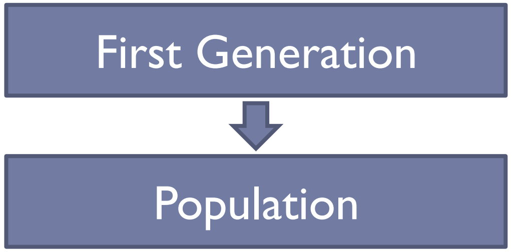
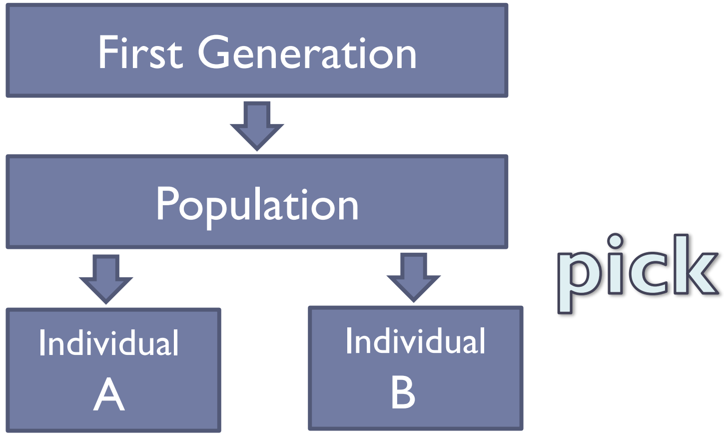
Using Evolutionary Algorithm to find image segmentation

Yossef Kitrossky & Yoad Lewenberg

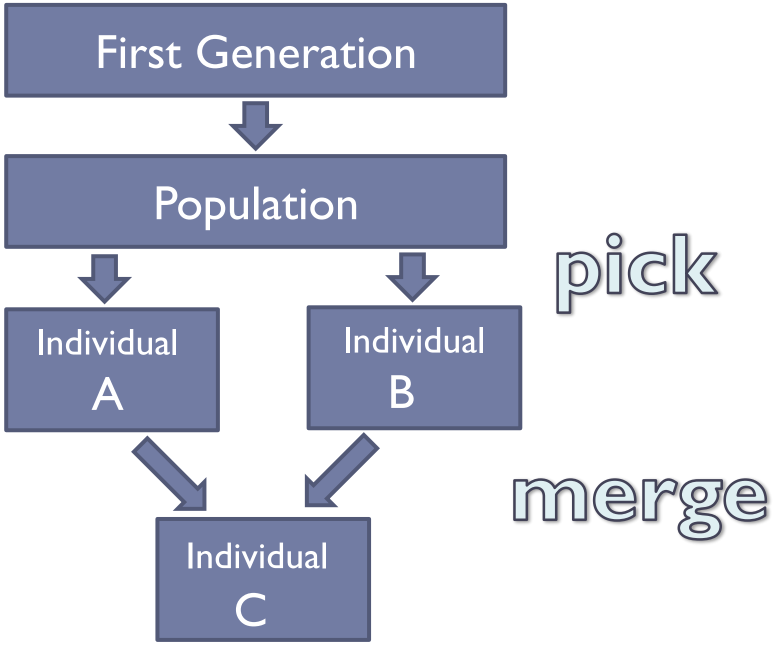
Evolutionary Algorithm



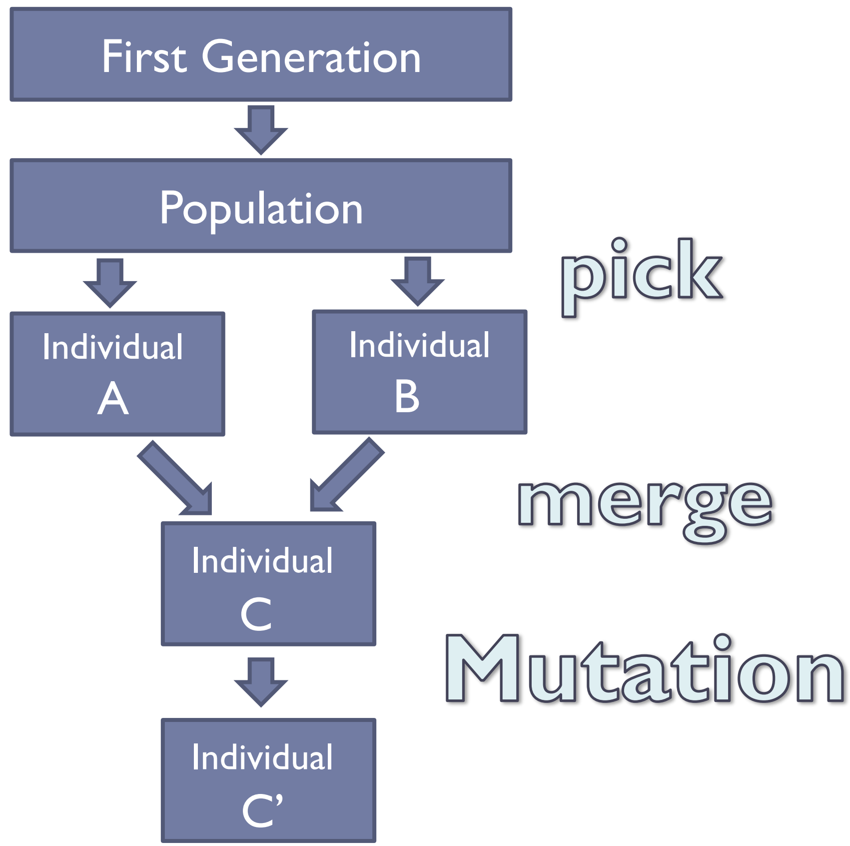
Evolutionary Algorithm



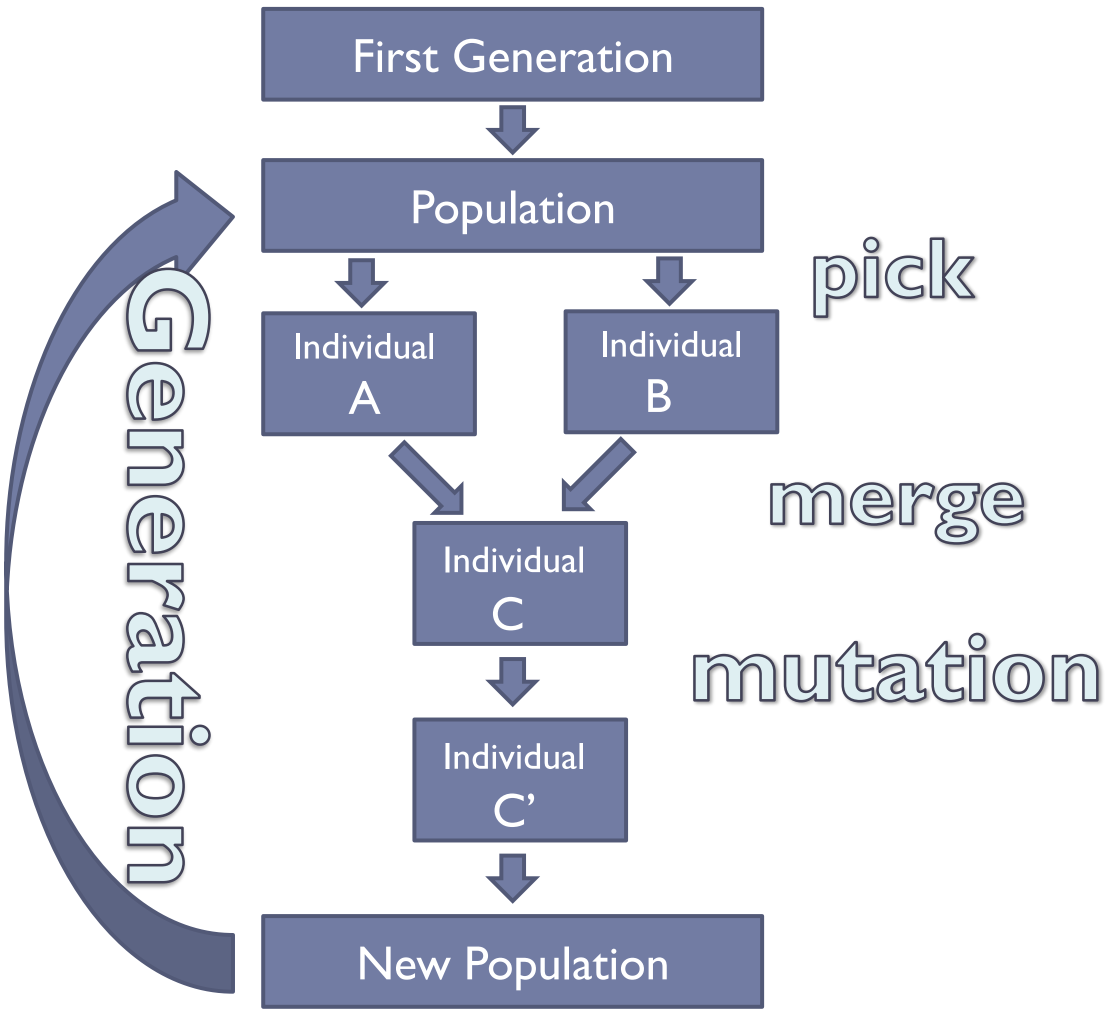
Evolutionary Algorithm



Evolutionary Algorithm



Evolutionary Algorithm

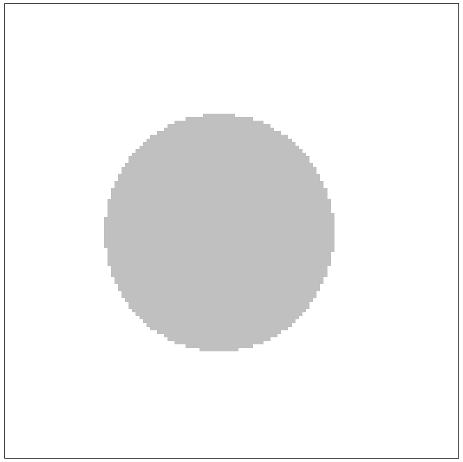


First Generation

- Random Matrix
- Circles and rectangle

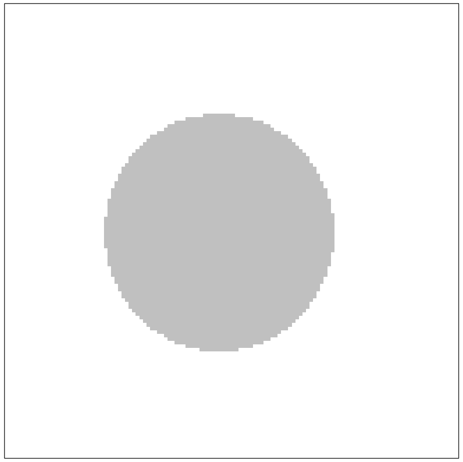
First Generation

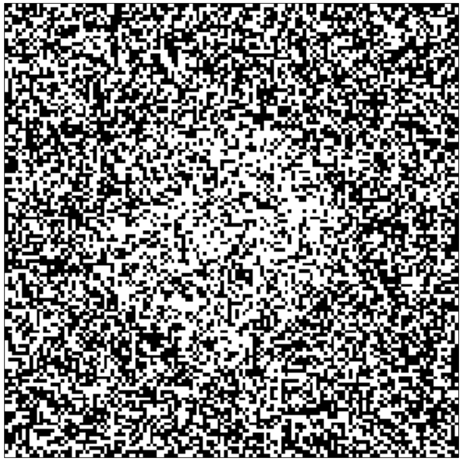
- Random Matrix
- Circles and rectangles



First Generation

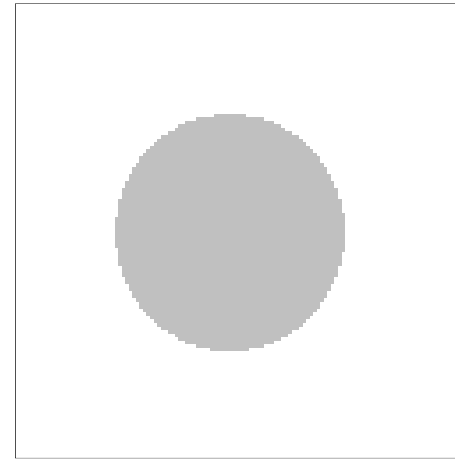
- Random Matrix
- Circles and rectangle

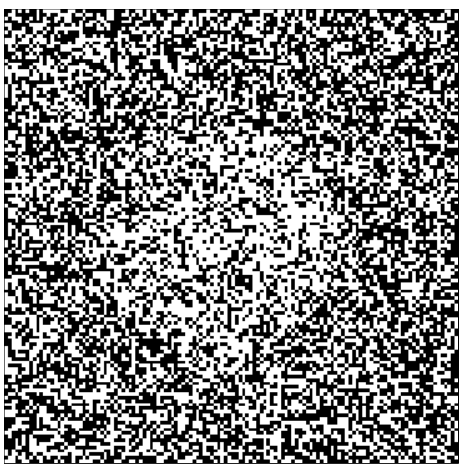


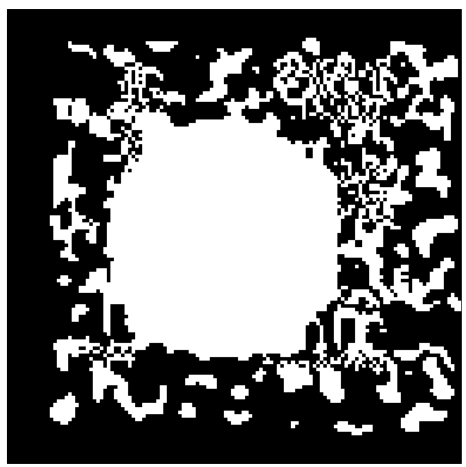


First Generation

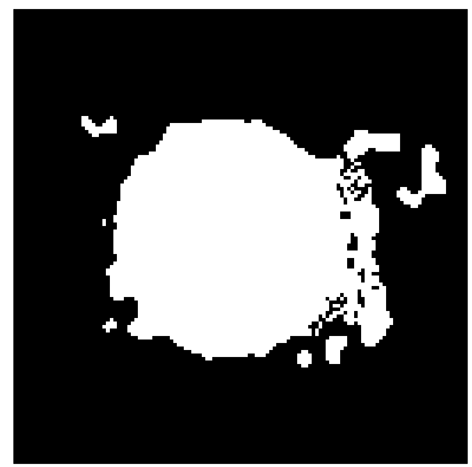
- Random Matrix
- Circles and rectangle







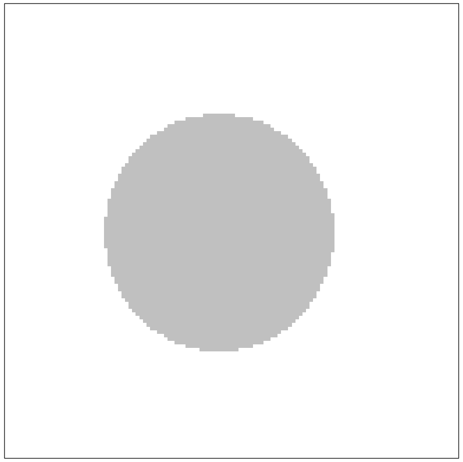
Mutation probability 0.02

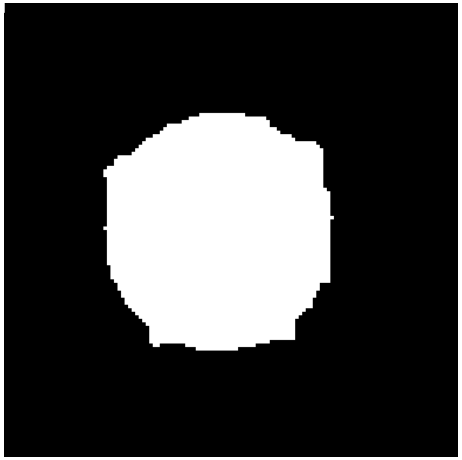


Mutation probability 0.2

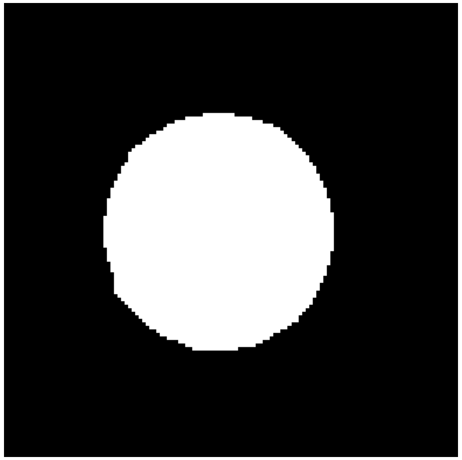
First Generation

- Random Matrix
- Circles and rectangles





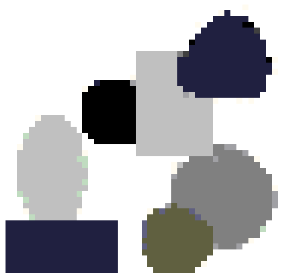
Mutation probability 0.02



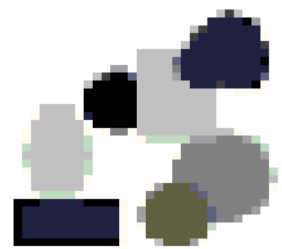
Mutation probability 0. 2

Evolution

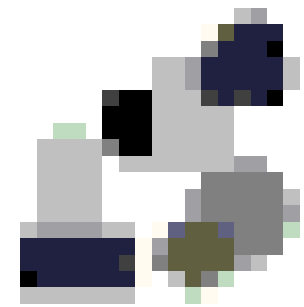
- Reducing image resolution



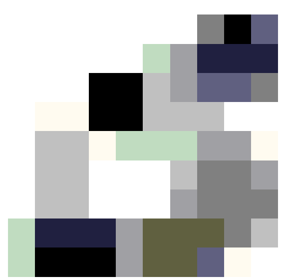
128*128



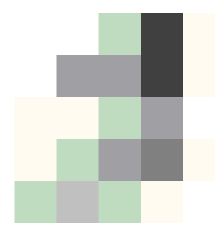
64*64



32*32

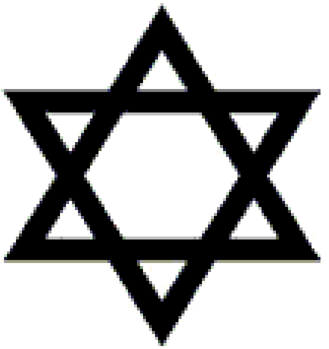


16*16

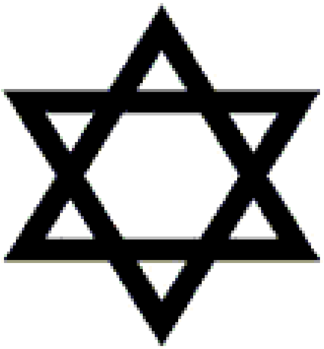


8*8

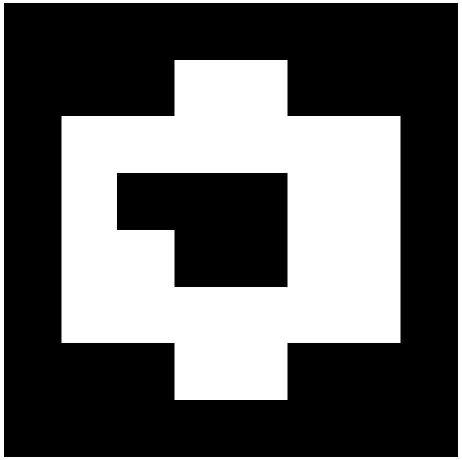
Evolution



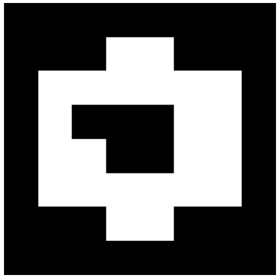
Evolution

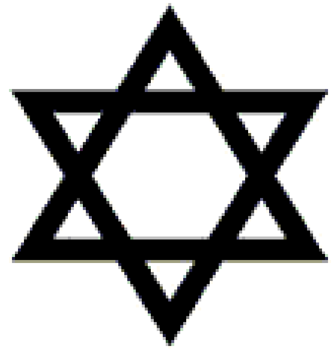


20 generation of evaluation according to 8*8 resized image

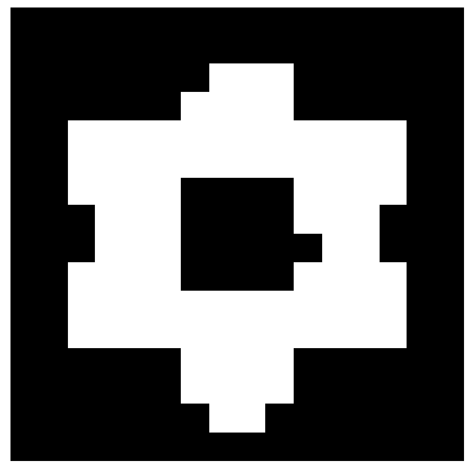


Evolution

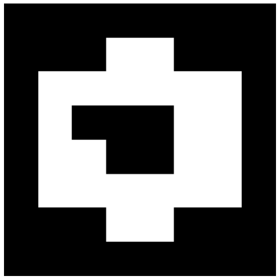


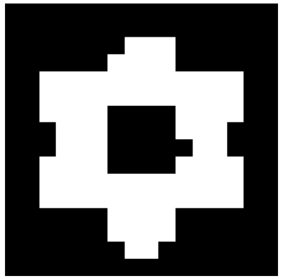


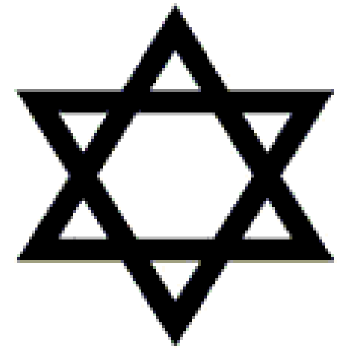
40 generation of evaluation according to 16*16 resized image



Evolution

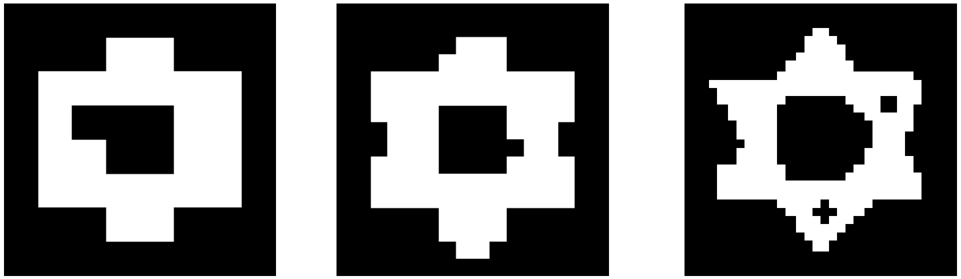


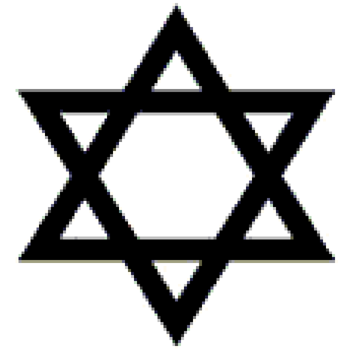




80 generation of evaluation according to 32*32 resized image

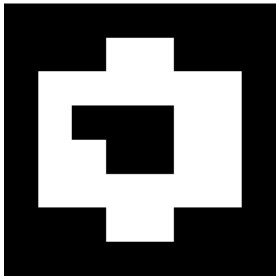
Evolution

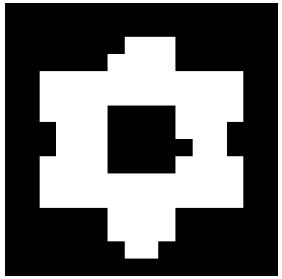


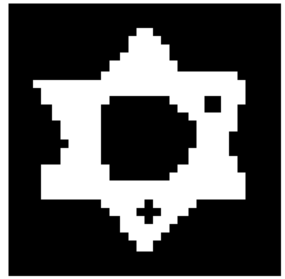


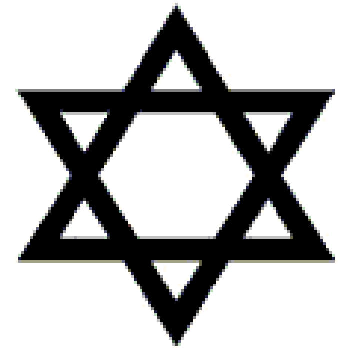
100 generation of evaluation according to 64*64 resized image

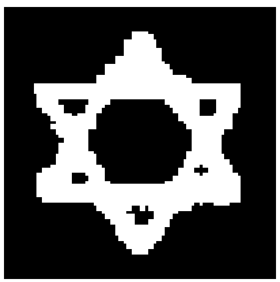
Evolution



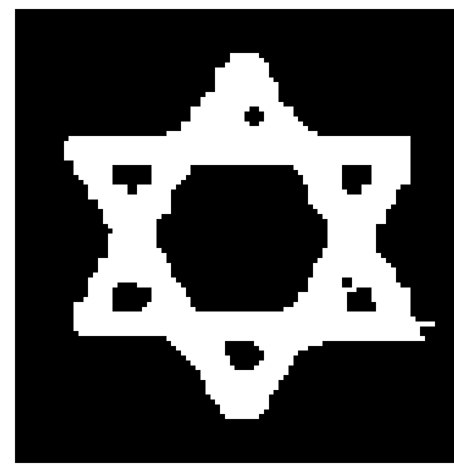




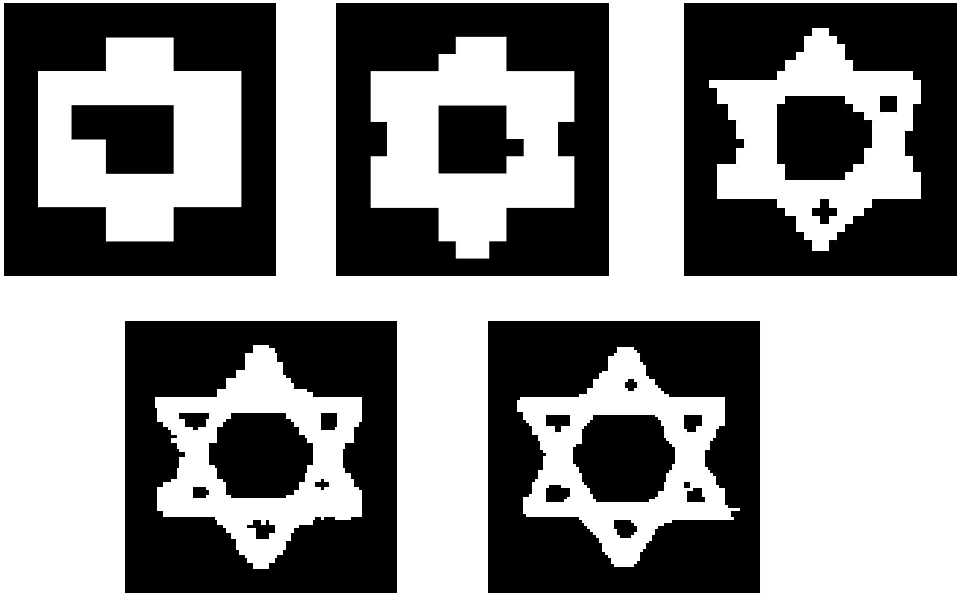


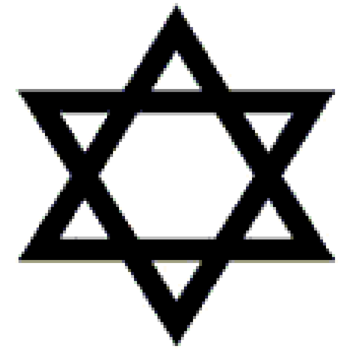


160 generation of evaluation according to original image



Evolution





Selection

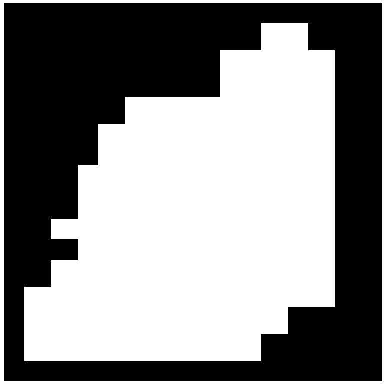
- The best 10% individuals join to the next generation as they are.
- For the last 90%:
  - Randomly choose 4 individuals.
  - The best one chosen as parent A.
  - In the same way parent B is chosen.
  - The offspring of A and B, be a member of the next generation.

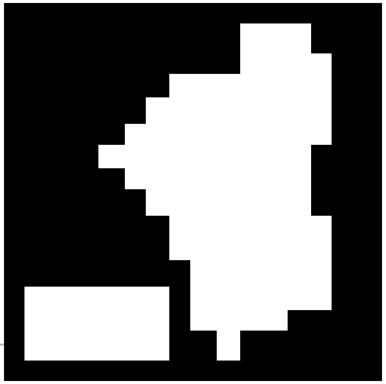
Merge

- Randomly choose pivot
- Randomly choose axis
- With some probability mutate the result

Merge

- Randomly choose pivot
- Randomly choose axis
- With some probability mutate the result





Merge

- Randomly choose pivot
- Randomly choose axis
- With some probability mutate the result

Pivot = 54,
y axis

Mutation

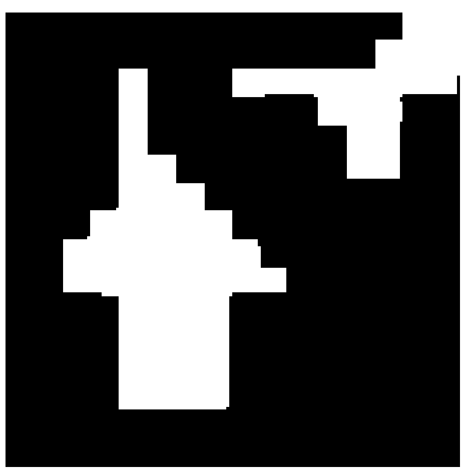
- Method 1
  - Flip random index
- Method 2
  - Add circle
  - Add rectangle
  - Smooth
  - Segment expansion

Mutation

- Method 1
  - Flip random index
- Method 2
  - Add circle
  - Add rectangle
  - Smooth
  - Segment expansion

Mutation

- Method 1
  - Flip random index
- Method 2
  - Add circle
  - Add rectangle
  - Smooth
  - Segment expansion

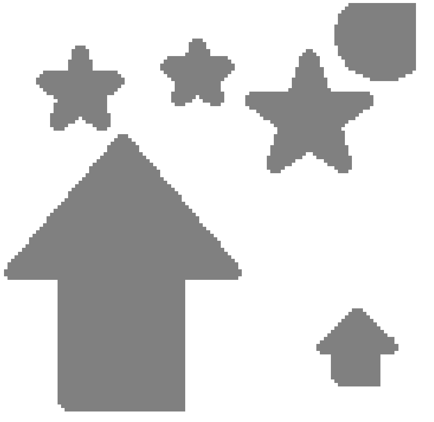


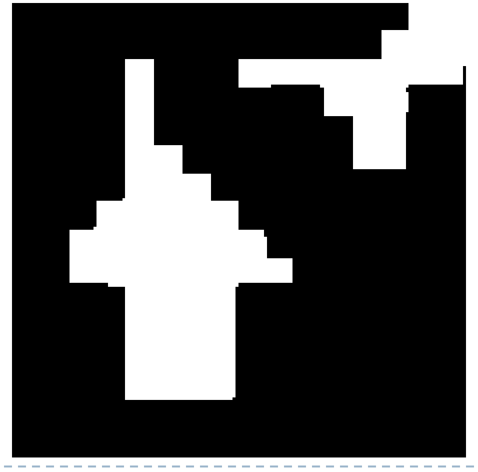
Mutation

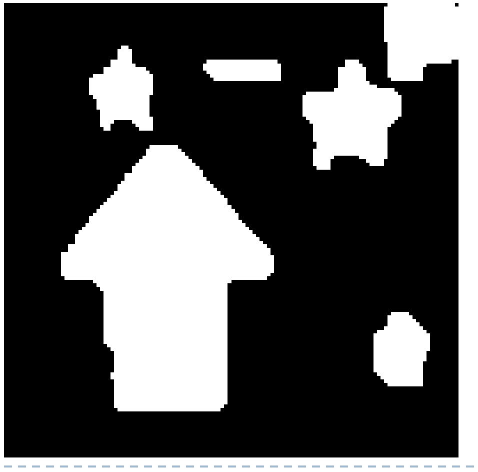
- Method 1
  - Flip random index
- Method 2
  - Add circle
  - Add rectangle
  - Smooth
  - Segment expansion

Mutation

- Method 1
  - Flip random index
- Method 2
  - Add circle
  - Add rectangle
  - Smooth
  - Segment expansion







Fitness Function

- I=

2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4

Fitness Function

- I=

2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	4	z	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4

Fitness Function

- I=

2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4
2	2	2	4	4	4	4	4

Fitness Function

- A=

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1

Fitness Function

- Low variance in each segment.
- High derivative at boundary points

Fitness

- A=

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1

At boundary point by x axis, $\frac{\partial I}{\partial x}$ should receive high values

Fitness

- I_x=

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0

Fitness

- A=

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1

At boundary point by y axis, $\frac{\partial I}{\partial y}$ should receive high values

Fitness Function

- I_y=

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	-1	-1	-1	-1	-1
0	0	0	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fitness

$$A_0 = \{(i,j) \in (1,\dots,n)^2 : A(i,j) = 0\}, \quad V_0 = \mathrm{var}(\{I(i,j) : (i,j) \in A_0\}),$$

$$A_1 = \{(i,j) \in (1,\dots,n)^2 : A(i,j) = 1\}, V_1 = \mathrm{var}(\{I(i,j) : (i,j) \in A_1\}),$$

$$I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}$$

$$\Phi_x = \{(i,j) : A(i,j) \neq A(i+1,j)\}, \ \psi_x = \Sigma_{(i,j)\in\Phi_x} \frac{1}{\alpha + |I_x(i,j) + I_x(i+1,j)|},$$

$$\Phi_y = \{(i,j) : A(i,j) \neq A(i,j+1)\}, \psi_y = \Sigma_{(i,j)\in\Phi_y} \frac{1}{\alpha + |I_x(i,j) + I_x(i,j+1)|}$$

$$a, b, c, d, \alpha > 0$$

$$R(A) = aV_0 + bV_1 + c\psi_x + d\psi_y$$

Image with noise

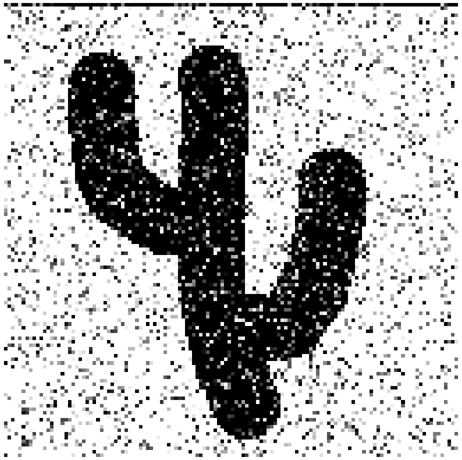
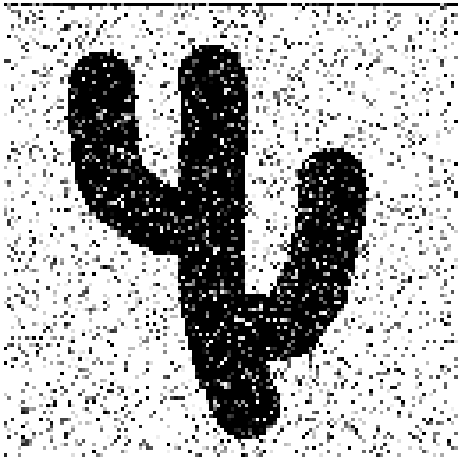


Image with noise



Running time

- For n*n image:
- *Creating the initial population.*
- *For every generation;*
  - *Ranking all the population*
  - *for every individual;*
    - *Pick parents*
    - *Merge*
    - *Mutate*
- Total running time: -

$$O(p \cdot n^2) + O(g \cdot (p \cdot n^2 + p(n^2))) = O(g \cdot p \cdot n^2) = O(n^2)$$

Running time

- For n*n image:
- *Creating the initial population.*
- *For every generation;*
  - *Ranking all the population*
  - *for every individual;*
    - *Pick parents*
    - *Merge*
    - *Mutate*
- Total running time:

$$O(p \cdot n^2) + O(g \cdot (p \cdot n^2 + p(n^2))) = O(g \cdot p \cdot n^2) = O(n^2)$$

$$p, g \geq O(n)$$

Running time

- For n*n image:
- *Creating the initial population.*
- *For every generation;*
  - *Ranking all the population*
  - *for every individual;*
    - *Pick parents*
    - *Merge*
    - *Mutate*
- Total running time:

$$O(p \cdot n^2) + O(g \cdot (p \cdot n^2 + p(n^2))) = O(g \cdot p \cdot n^2) \geq O(n^4)$$

$$p, g \geq O(n)$$