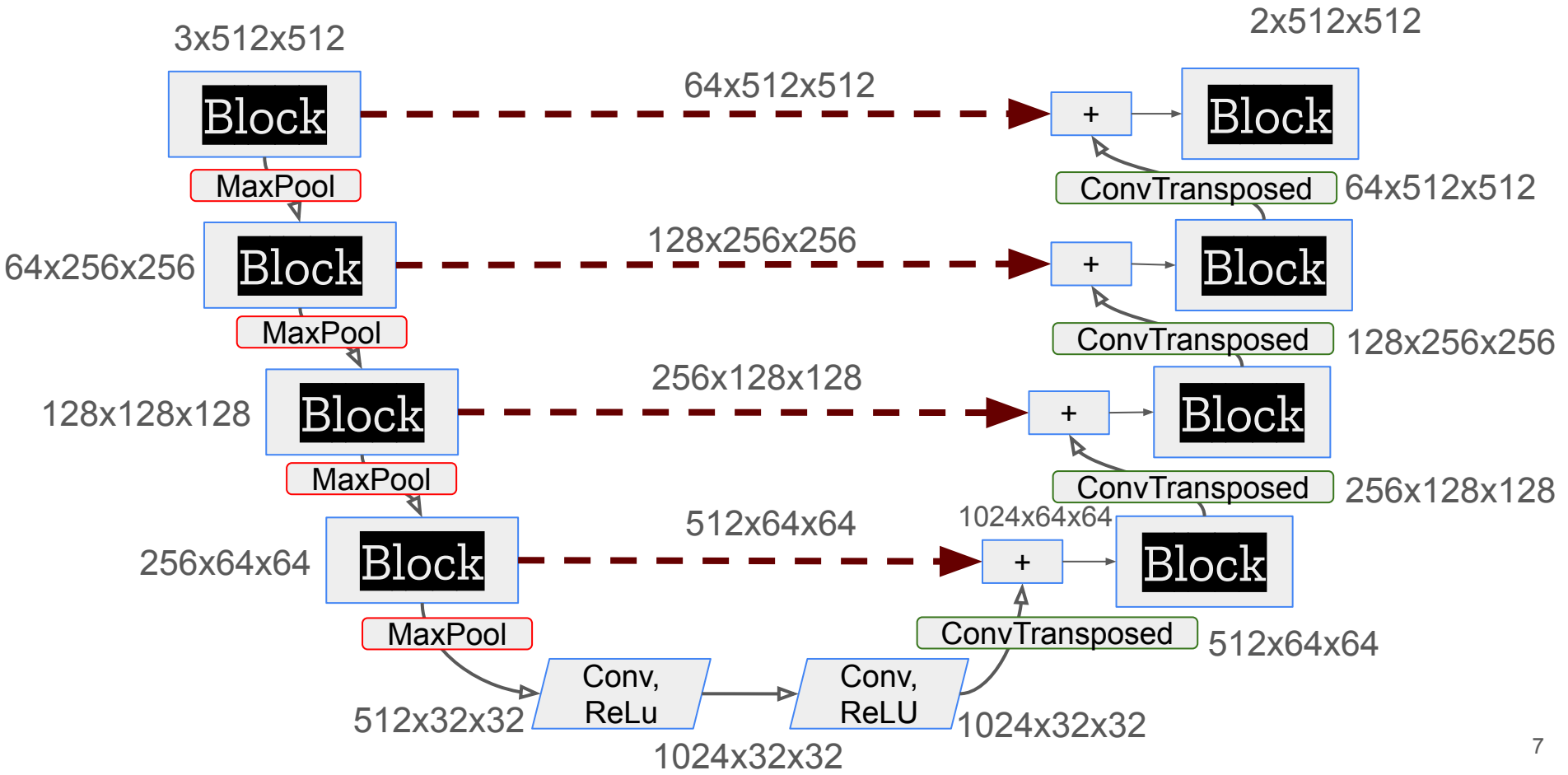
```
@phdthesis{MnihThesis,
    author = {Volodymyr Mnih},
    title = {Machine Learning for Aerial Image Labeling},
    school = {University of Toronto},
    year = {2013}
}
```



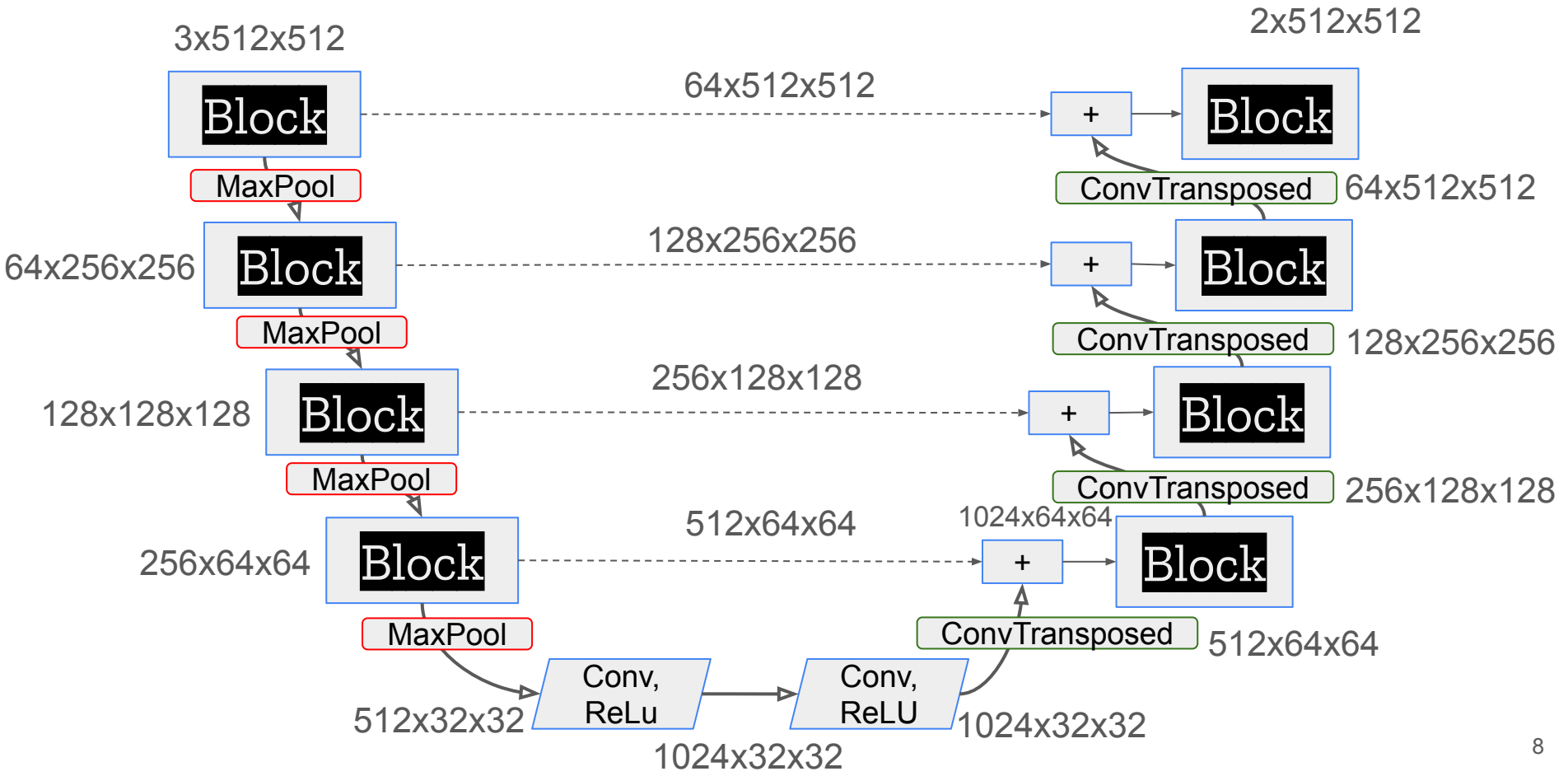

Encoder
Binary Segmentation: UNet
Decoder
3x512x512
Block
64x512x512
+
Block
MaxPool
ConvTransposed
64x512x512
64x256x256
Block
128x256x256
+
Block
MaxPool
ConvTransposed
128x256x256
128x128x128
Block
256x128x128
+
Block
MaxPool
ConvTransposed
256x128x128
256x64x64
Block
512x64x64
1024x64x64
+
Block
Bottleneck
MaxPool
ConvTransposed
512x64x64
Conv, ReLu
Conv, ReLU
512x32x32
1024x32x32
1024x32x32


Block
Conv,
ReLU
Conv,
ReLU

Binary Segmentation: UNet



Binary Segmentation: UNet



Original Image

Ground Truth Mask

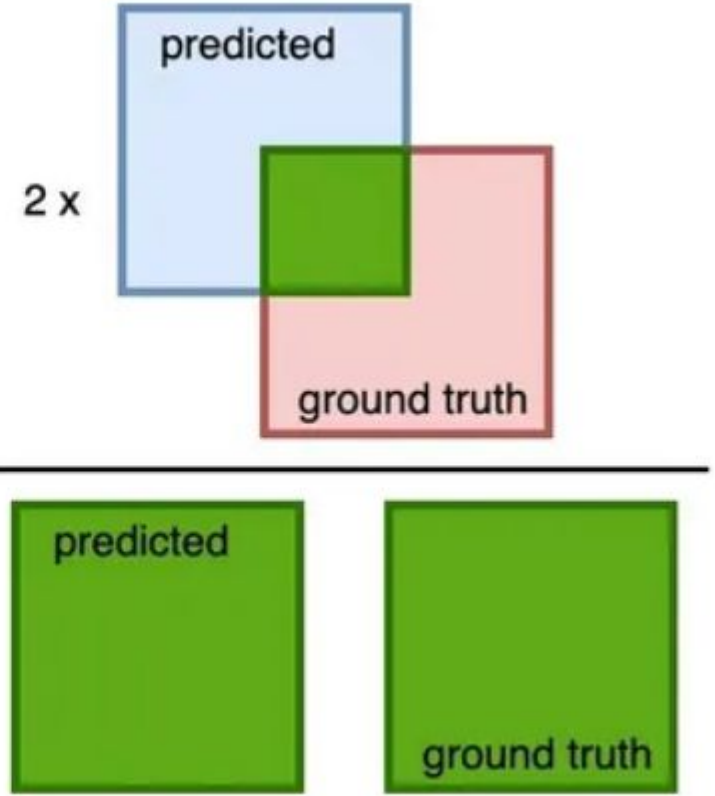
Predicted Mask

Original Image

Ground Truth Mask

Predicted Mask

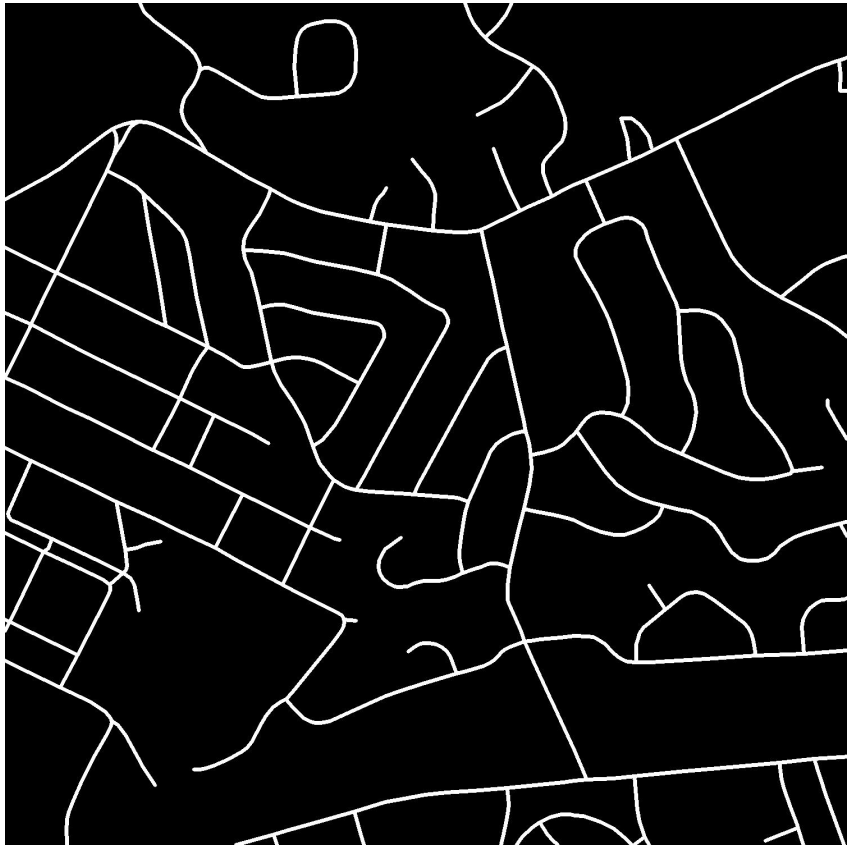
$$\text{Dice coefficient} = \frac{\text{2x area of overlapped (green)}}{\text{total area (green)}} = \frac{2 \times}{}$$

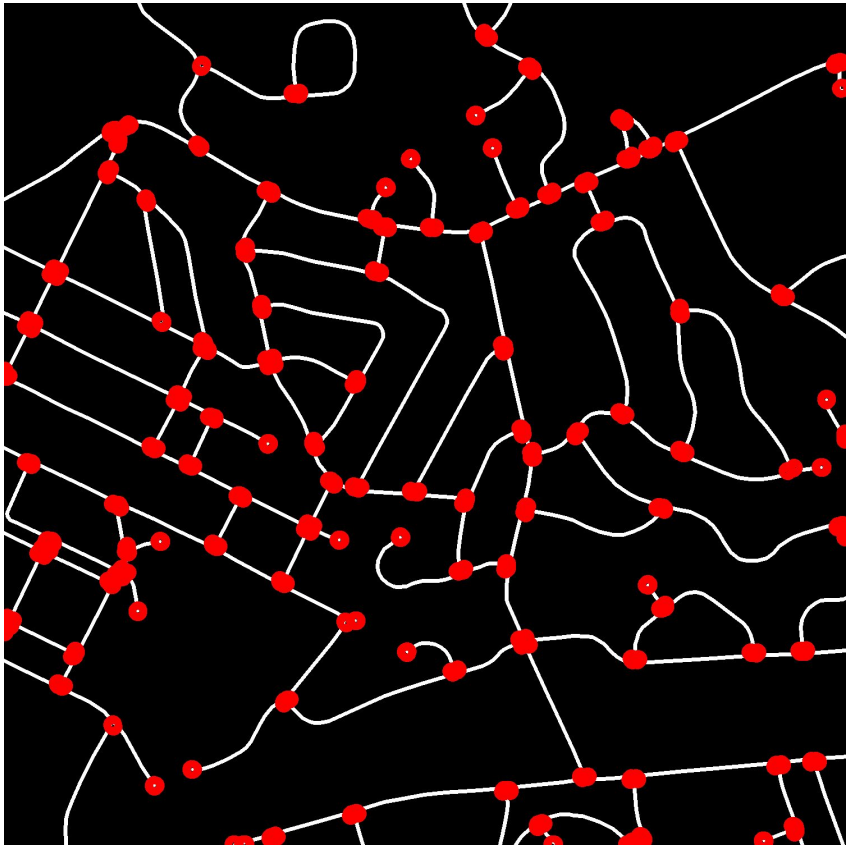


Dice loss is defined accordingly as 1 — Dice_Coefficient

★ https://cvinvolution.medium.com/dice-loss-in-medical-image-segmentation-d0e476eb486

Crossroads detection: what we have and what we want

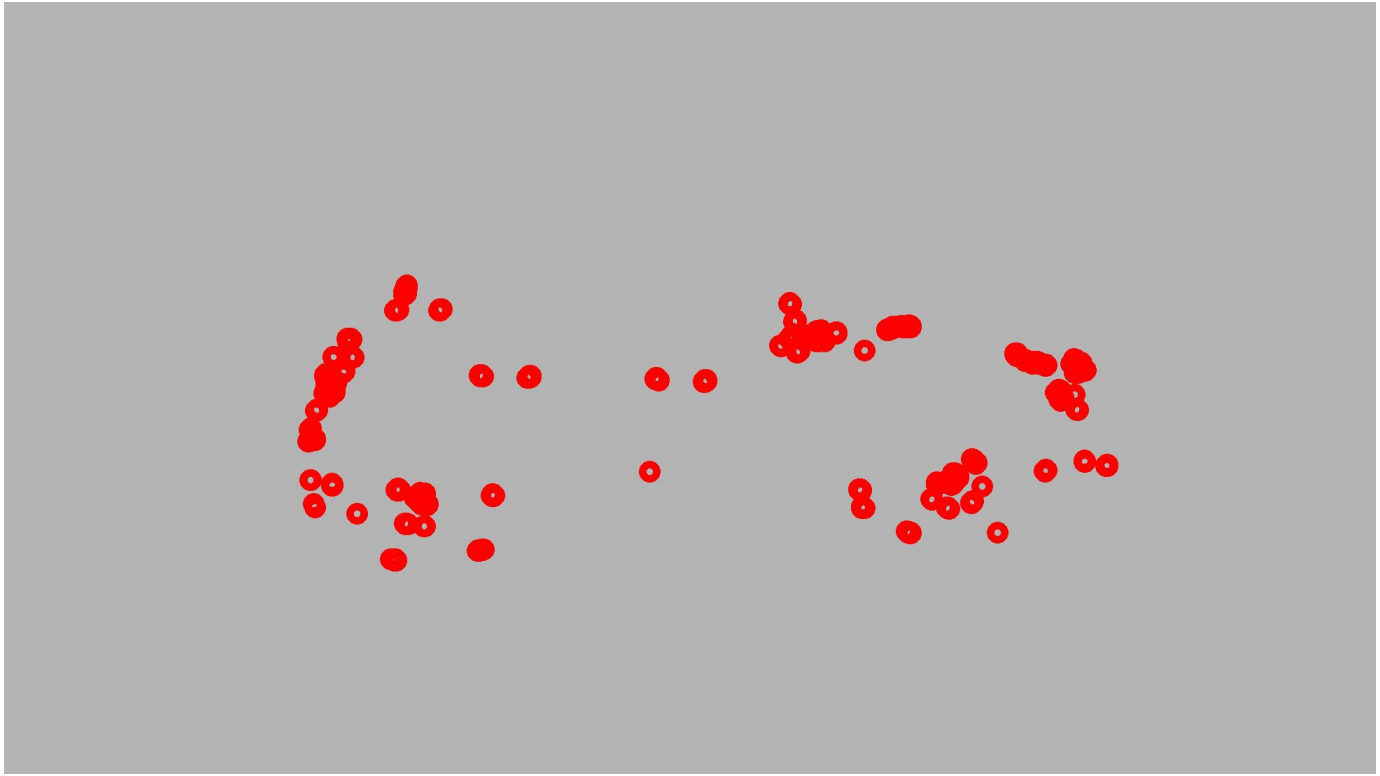




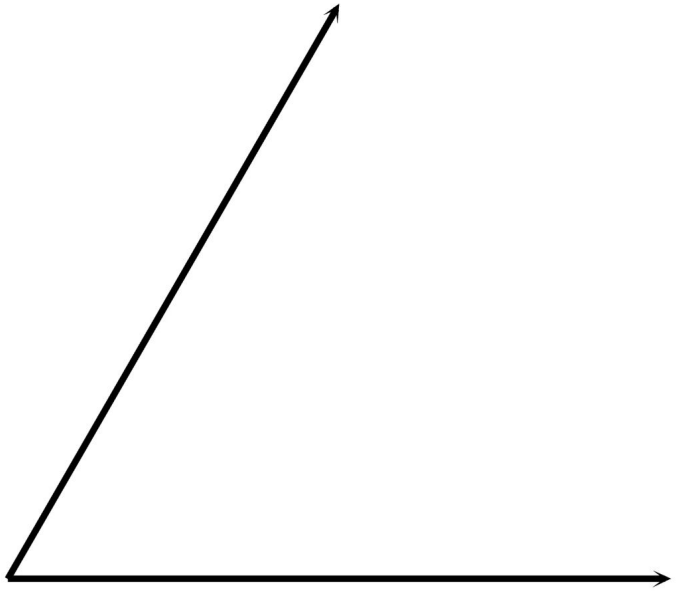
Crossroads detection: corner detector

Crossroads detection: corner detector

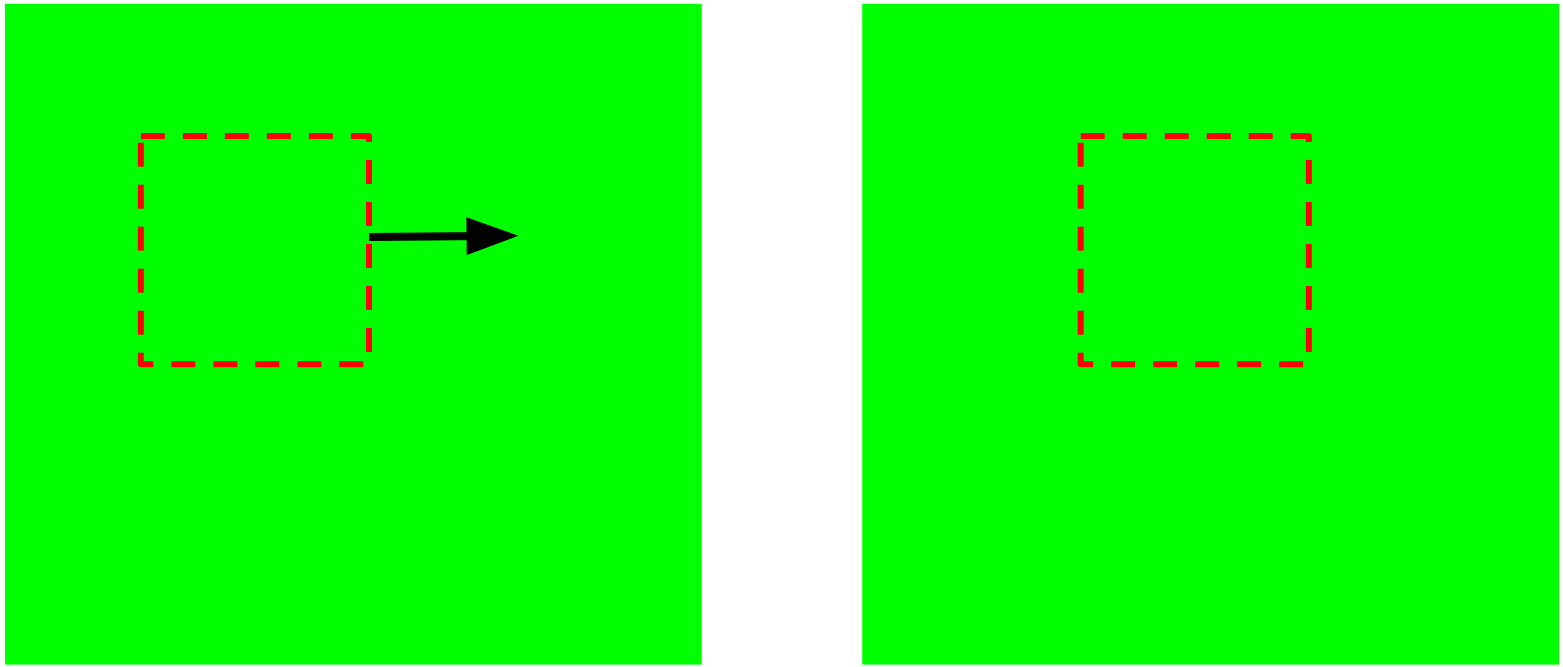
Crossroads detection: corner detector



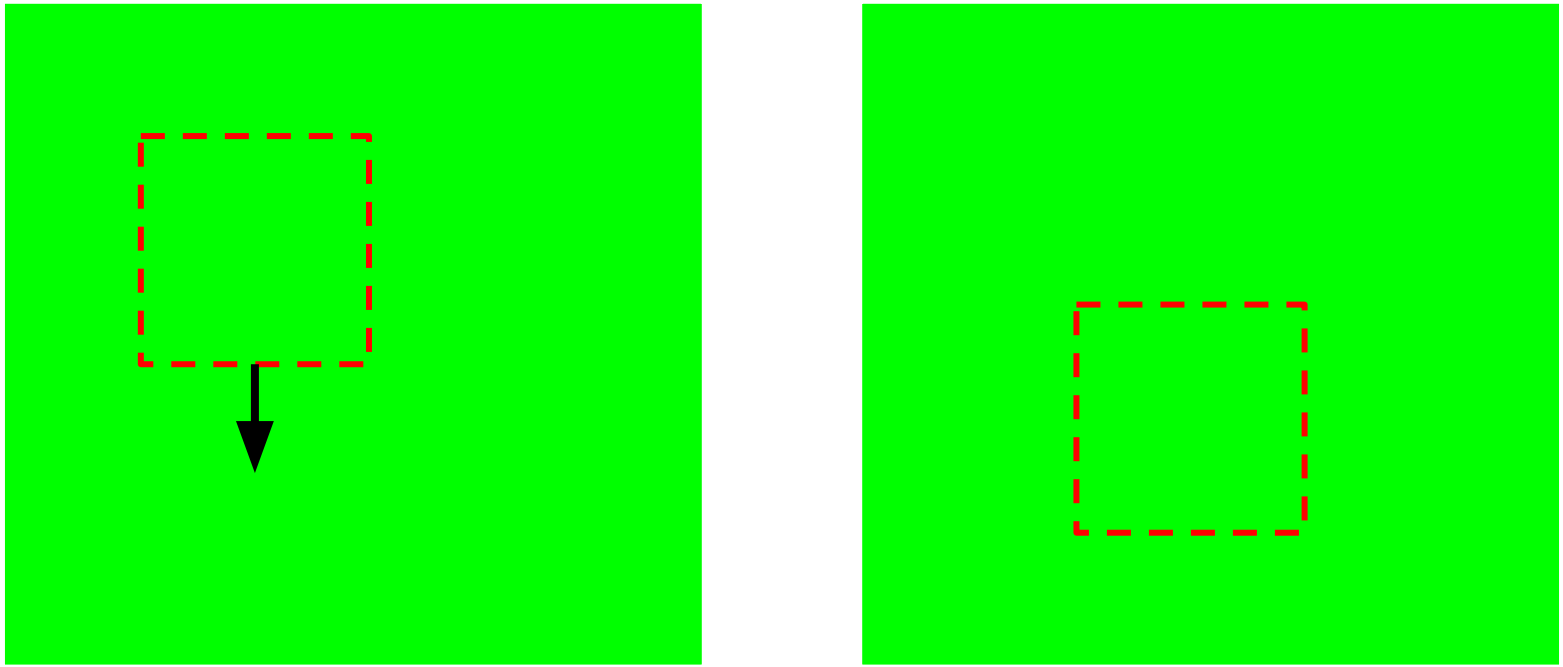
Crossroads detection: Harris corner detector



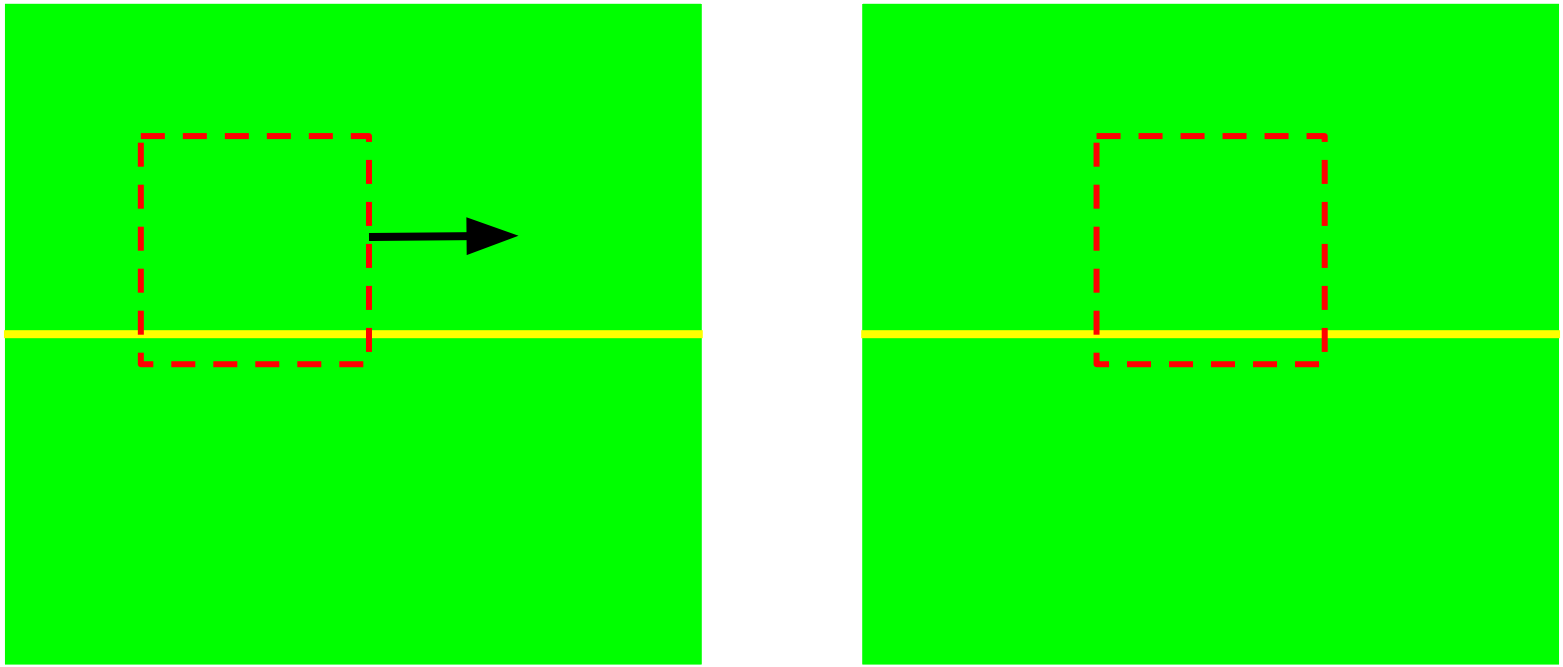
Crossroads detection: Harris corner detector approach



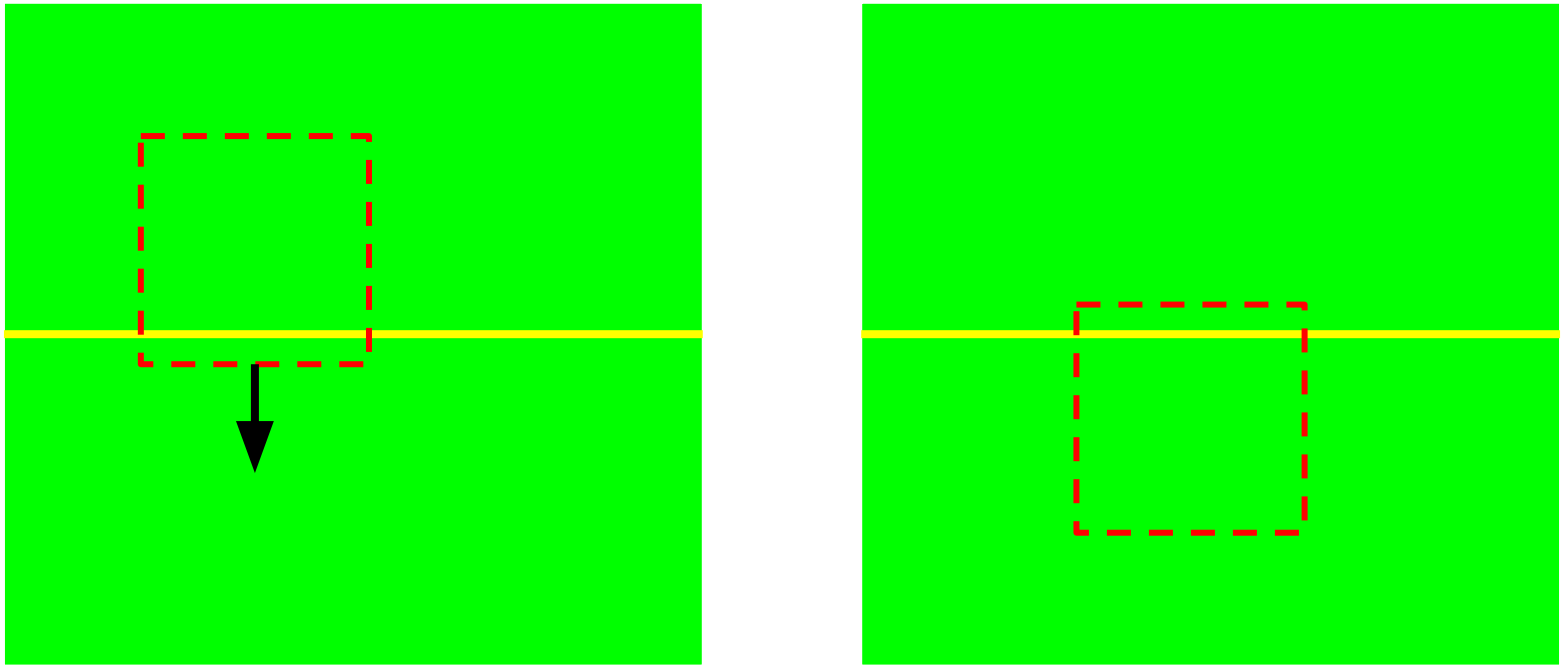
Crossroads detection: Harris corner detector approach



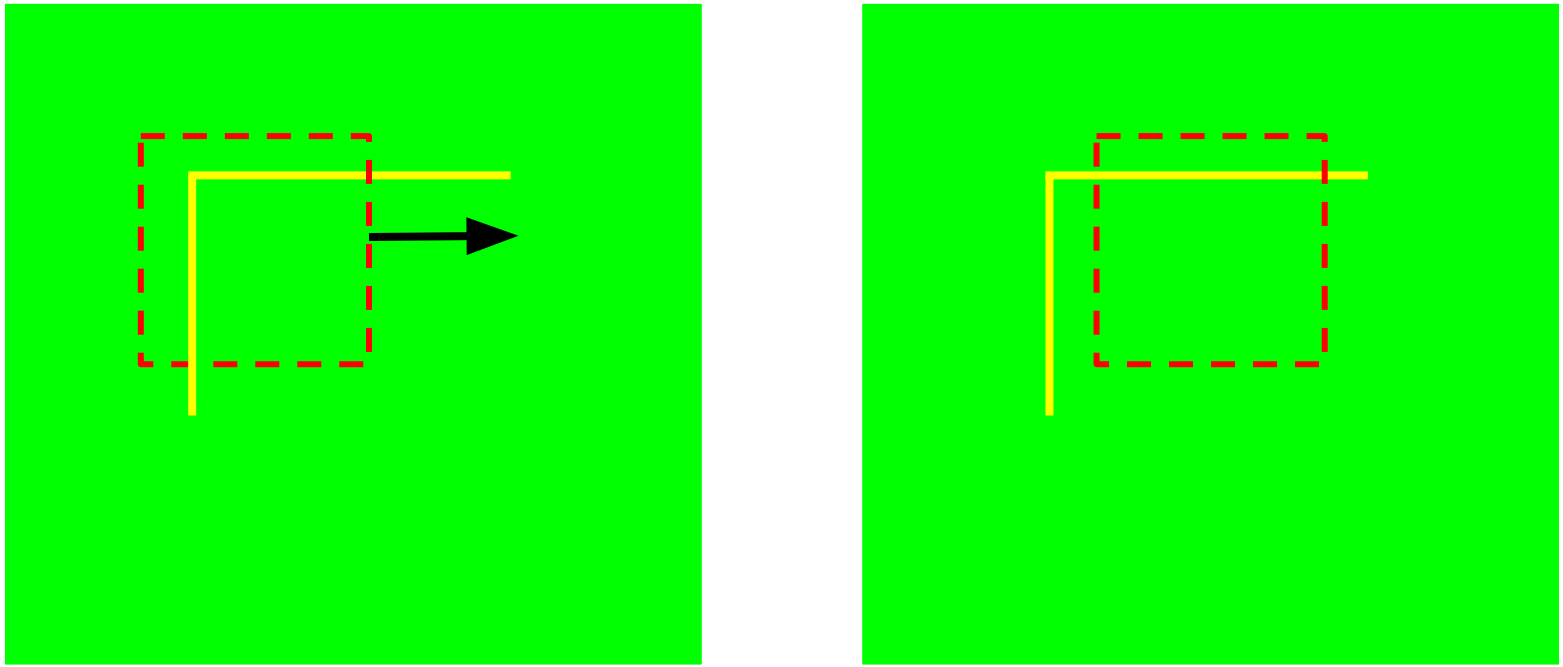
Crossroads detection: Harris corner detector approach



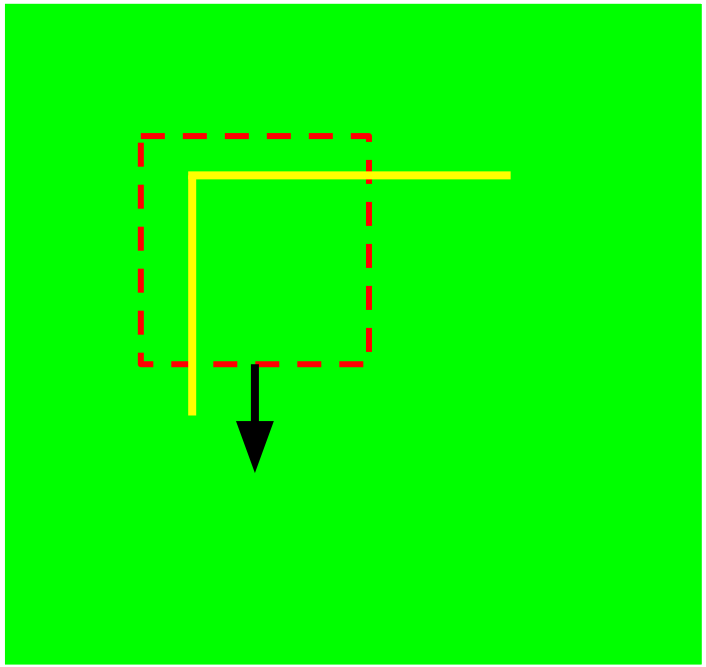
Crossroads detection: Harris corner detector approach

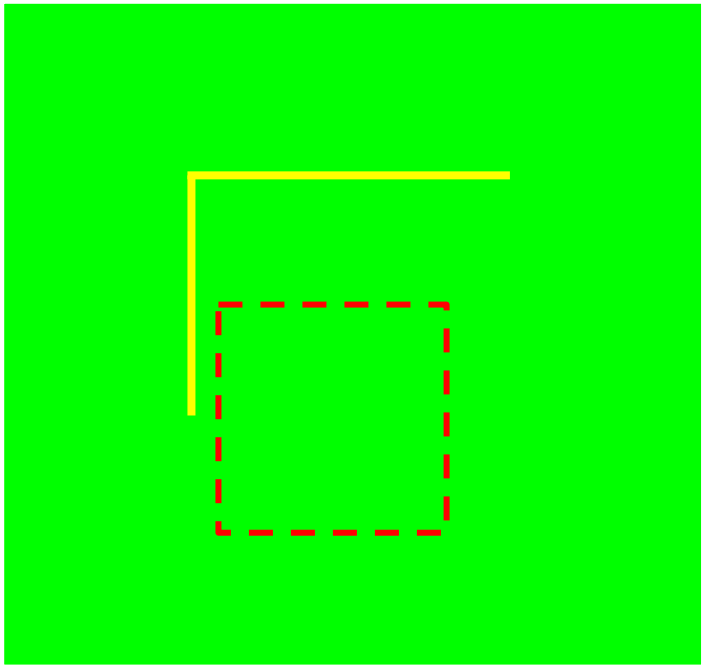


Crossroads detection: Harris corner detector approach



Crossroads detection: Harris corner detector approach





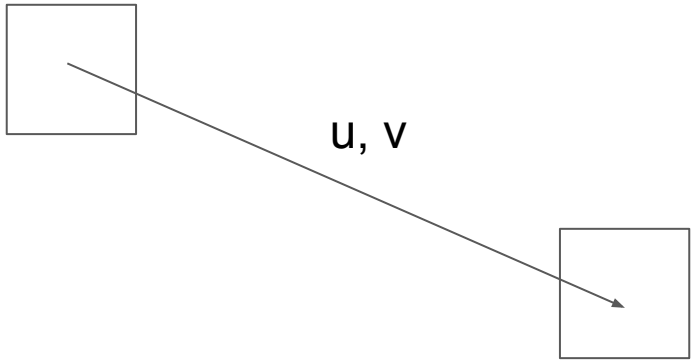
Crossroads detection: Harris corner detector approach

Idea: find points whose neighbors change greatly when shifted in any direction

Crossroads detection: Harris corner detector

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{window function}} \Big[\underbrace{I(x+u, y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)}_{\text{intensity}} \Big]^2$$

Crossroads detection: Harris corner detector



Crossroads detection: Harris corner detector

E(u, v) is large => **it's corner!**

Crossroads detection: Harris corner detector

****some math magic happens****

Crossroads detection: Harris corner detector

$$E(u,v) \approx \begin{pmatrix} u & v \end{pmatrix} M \begin{pmatrix} u \\ v \end{pmatrix}$$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

Crossroads detection: Harris corner detector

****some math magic happens****

Crossroads detection: Harris corner detector

$$R = \det M - k \cdot \mathrm{trace}^2(M)$$

- $\det M = \lambda_1 \lambda_2;$
- $\mathrm{trace}(M) = \lambda_1 + \lambda_2;$

Crossroads detection: Harris corner detector

- $|R|$ is small $\Rightarrow$ flat
- $R < 0 \Rightarrow$ edge
- R is large $\Rightarrow$ corner

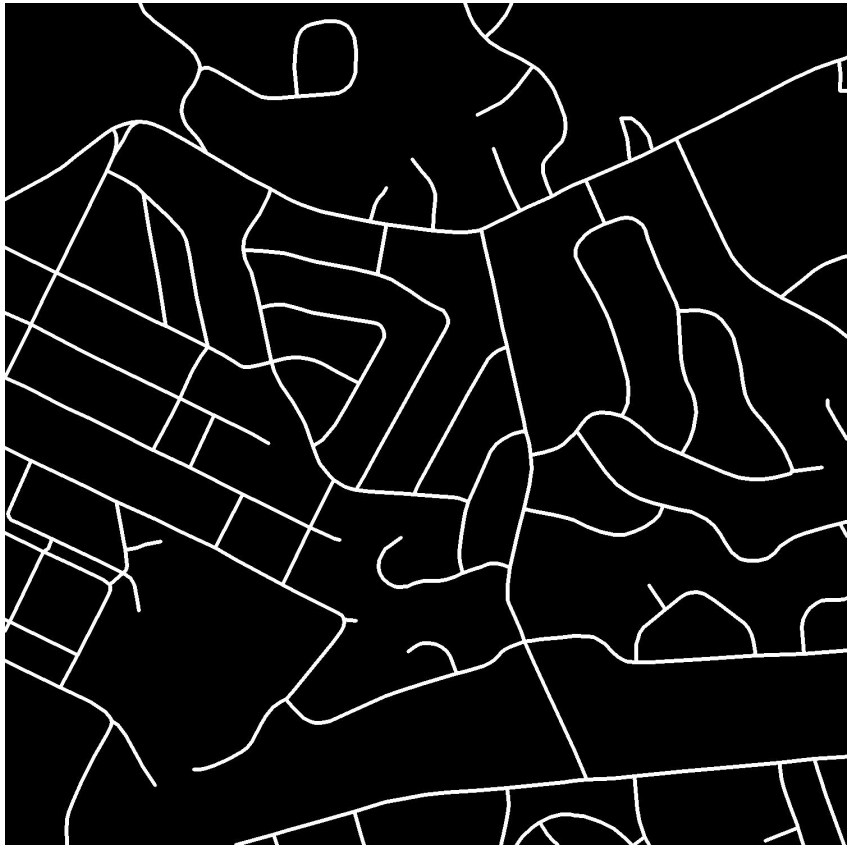
Crossroads detection: Harris corner detector

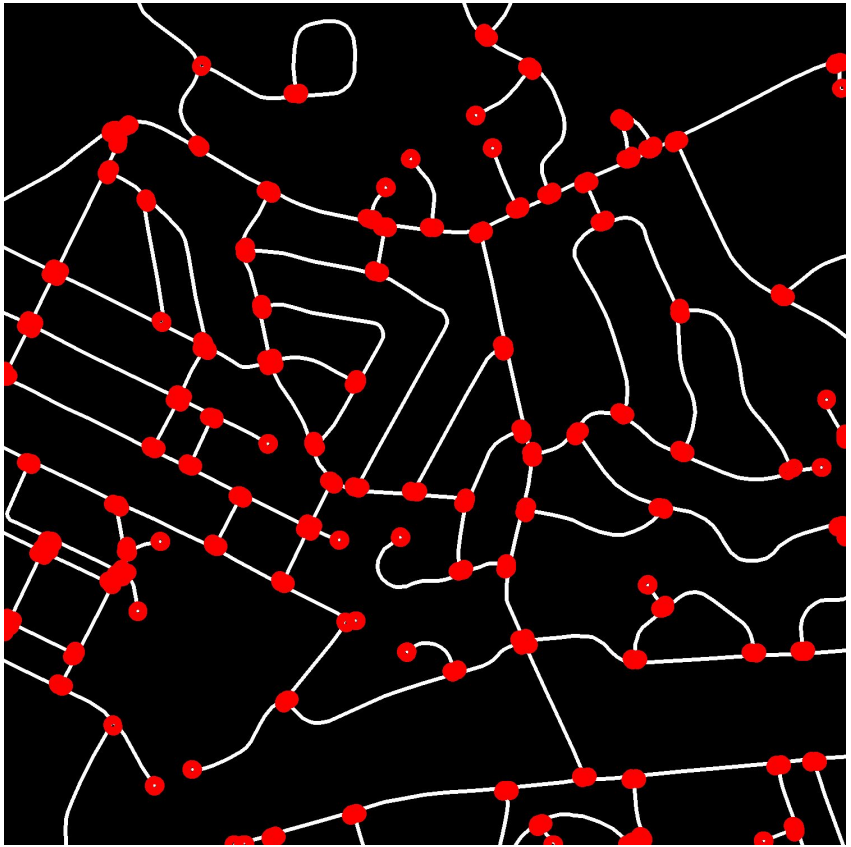
```python
import cv2

image = cv2.imread(f"image.png")
image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

harris_result = cv2.cornerHarris(
    src=image,
    blockSize=3,  # Size of neighborhood
    ksize=9,  # Sobel parameter
    k=0.06  # Empirical
)
```

Crossroads detection: Harris corner detector

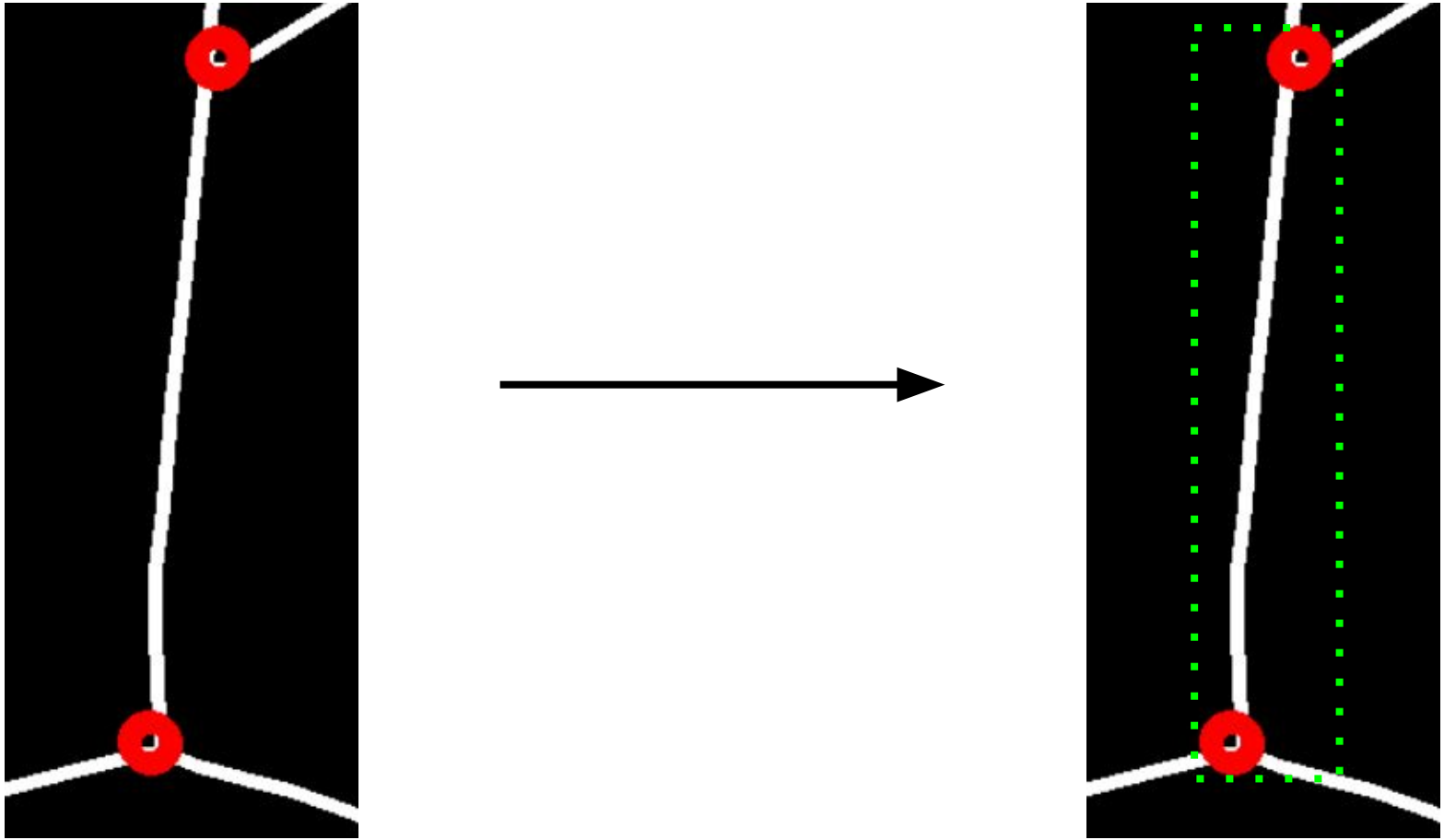
```
import networkx as nx

graph = nx.Graph()
height, width = harris_result.shape
THRESHOLD = 200

for y in range(height):
    for x in range(width):
        r = harris_result[y, x]
        if r > THRESHOLD:
            graph.add_node((x, y))
```

Crossroads detection: result





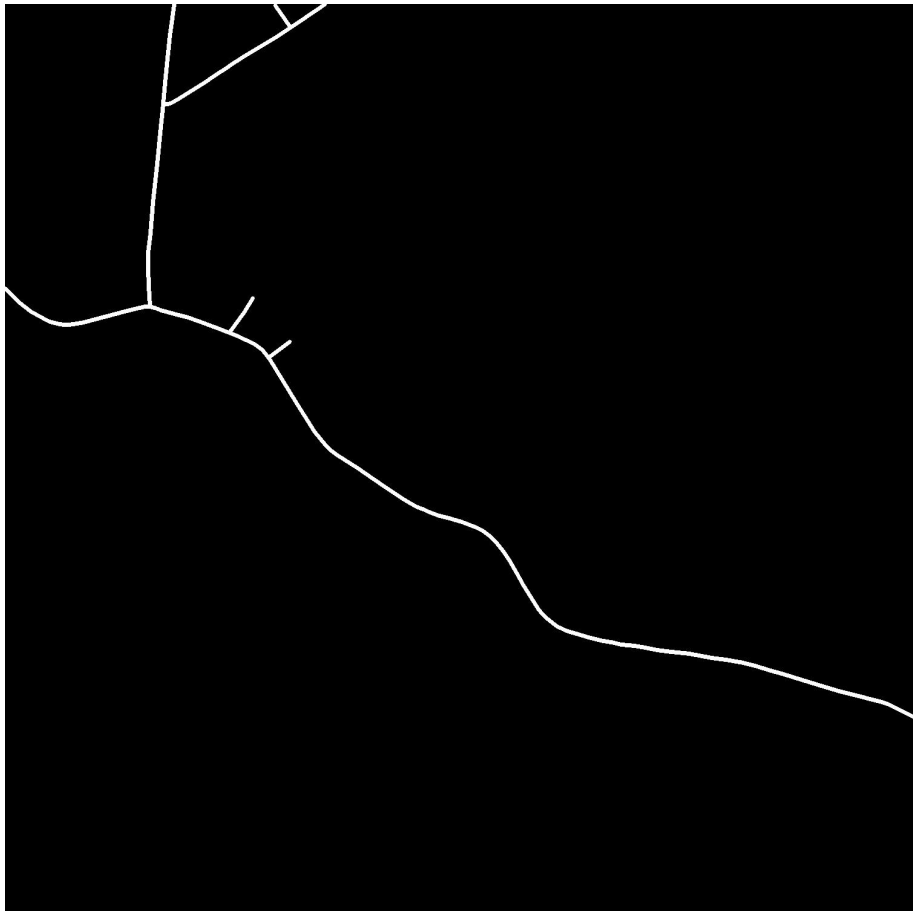
Making a graph



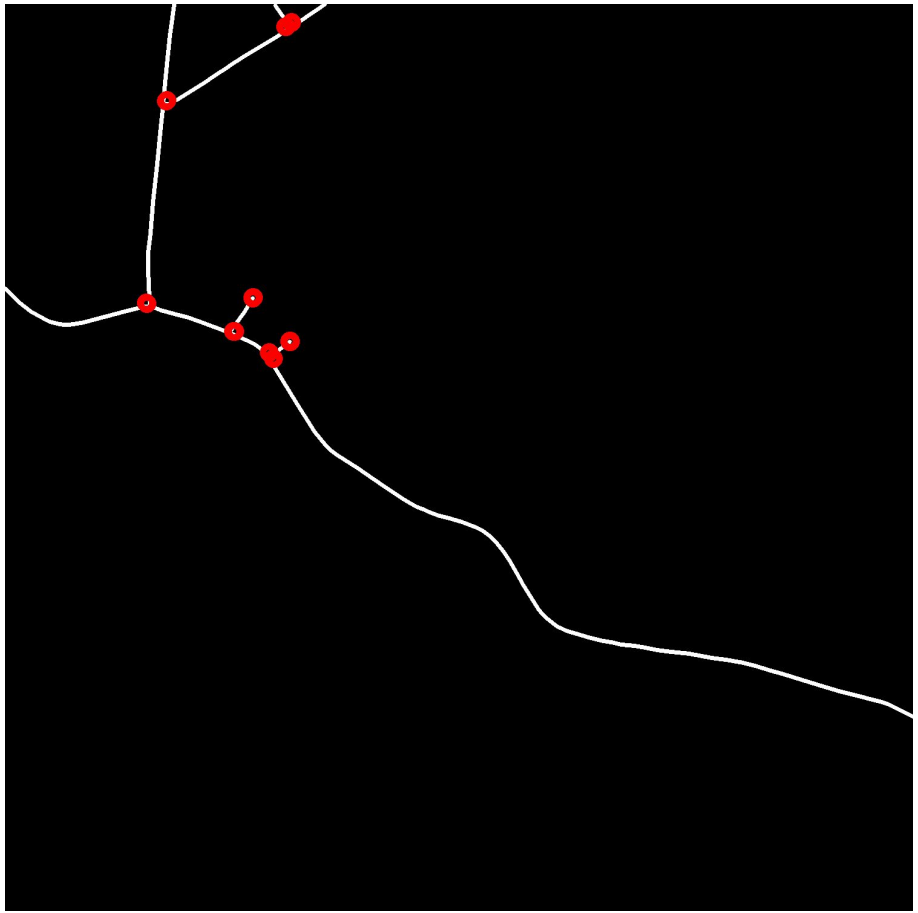
Making a graph

```
EDGE_THRESHOLD = 24

for i in range(len(graph.nodes)):
    n1 = graph.nodes[i]
    x1 = n1[0]
    y1 = n1[1]
    for j in range(i + 1, len(graph.nodes)):
        n2 = graph.nodes[j]
        x2 = n2[0]
        y2 = n2[1]
        min_x = min(x1, x2)
        max_x = max(x1, x2)
        min_y = min(y1, y2)
        max_y = max(y1, y2)
        intensity = image[min_y:max_y, min_x:max_x].sum()
        size1 = max_x - min_x + 1
        size2 = max_y - min_y + 1
        area = size1 * size2
        intensity /= area
        if intensity > EDGE_THRESHOLD:
            graph.add_edge(n1, n2)
```

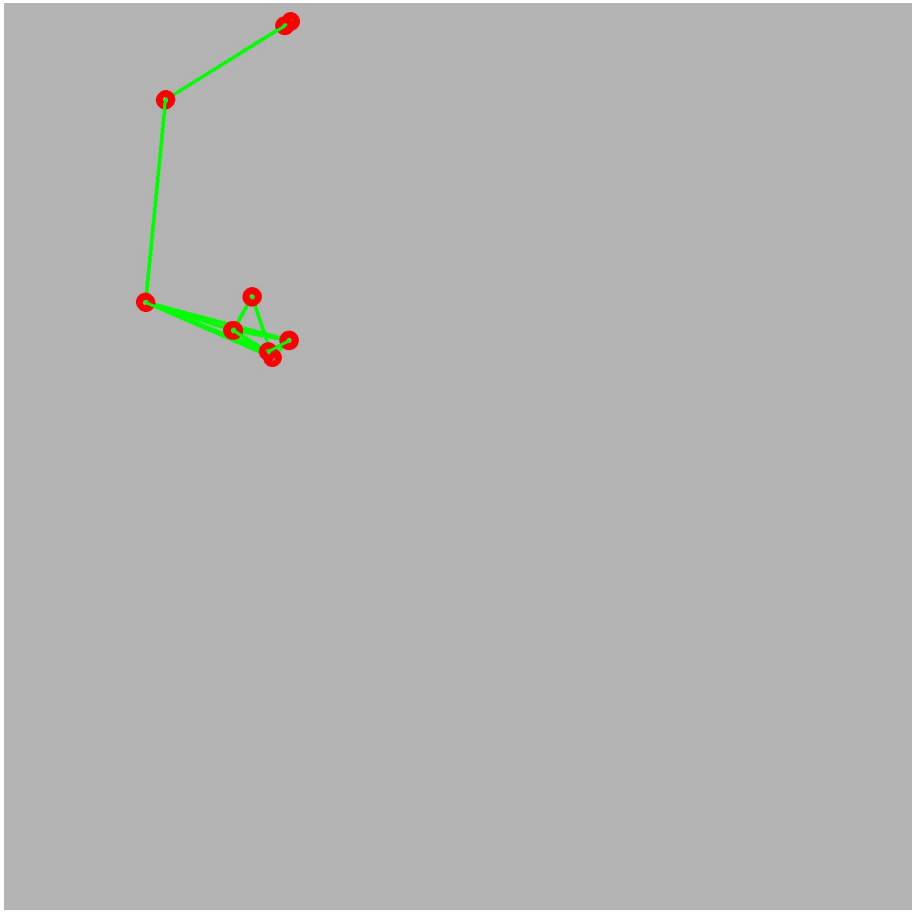
Crossroads detection + Making a graph: result-1



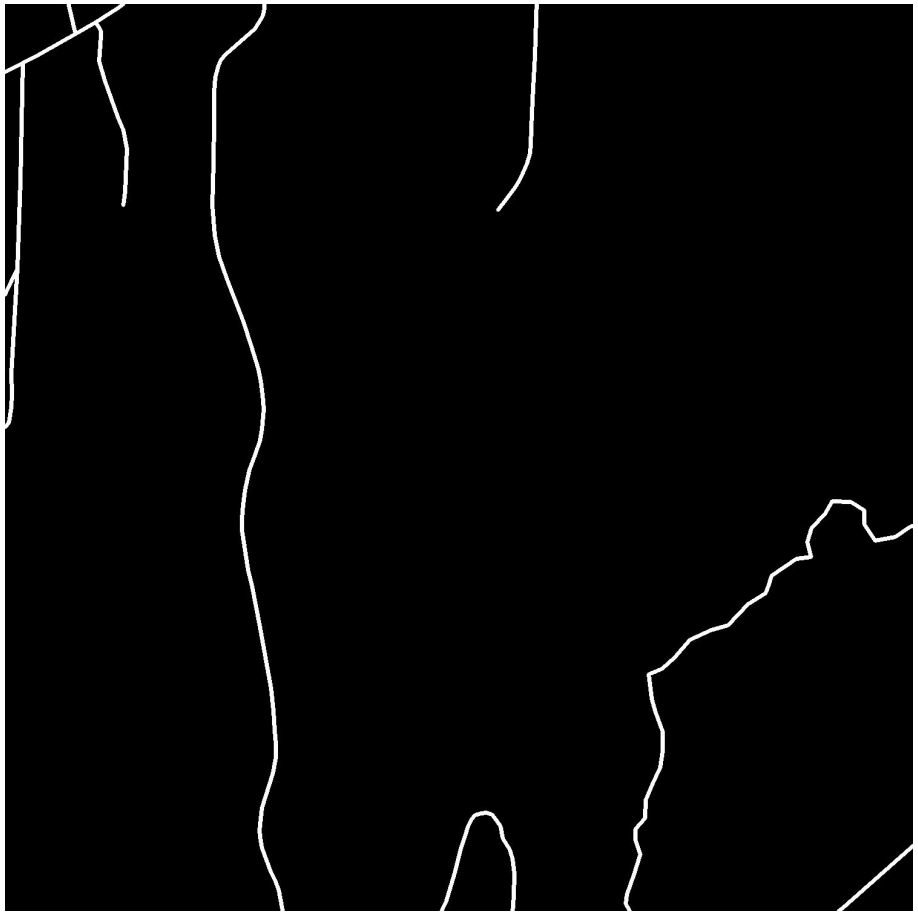
Crossroads detection + Making a graph: result-1



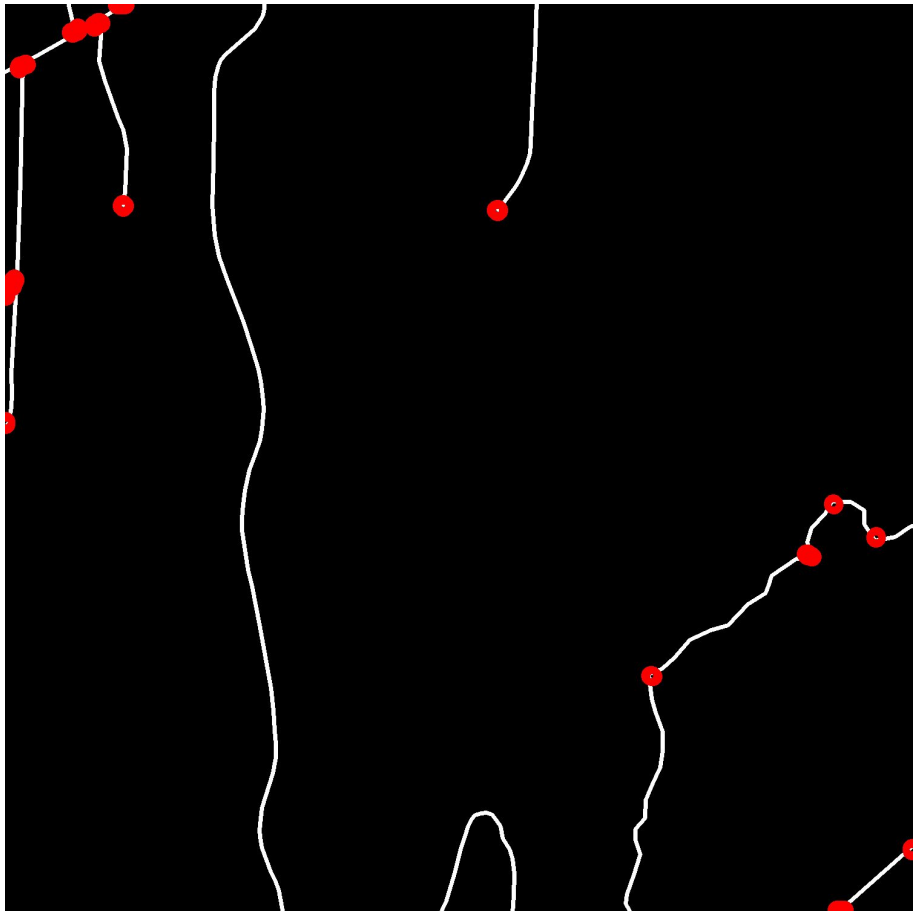
Crossroads detection + Making a graph: result-1



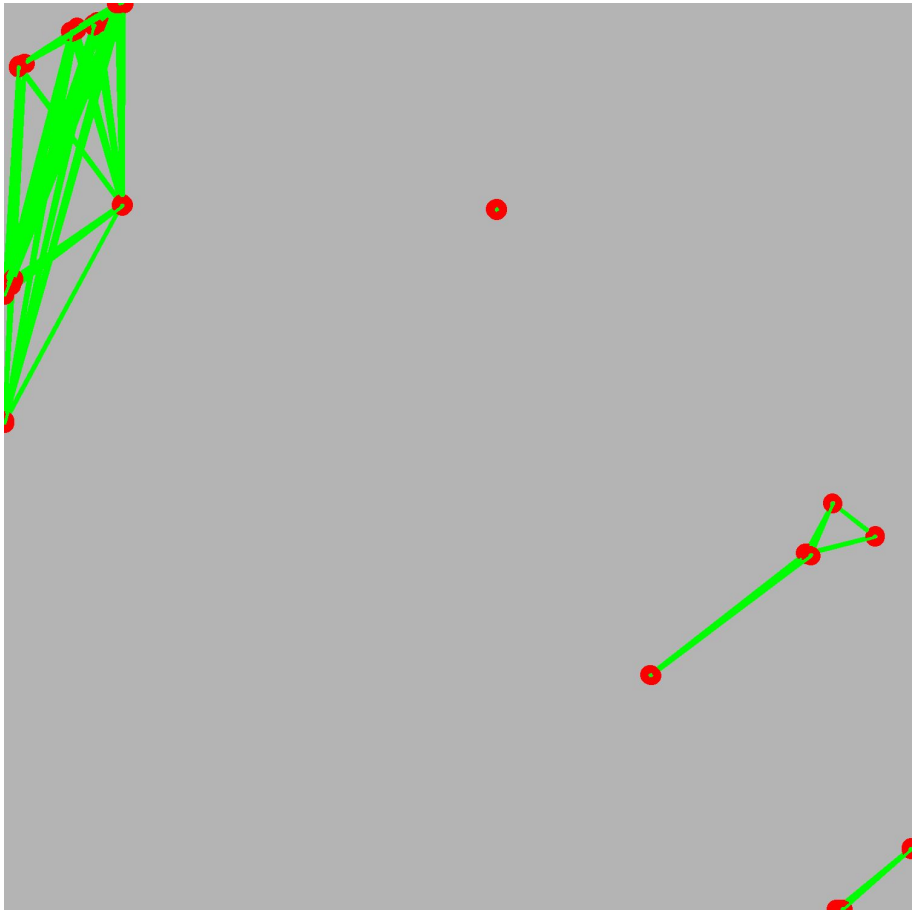
Crossroads detection + Making a graph: result-2



Crossroads detection + Making a graph: result-2

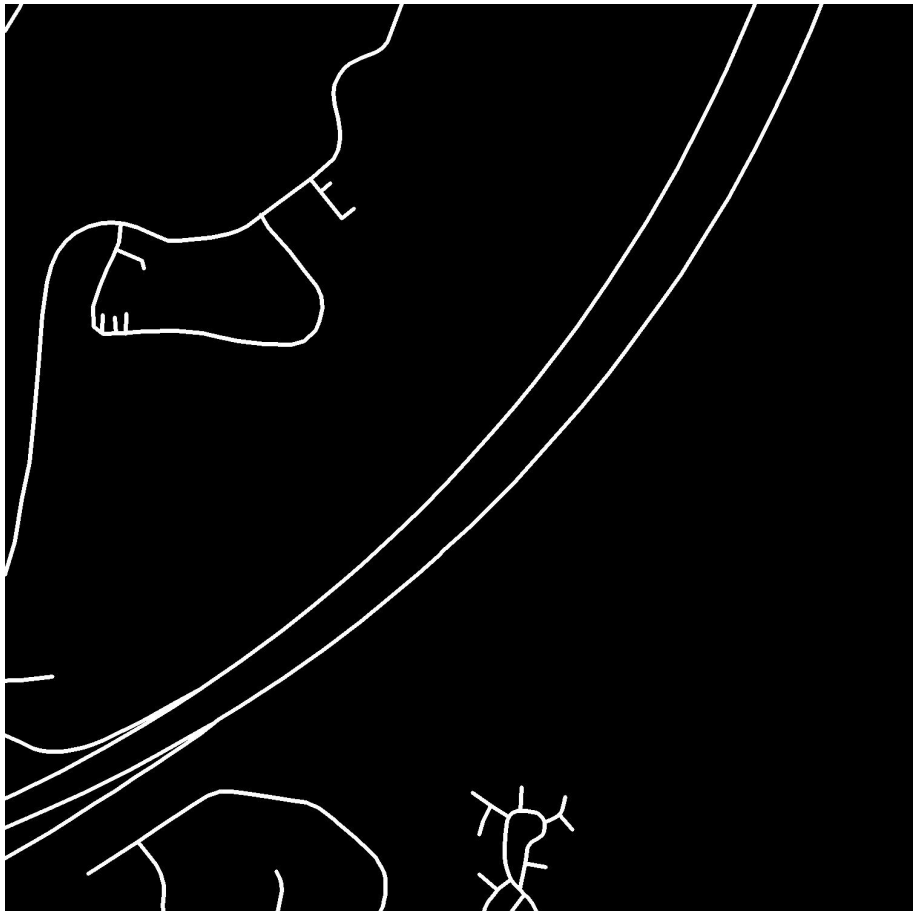


Crossroads detection + Making a graph: result-2

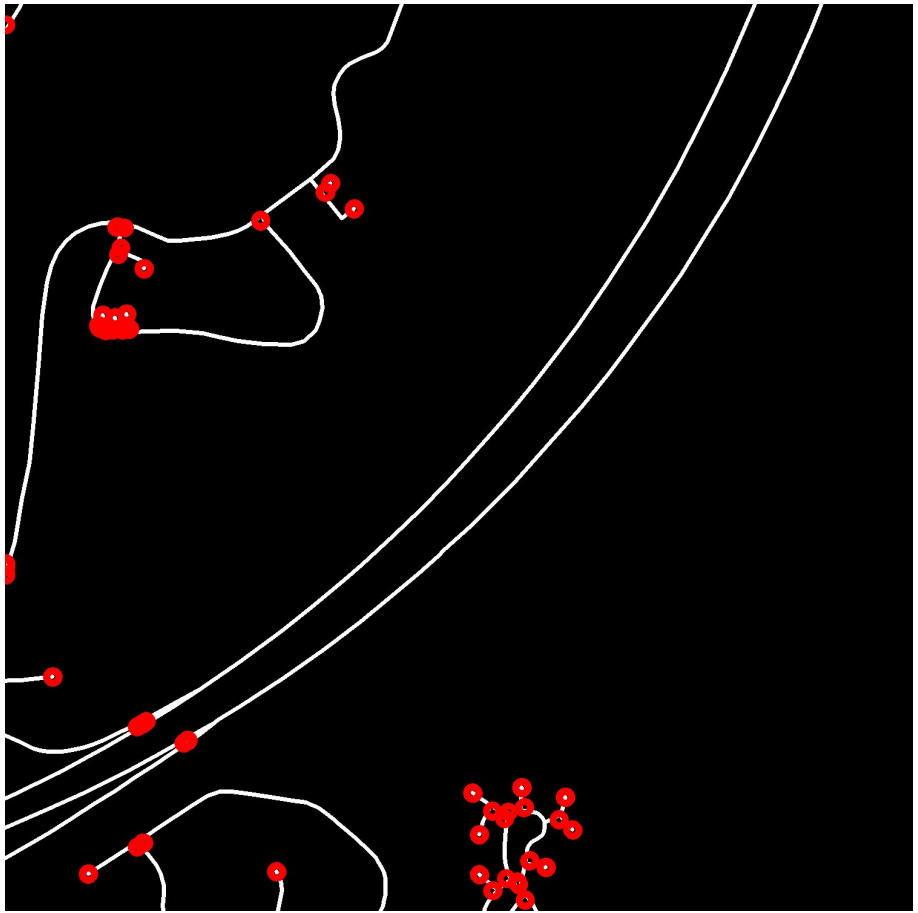


Crossroads detection + Making a graph: result-2

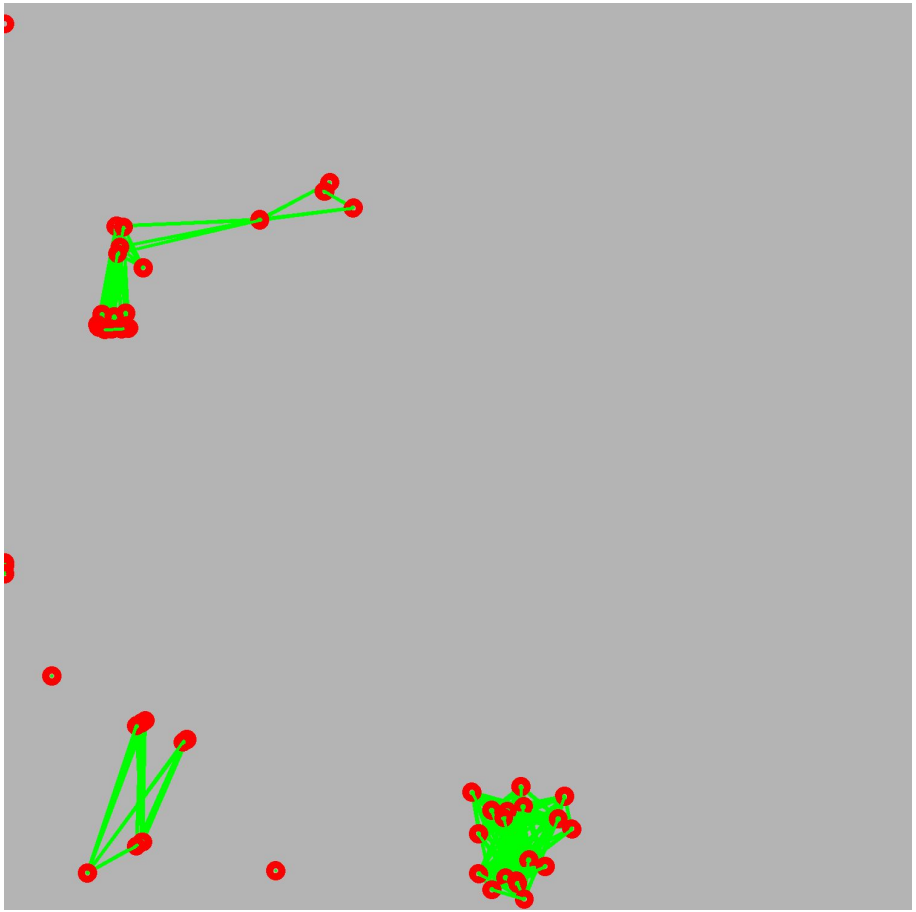
Crossroads detection + Making a graph: result-3



Crossroads detection + Making a graph: result-3

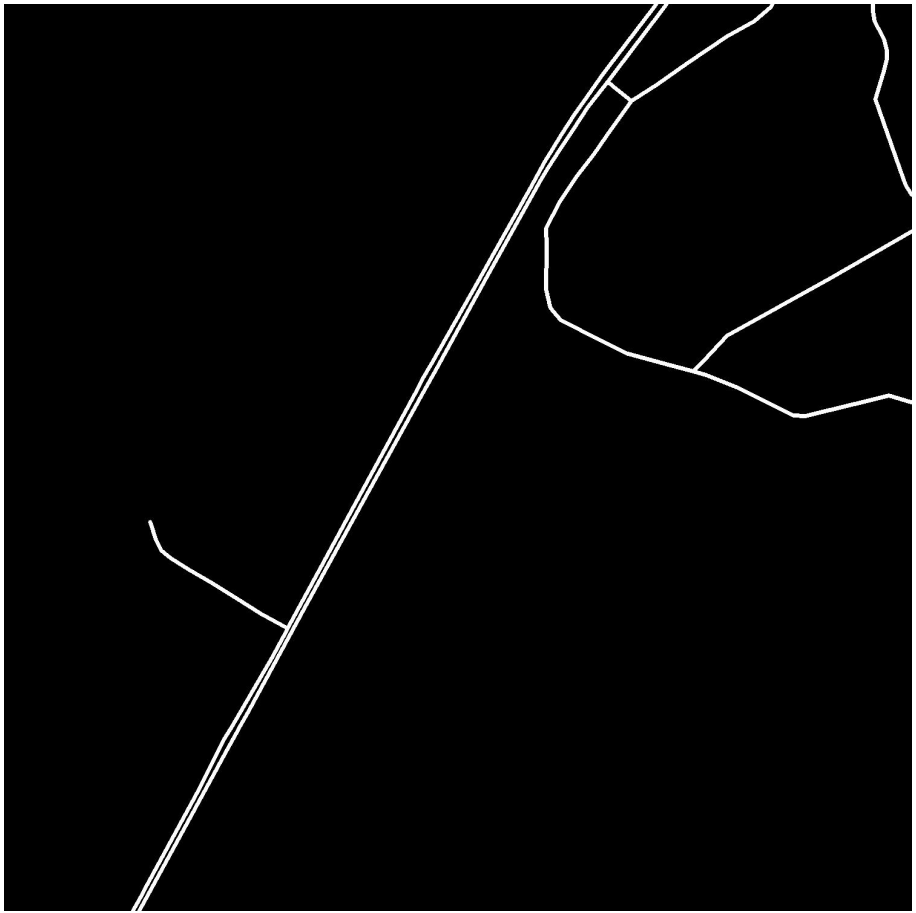


Crossroads detection + Making a graph: result-3

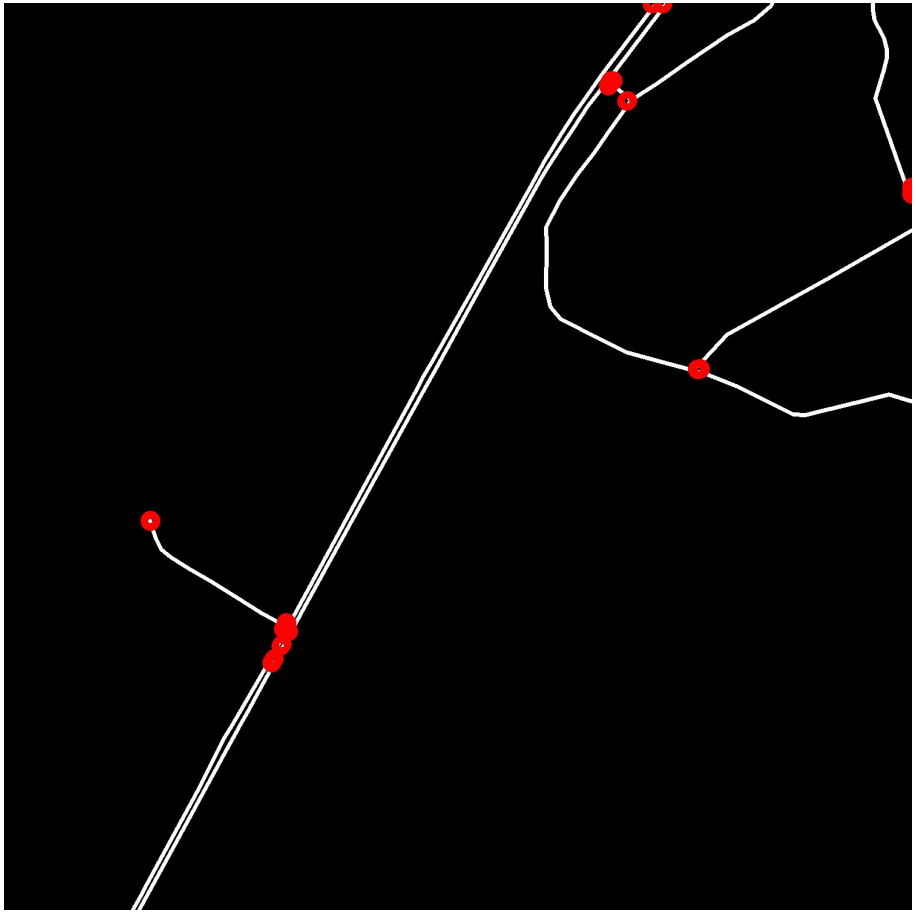


Crossroads detection + Making a graph: result-3

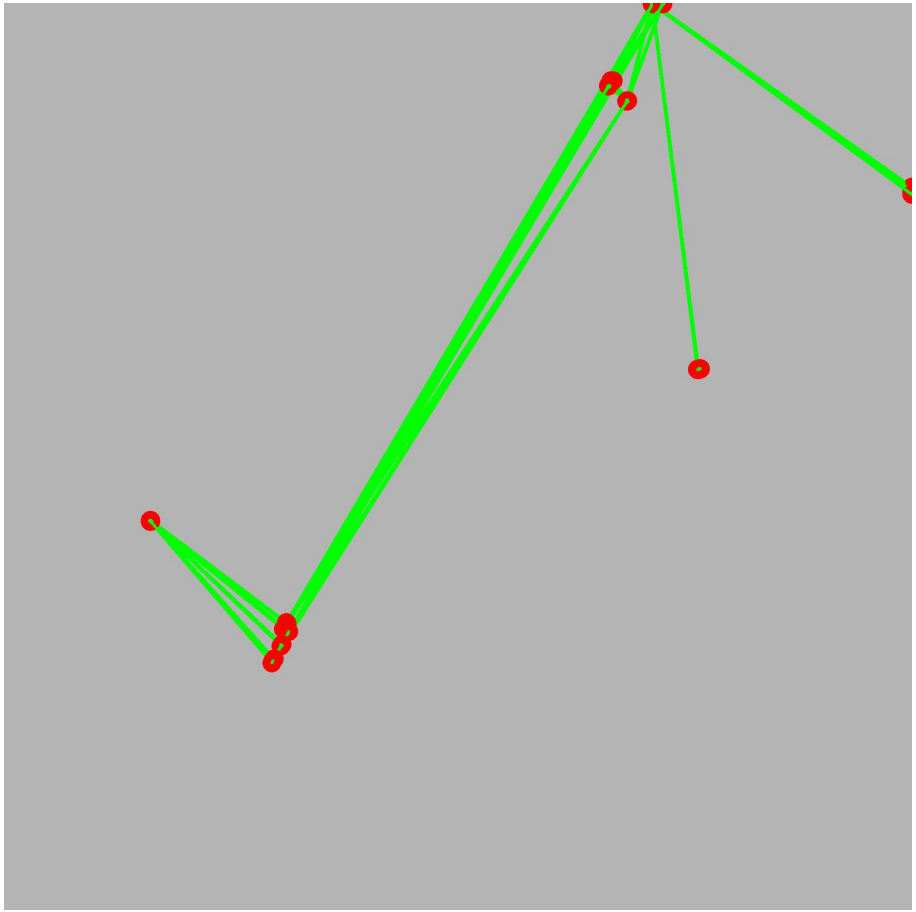
Crossroads detection + Making a graph: result-4



Crossroads detection + Making a graph: result-4

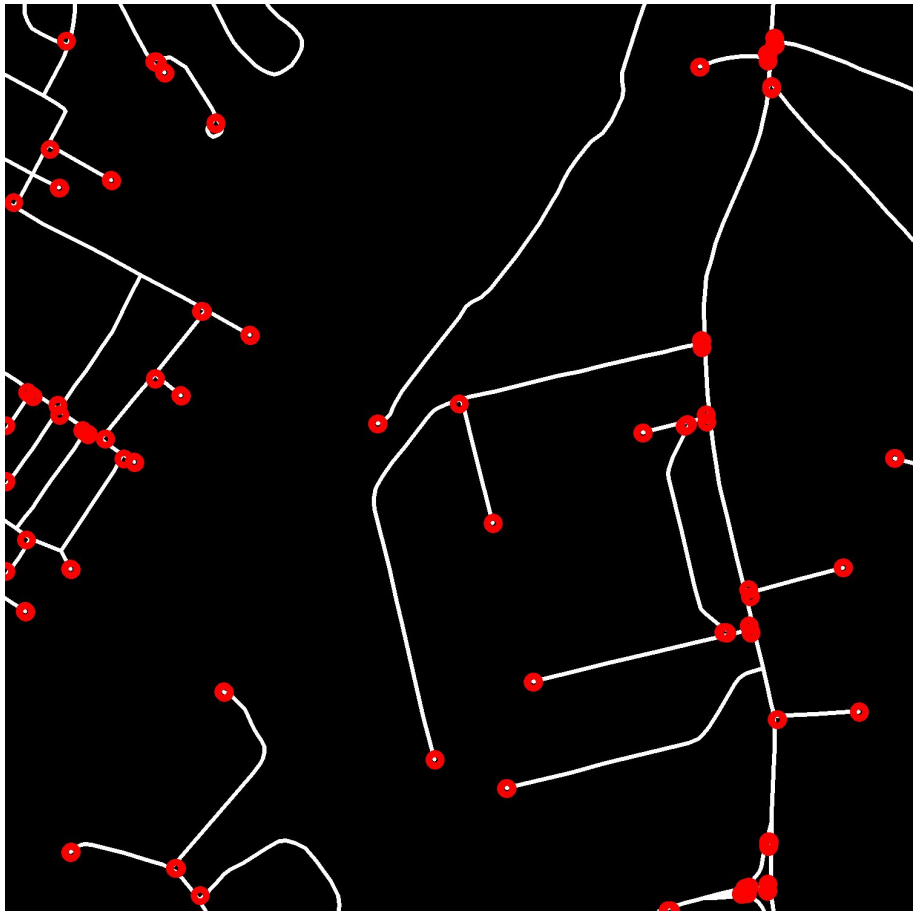


Crossroads detection + Making a graph: result-4

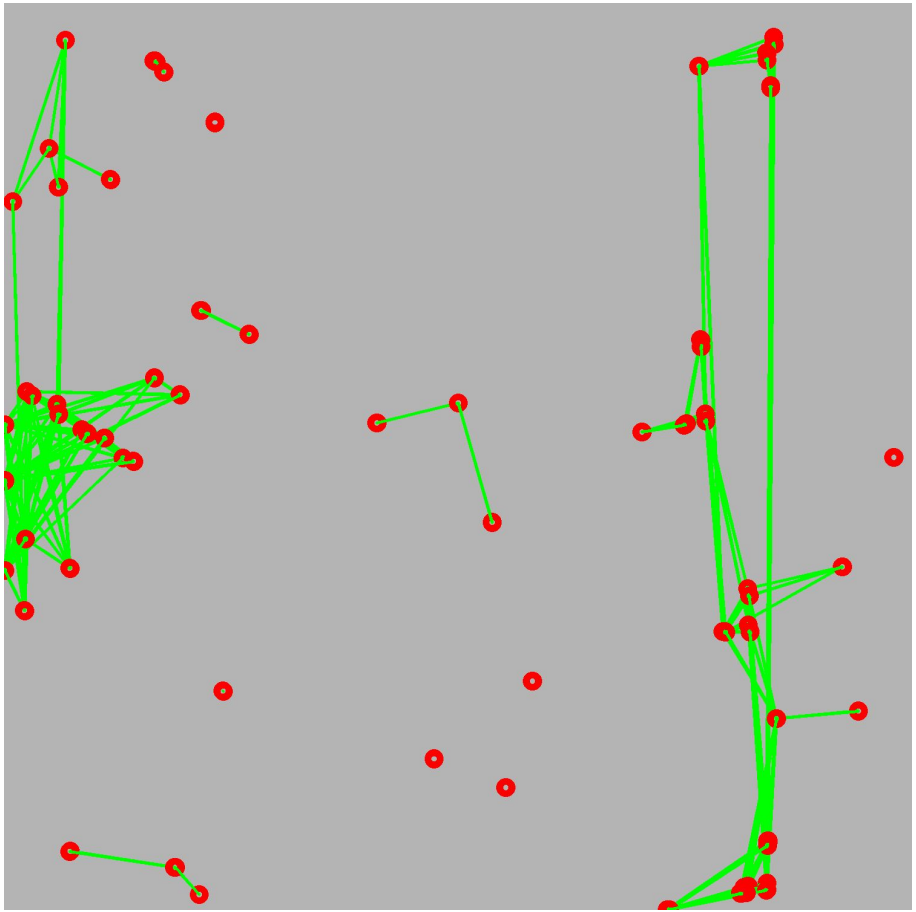


Crossroads detection + Making a graph: result-5

Crossroads detection + Making a graph: result-5



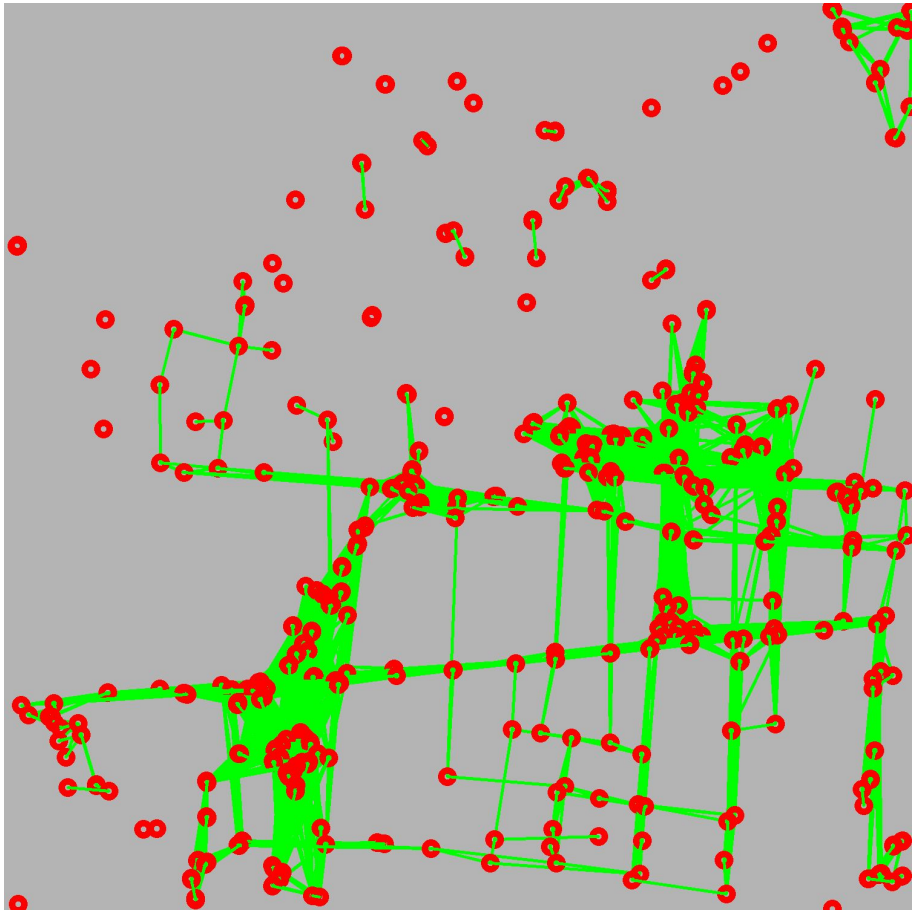
Crossroads detection + Making a graph: result-5



Crossroads detection + Making a graph: result-6

Crossroads detection + Making a graph: result-6

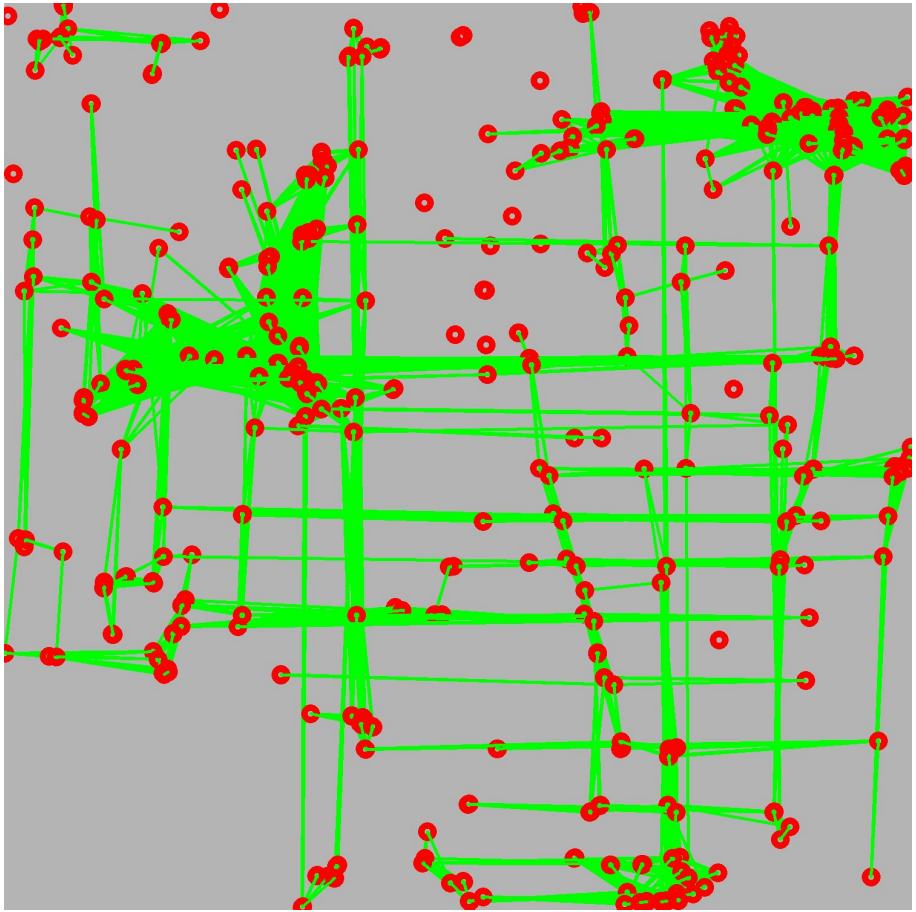
Crossroads detection + Making a graph: result-6



Crossroads detection + Making a graph: result-7

Crossroads detection + Making a graph: result-7

Crossroads detection + Making a graph: result-7



Спасибо за внимание