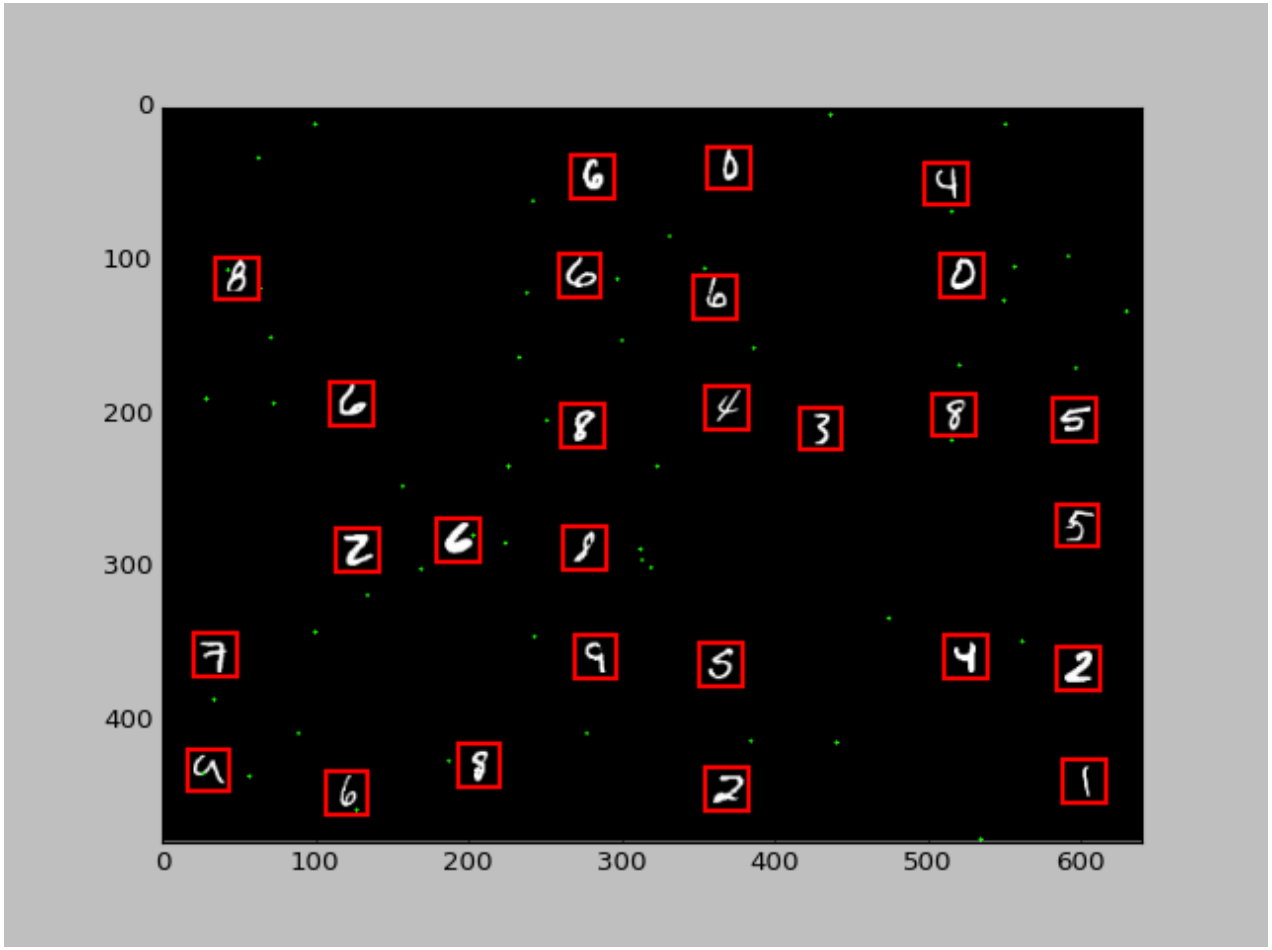
Sabiranje brojeva na slikama uz korišćenje veštačkih neuronskih mreža



Autor: Marko Vještica

Predmet: Soft kompjuting

Ideja

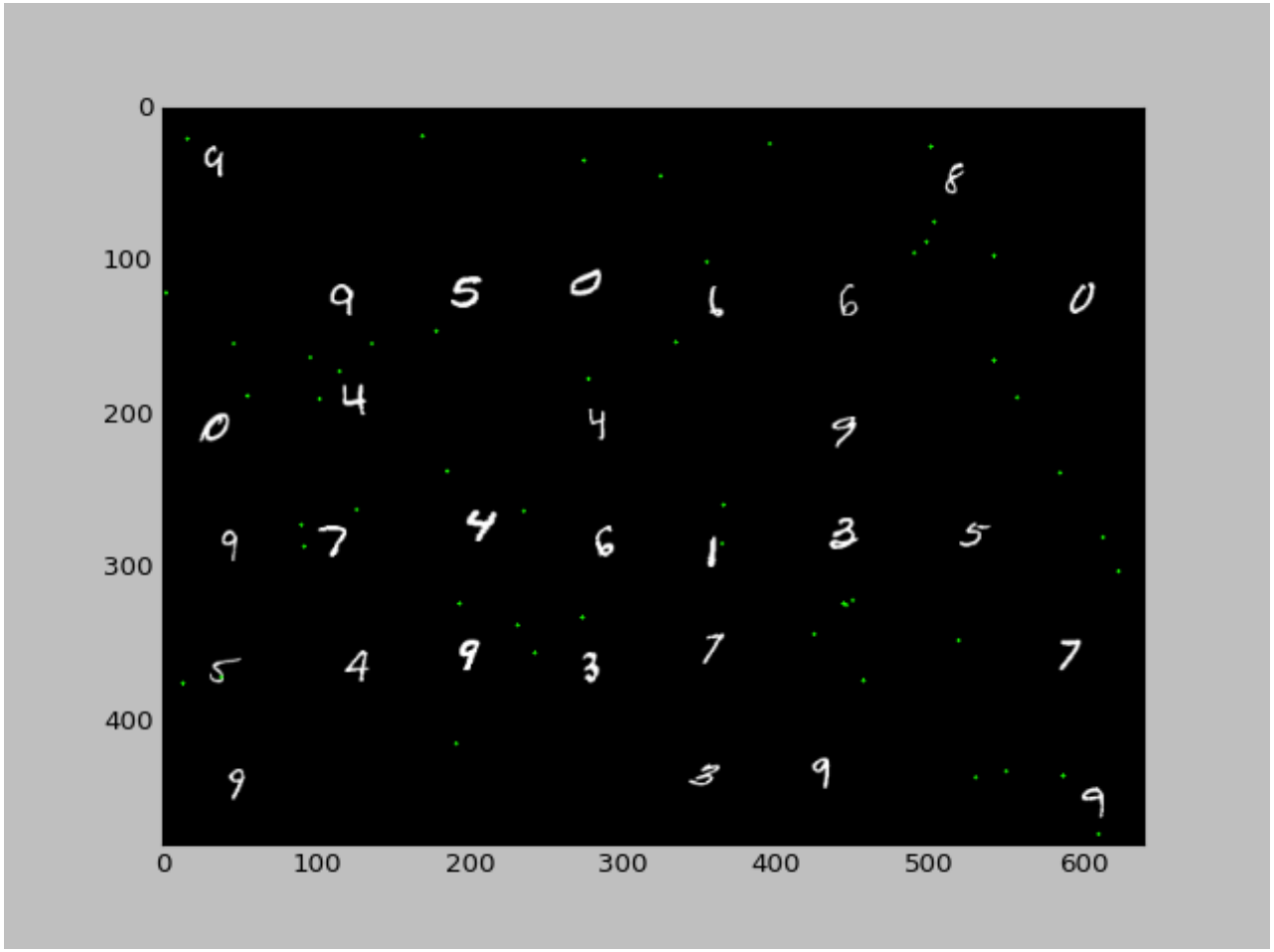
1. Pretražiti ceo prostor slike
   - Potrebno svaki deo slike proveriti
   - Moguće lošije prepoznavanje brojeva na slici
   - Performanse računara
2. Pretražiti odeđene regione slike
   - Potrebno pronaći regione
   - Performanse računara

Razvoj

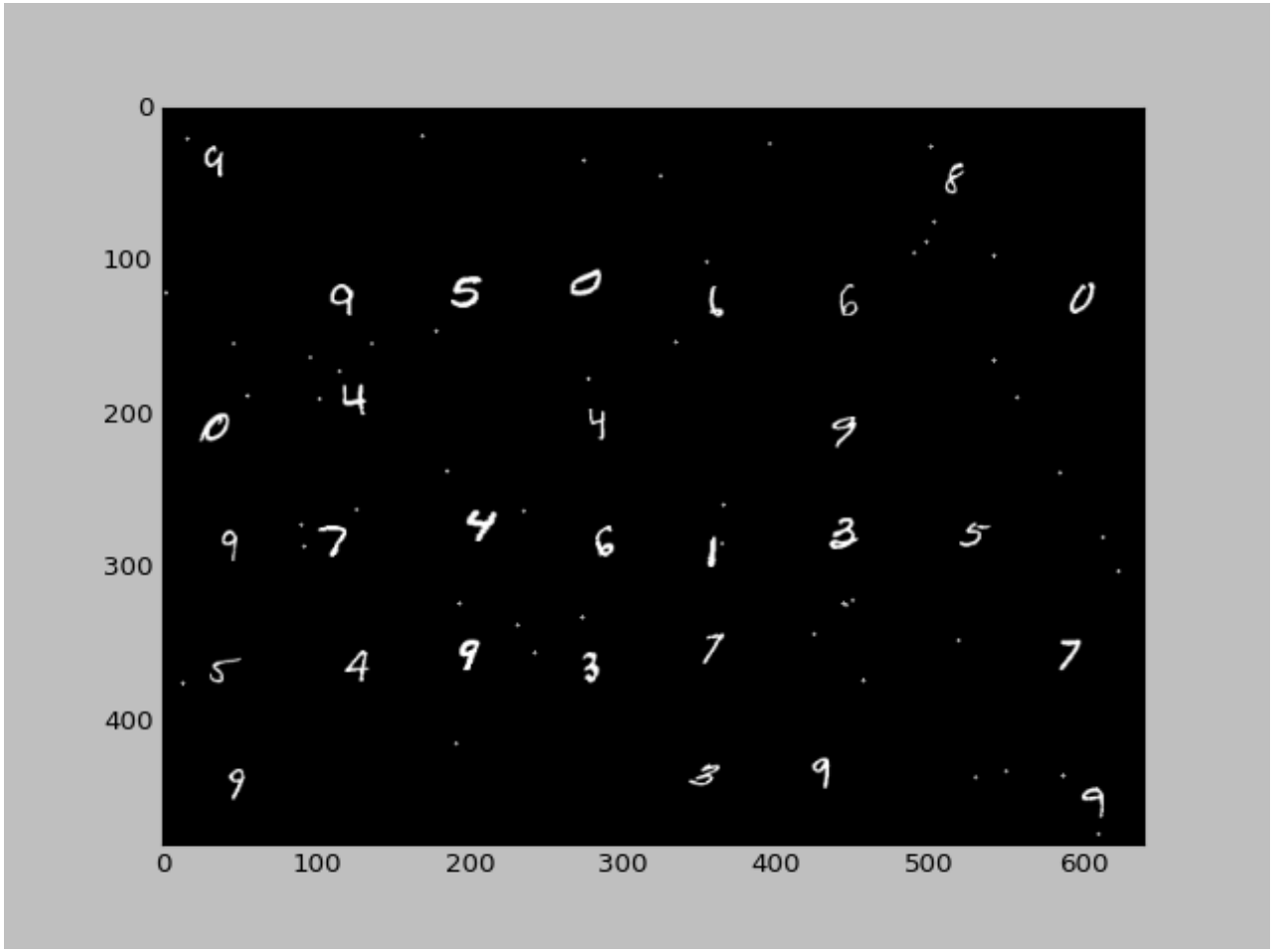
Paralelno se razvija:

- Algoritam za određivanje regiona:
  - Uklanjanje šuma
  - Pronalaženje brojeva
  - Centriranje regiona
- Model neuronske mreže:
  - Obučavajući skup
  - Broj epoha
  - Aktivaciona funkcija
  - Broj slojeva neuronske mreže
  - ...

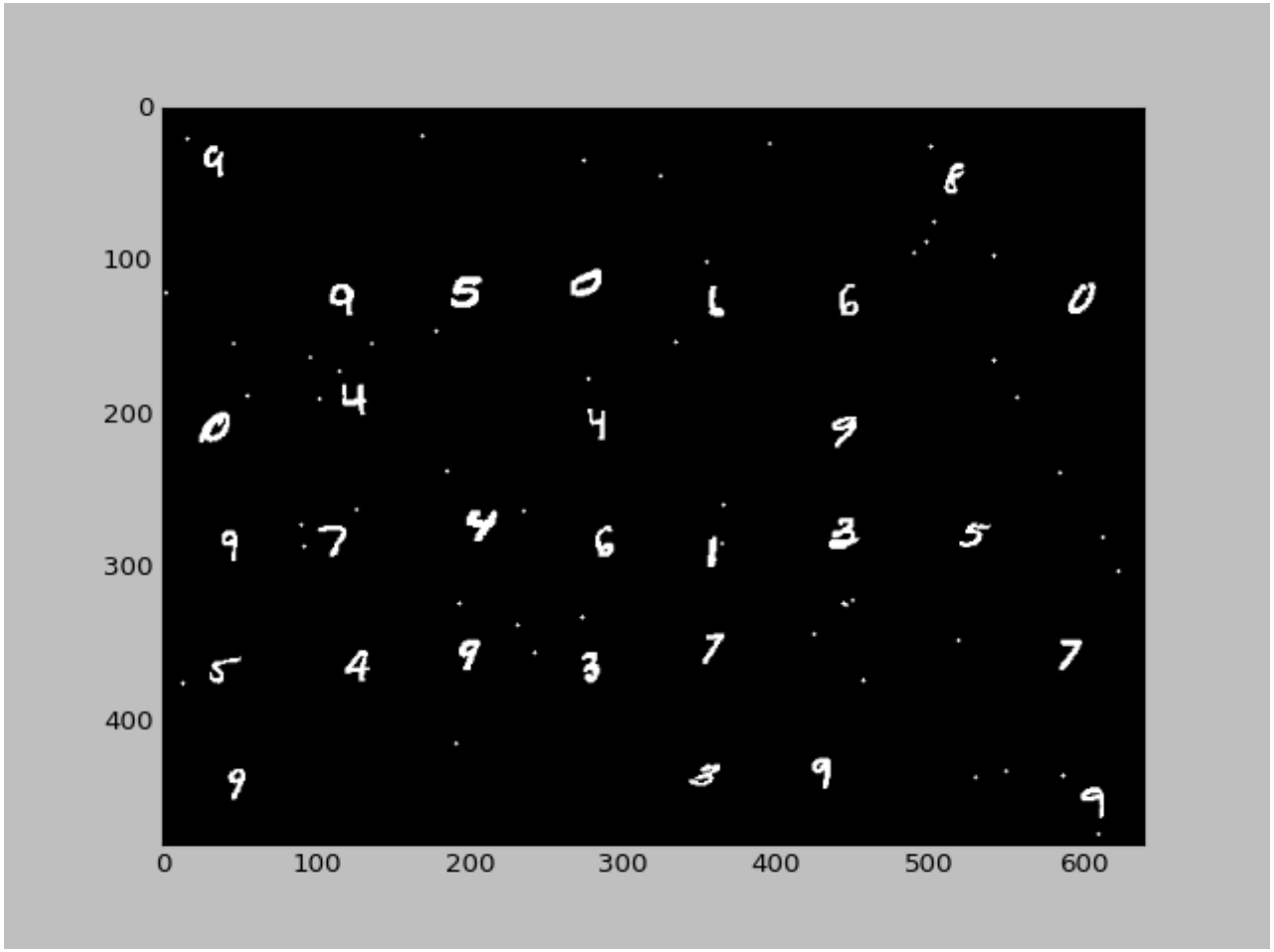
Algoritam za određivanje regiona



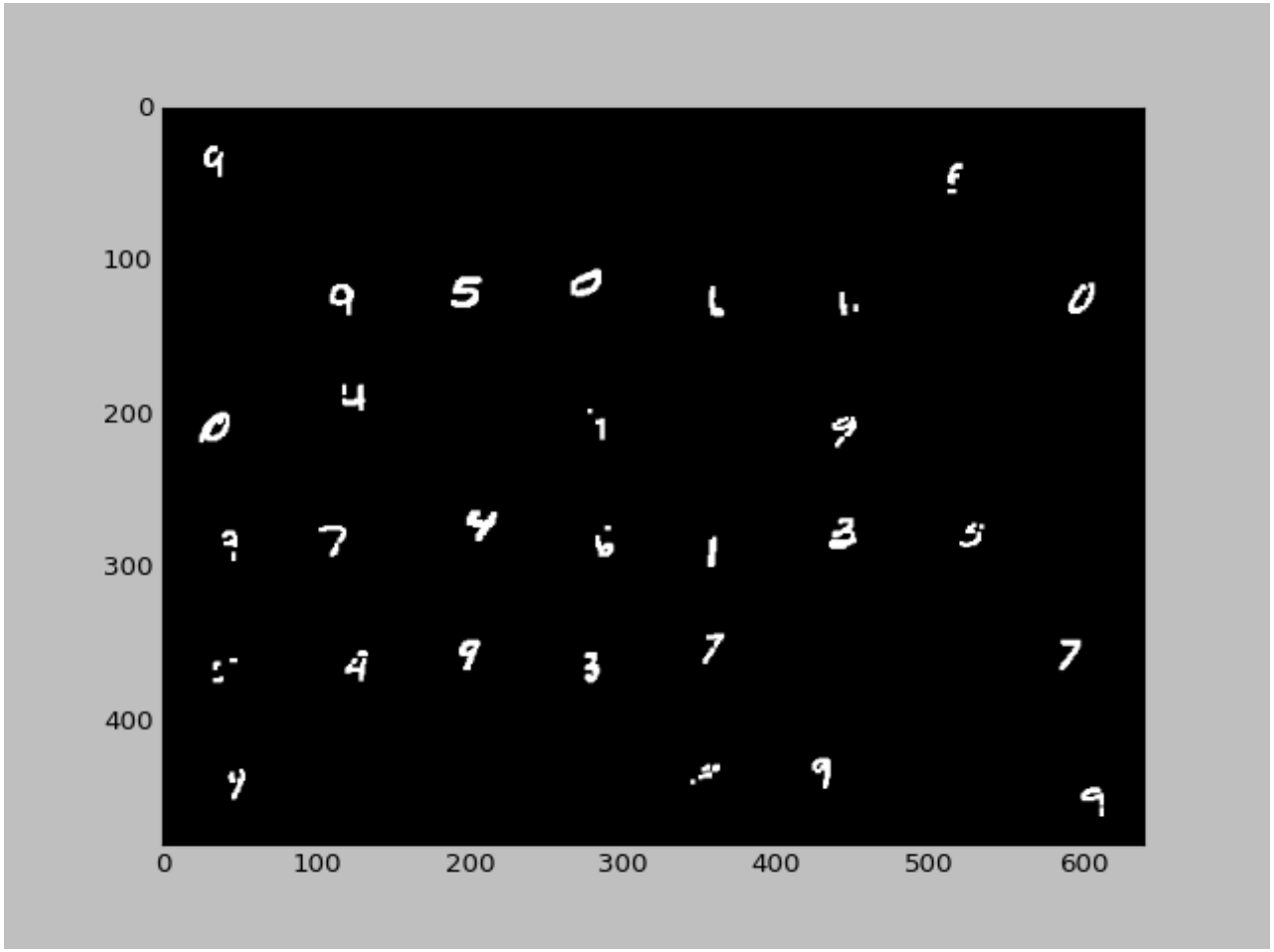
Grayscale



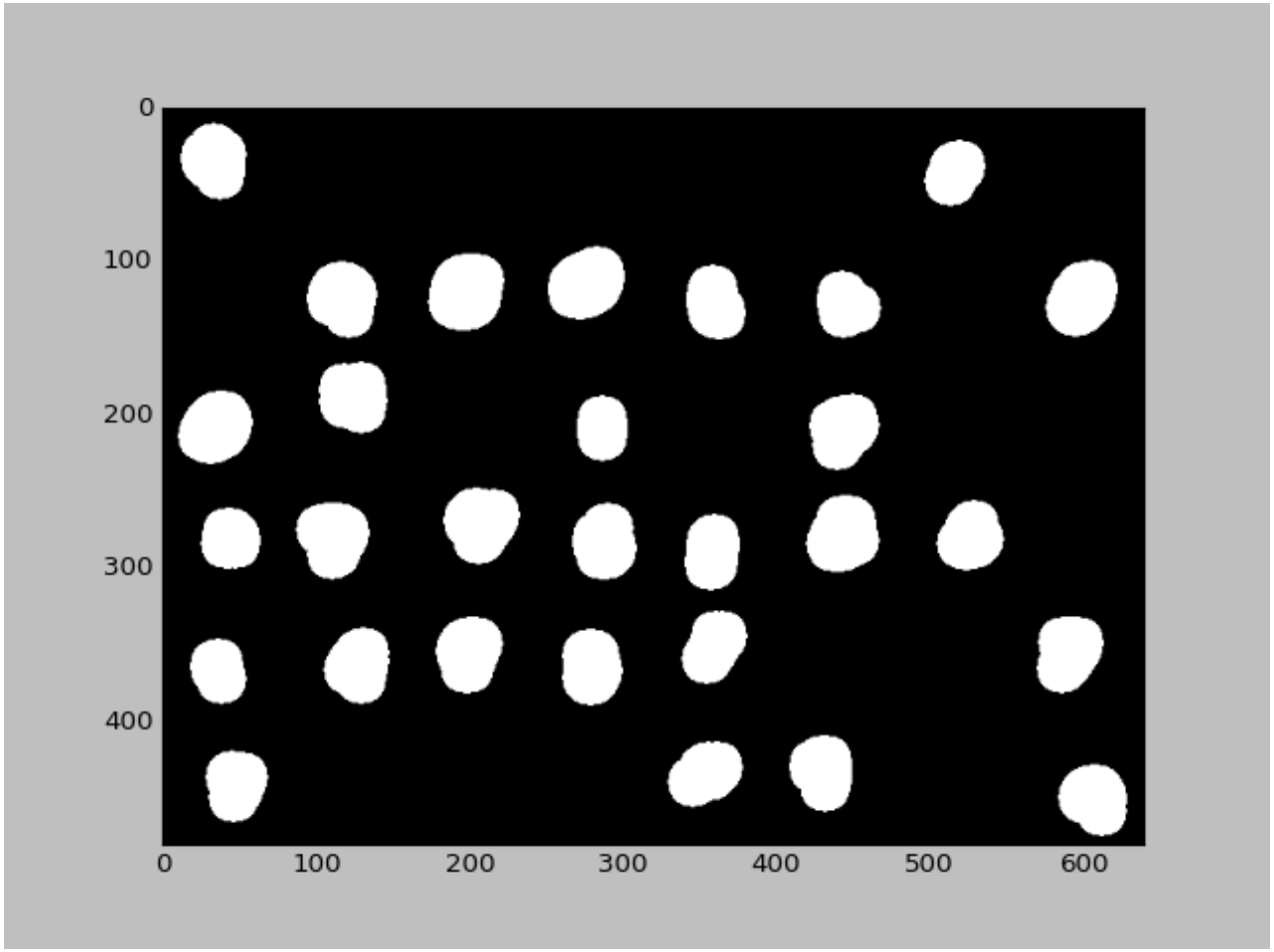
Globalni threshold
img_gray > 0.05



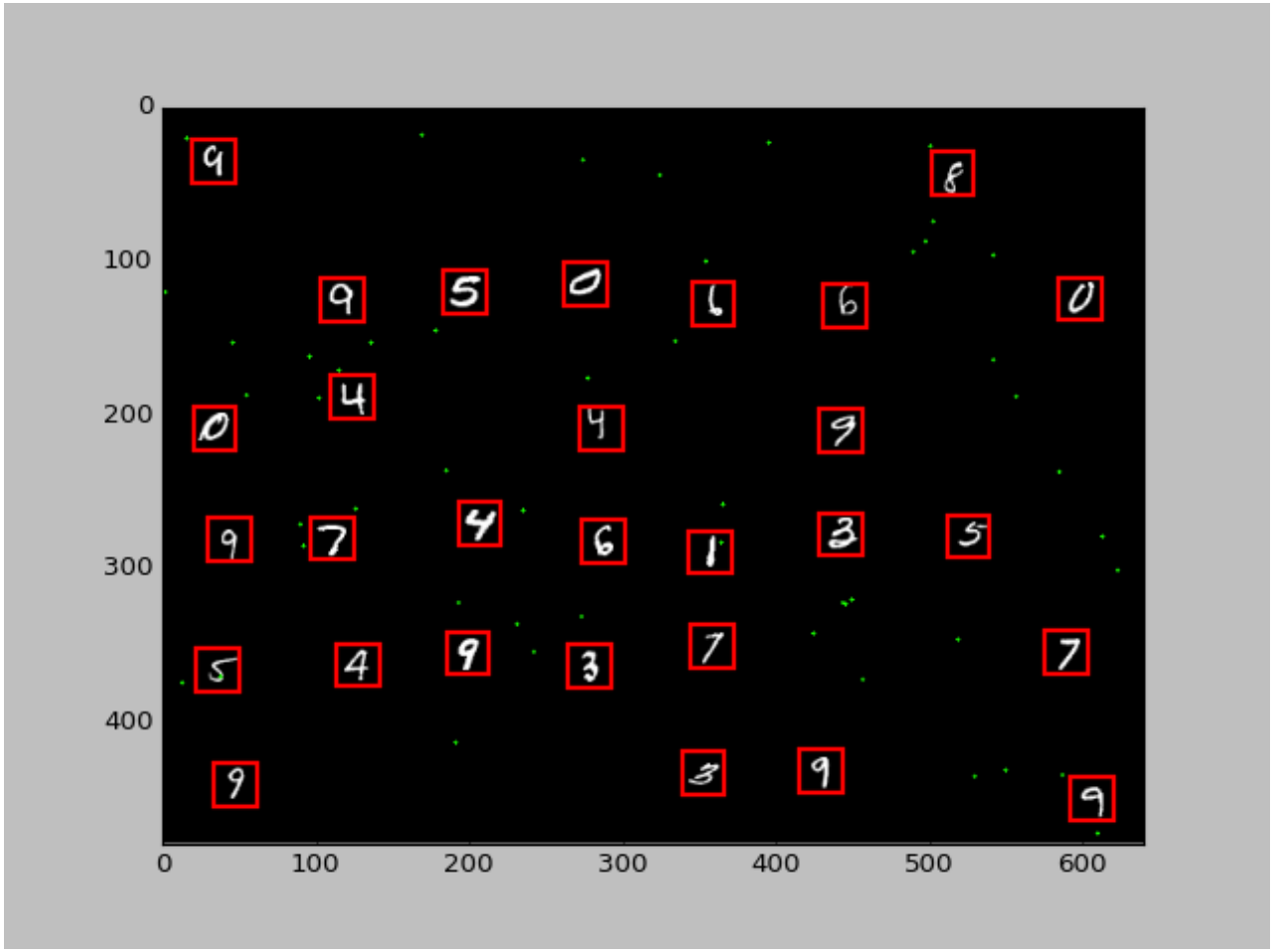
Opening – uništavanje tačaka square(3)



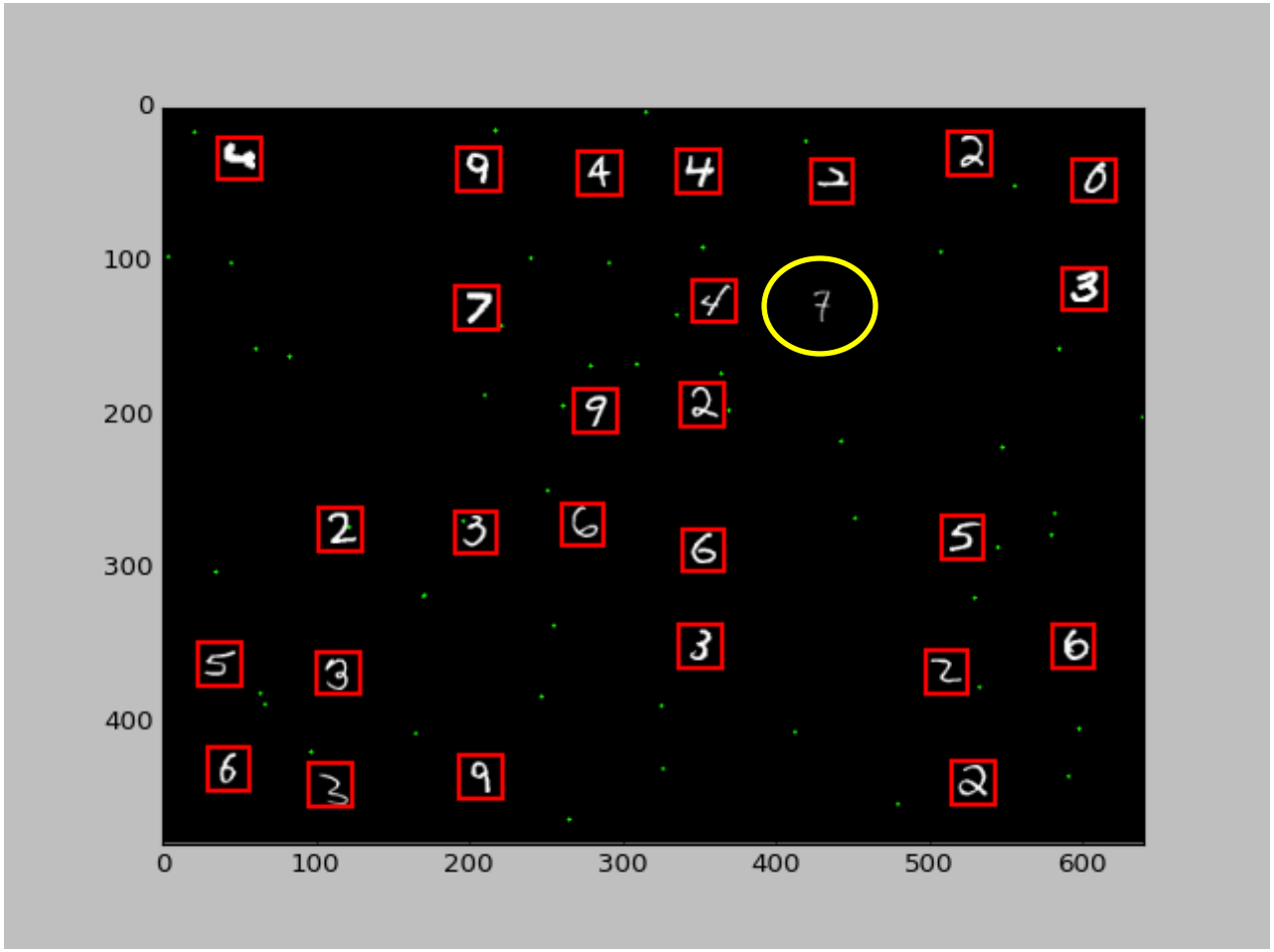
Dilation – lakše određivanje regiona disk(15)



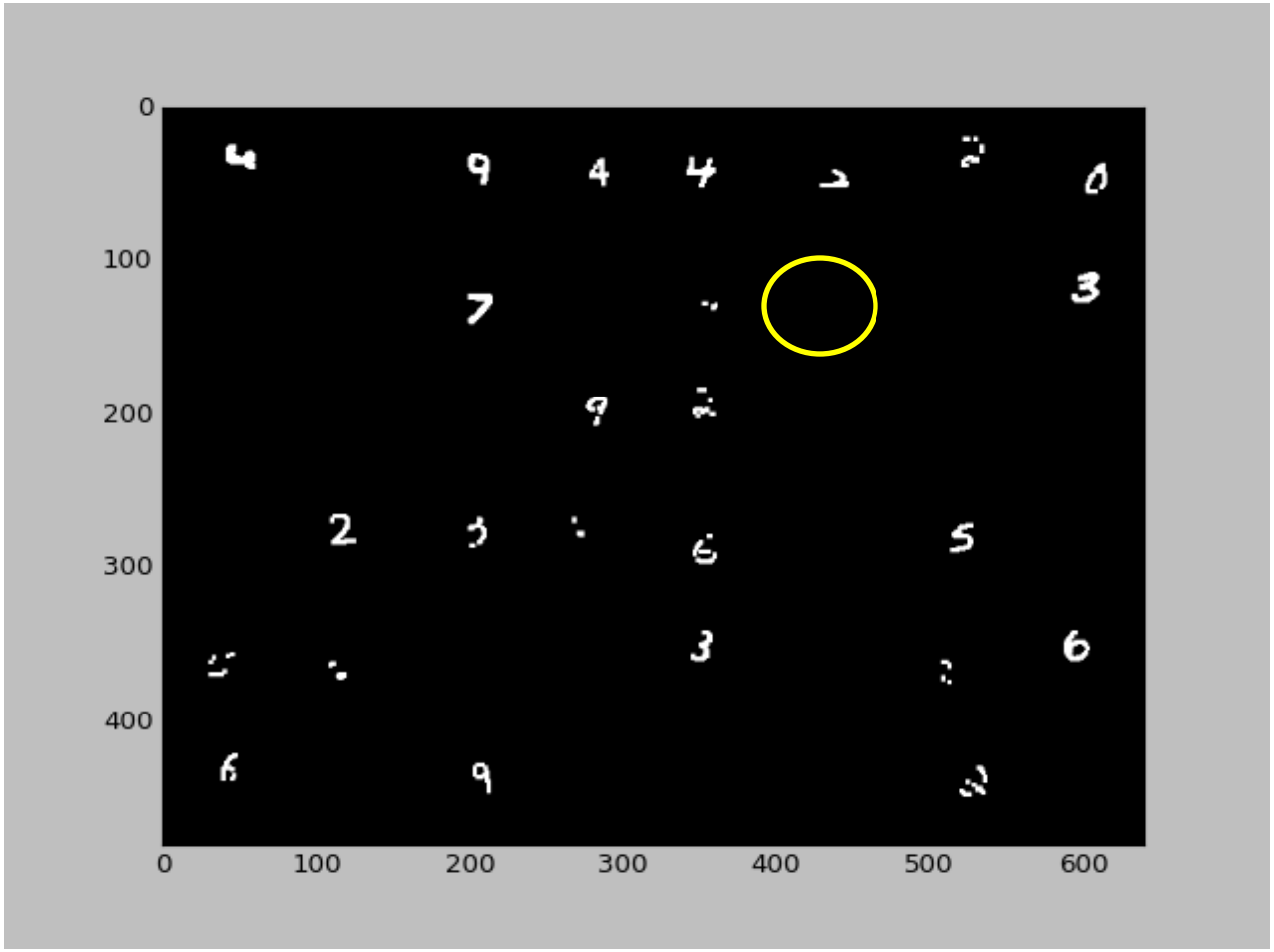
Tačnost: 23%
Da li dobro detektuje sve brojeve?



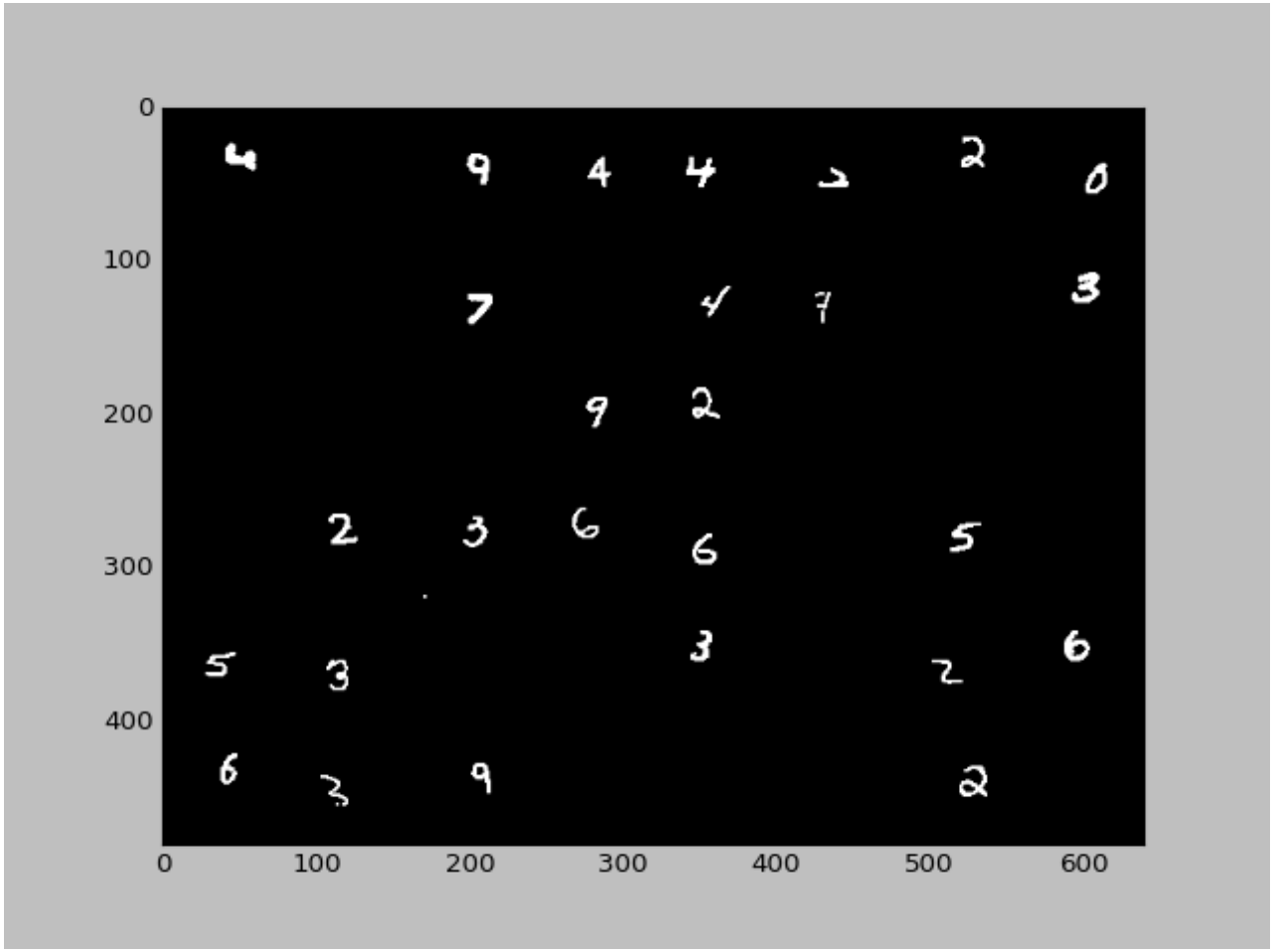
Problem: uklanjanje brojeva



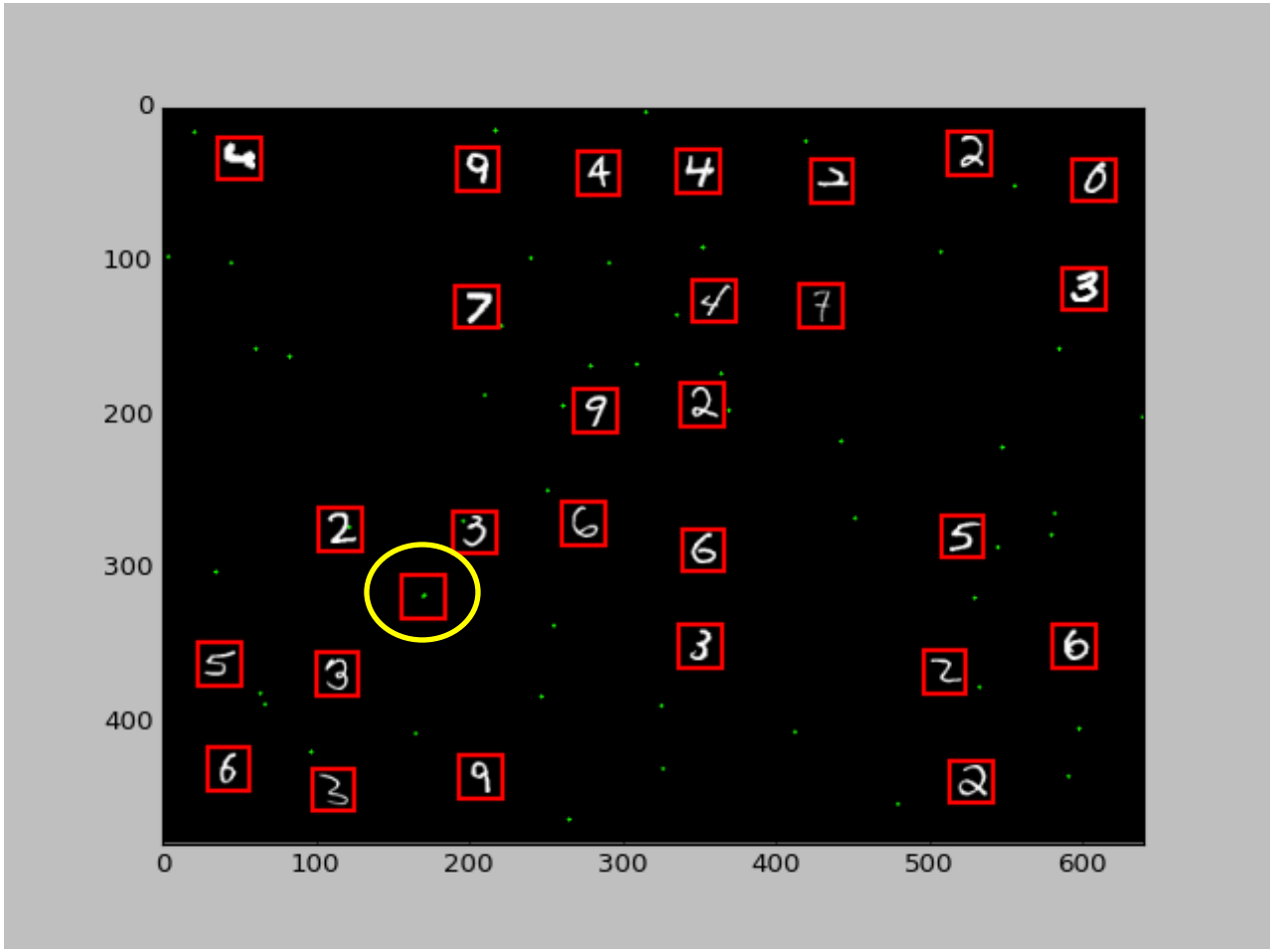
Razlog: Opening square(3)



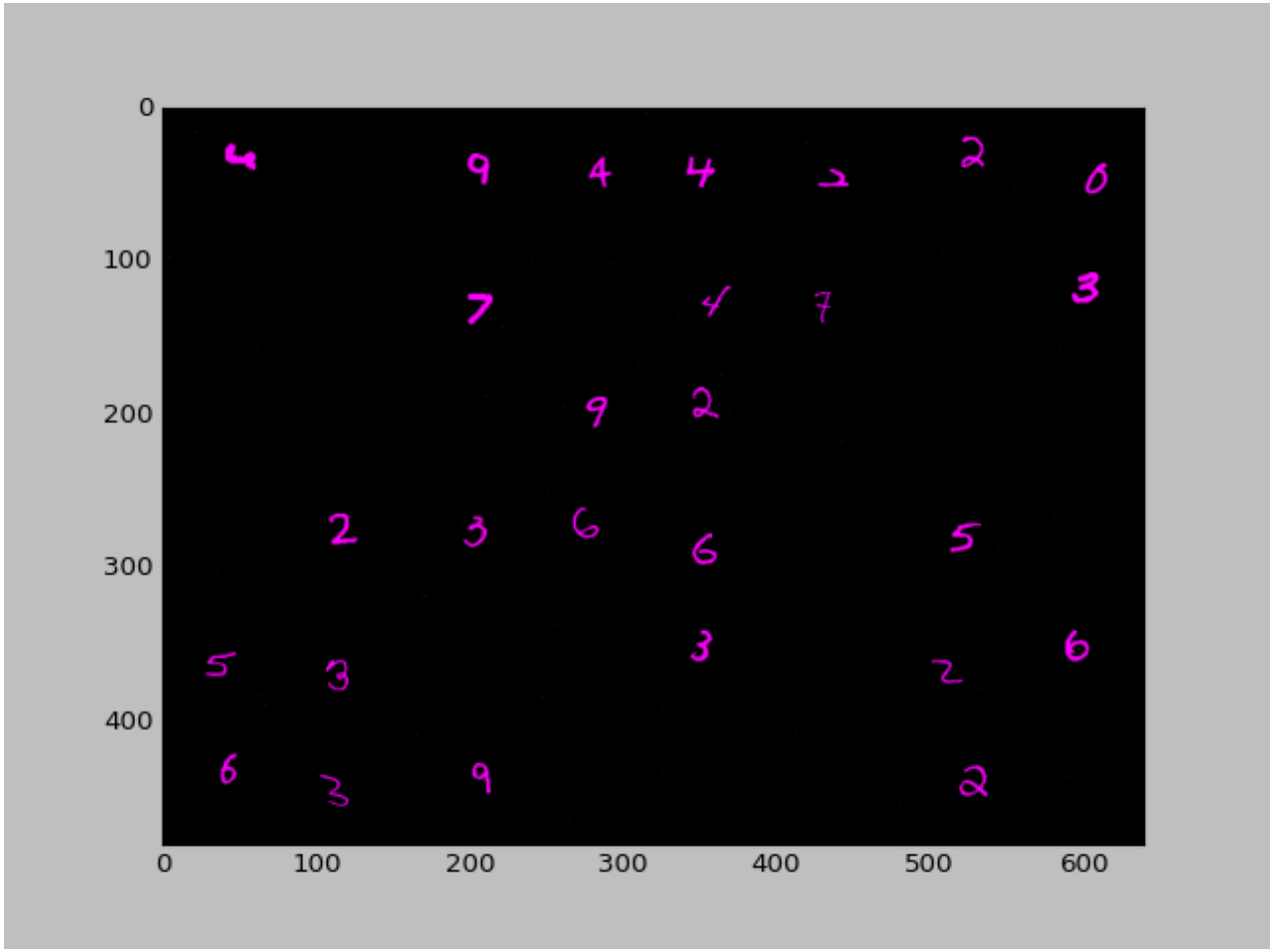
Opening square(2)



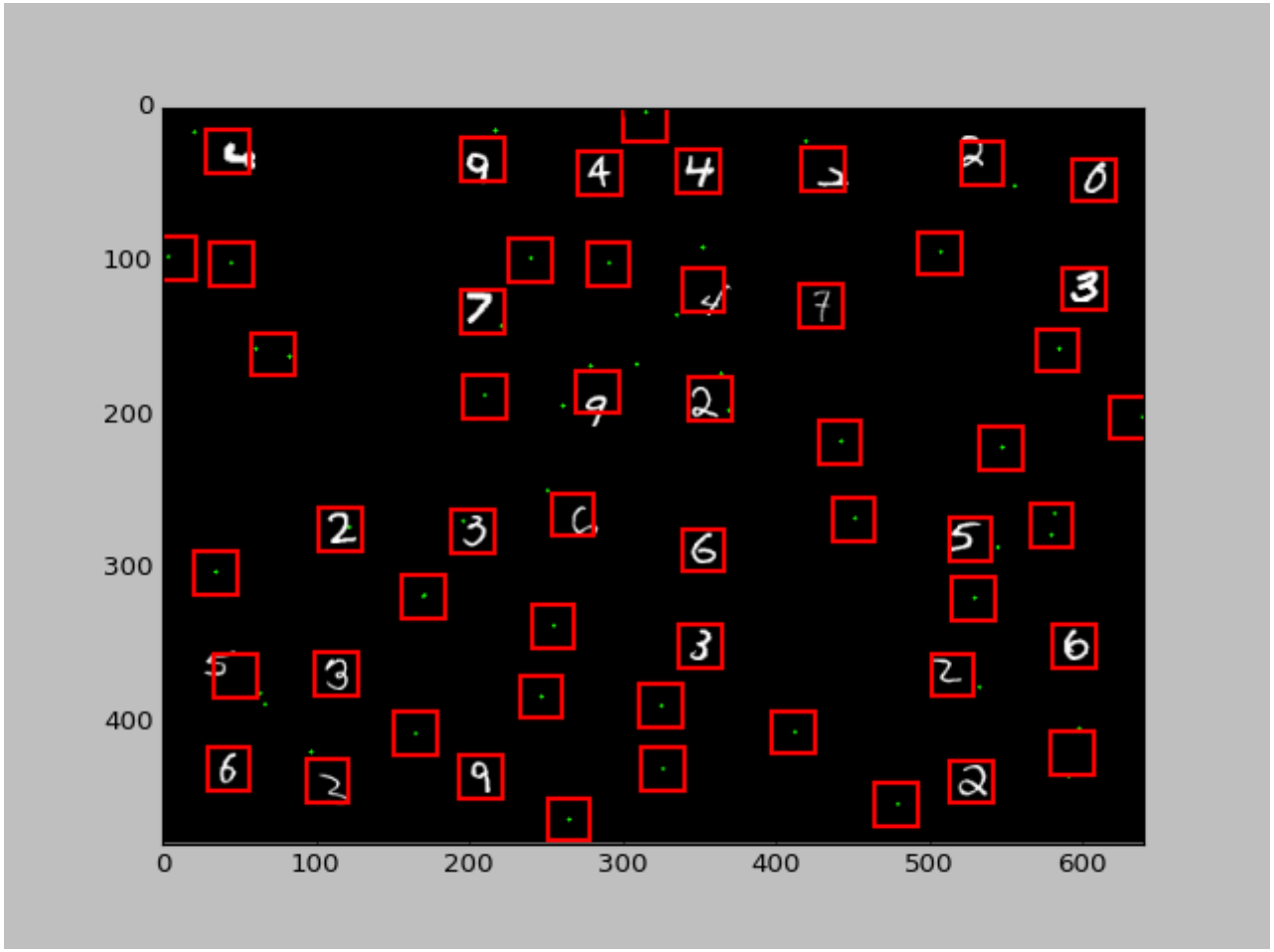
Rešeno: uklanjanje brojeva
Problem: ostaju tačke



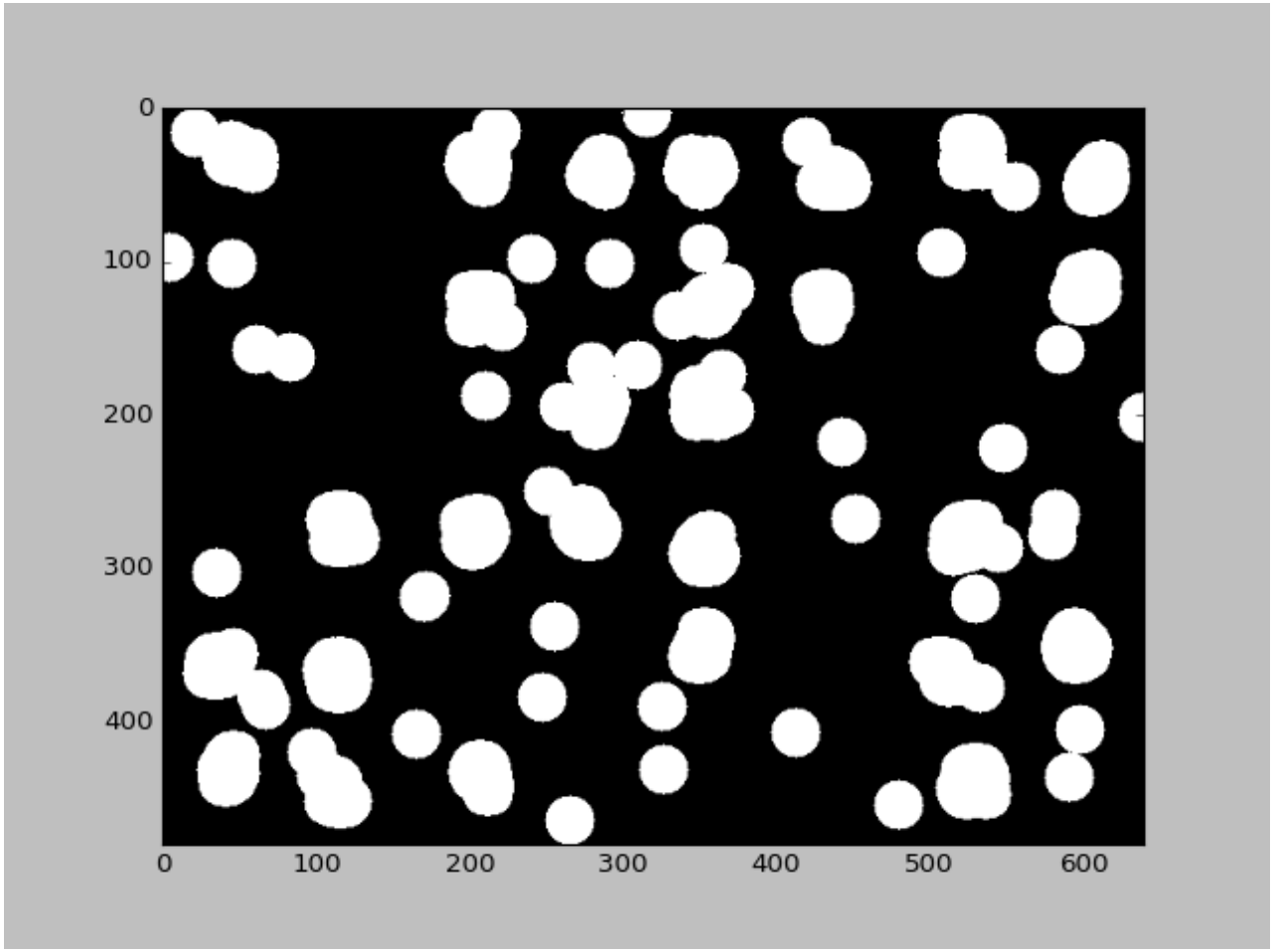
Uklanjanje zelene boje: img[:, :, 1] = 0
Možda ne treba opening



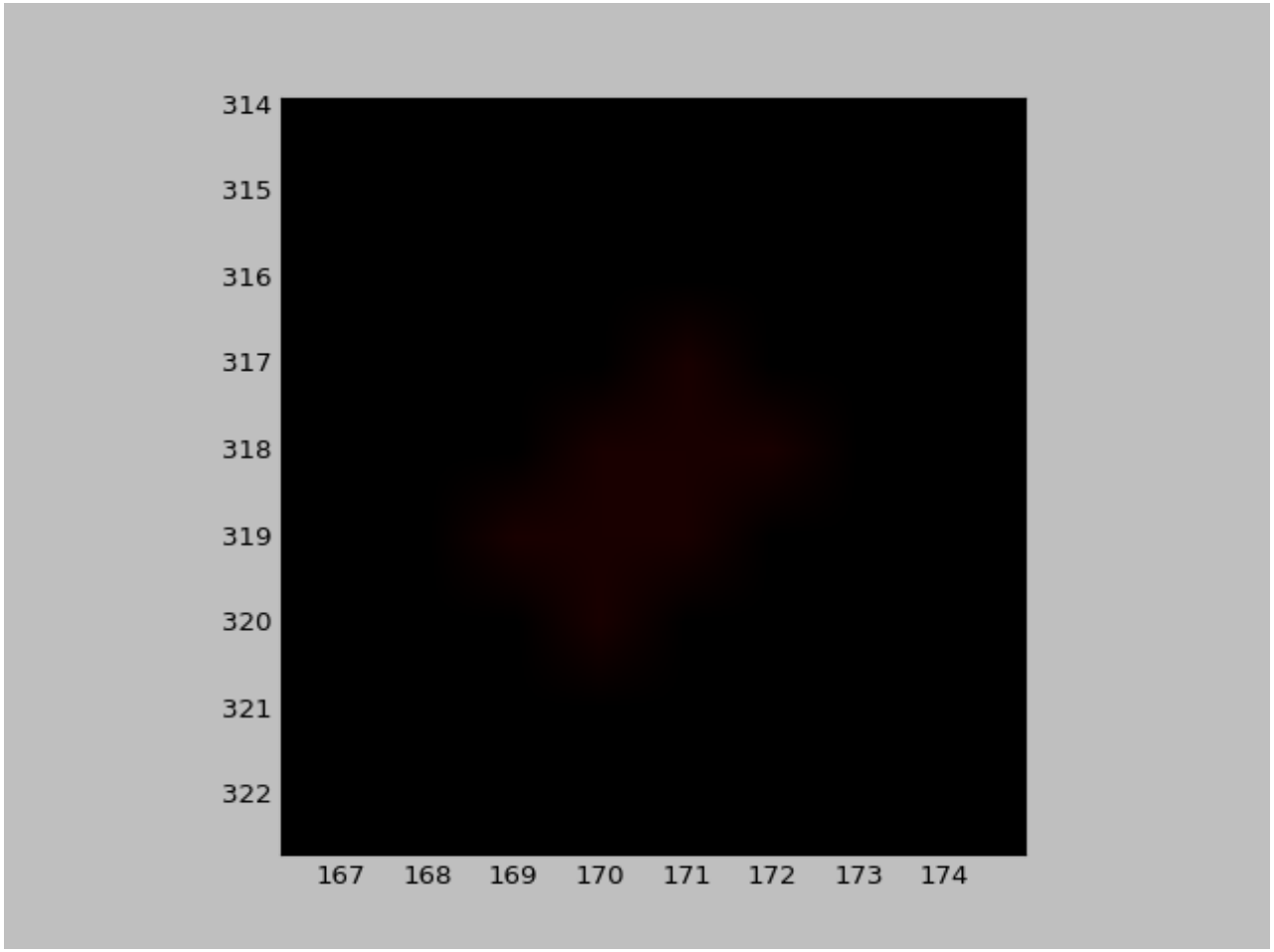
Problem: ostaju tačke



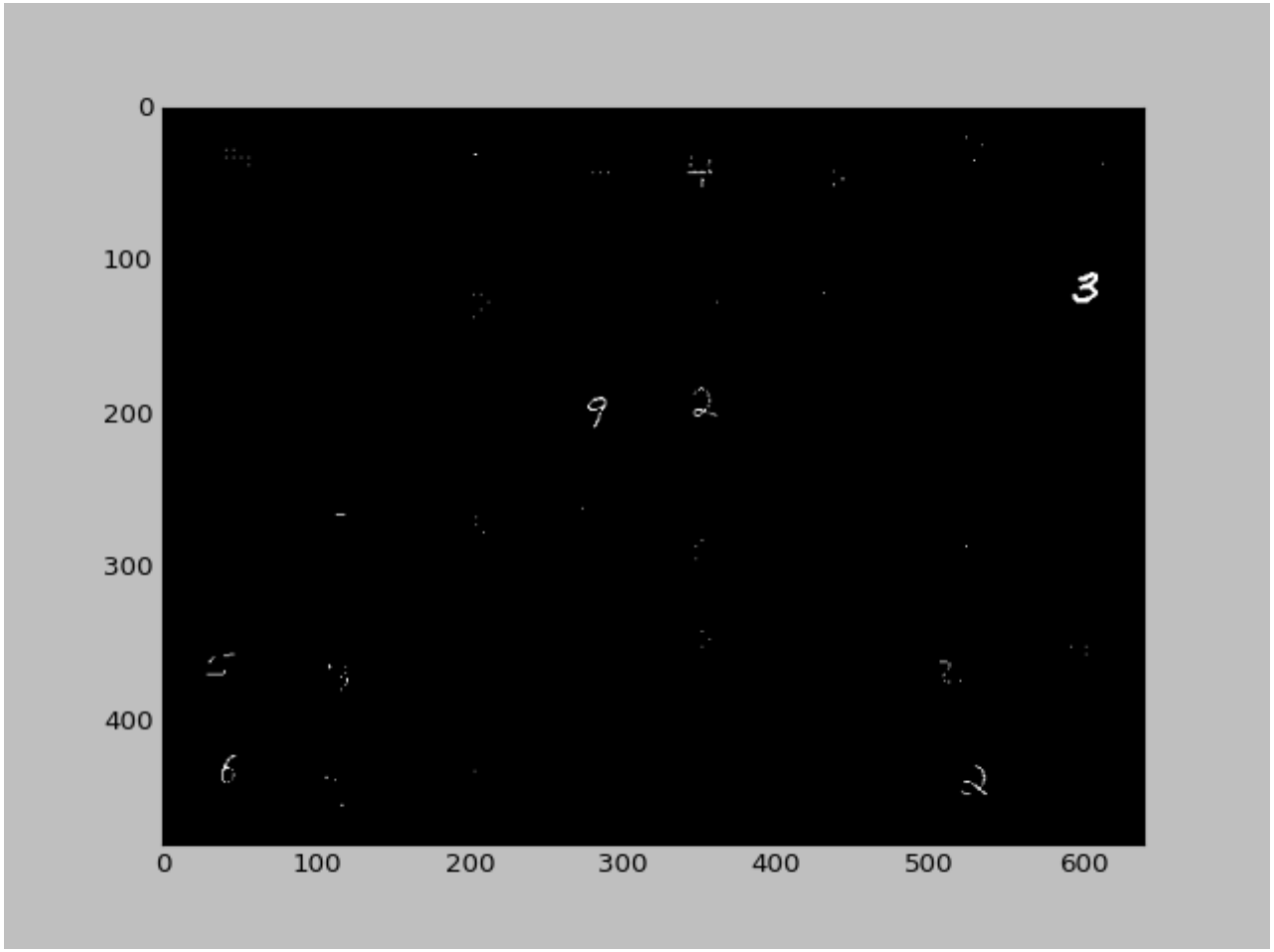
Problem: ostaju tačke



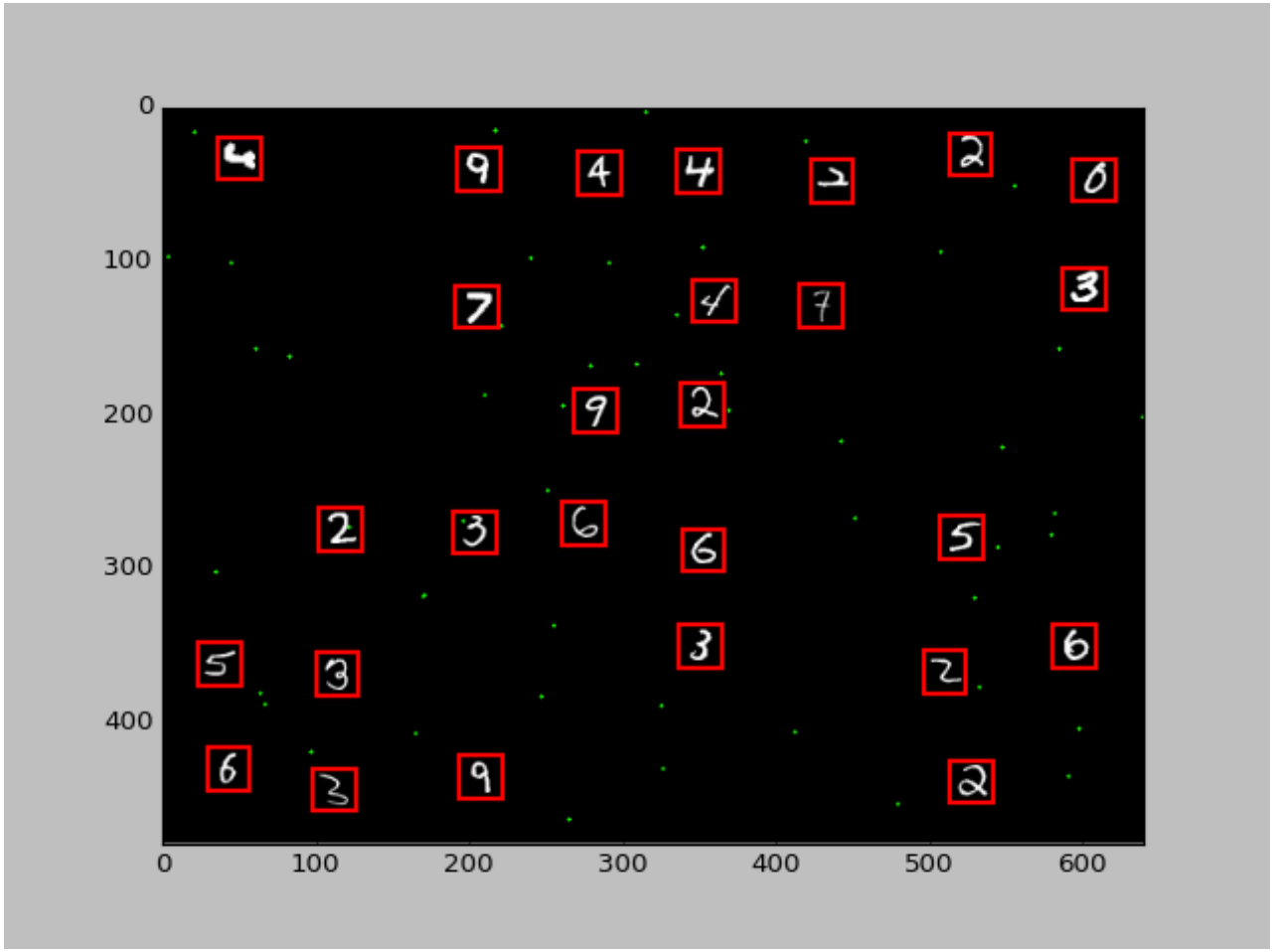
Ostaju nijanse plave i crvene



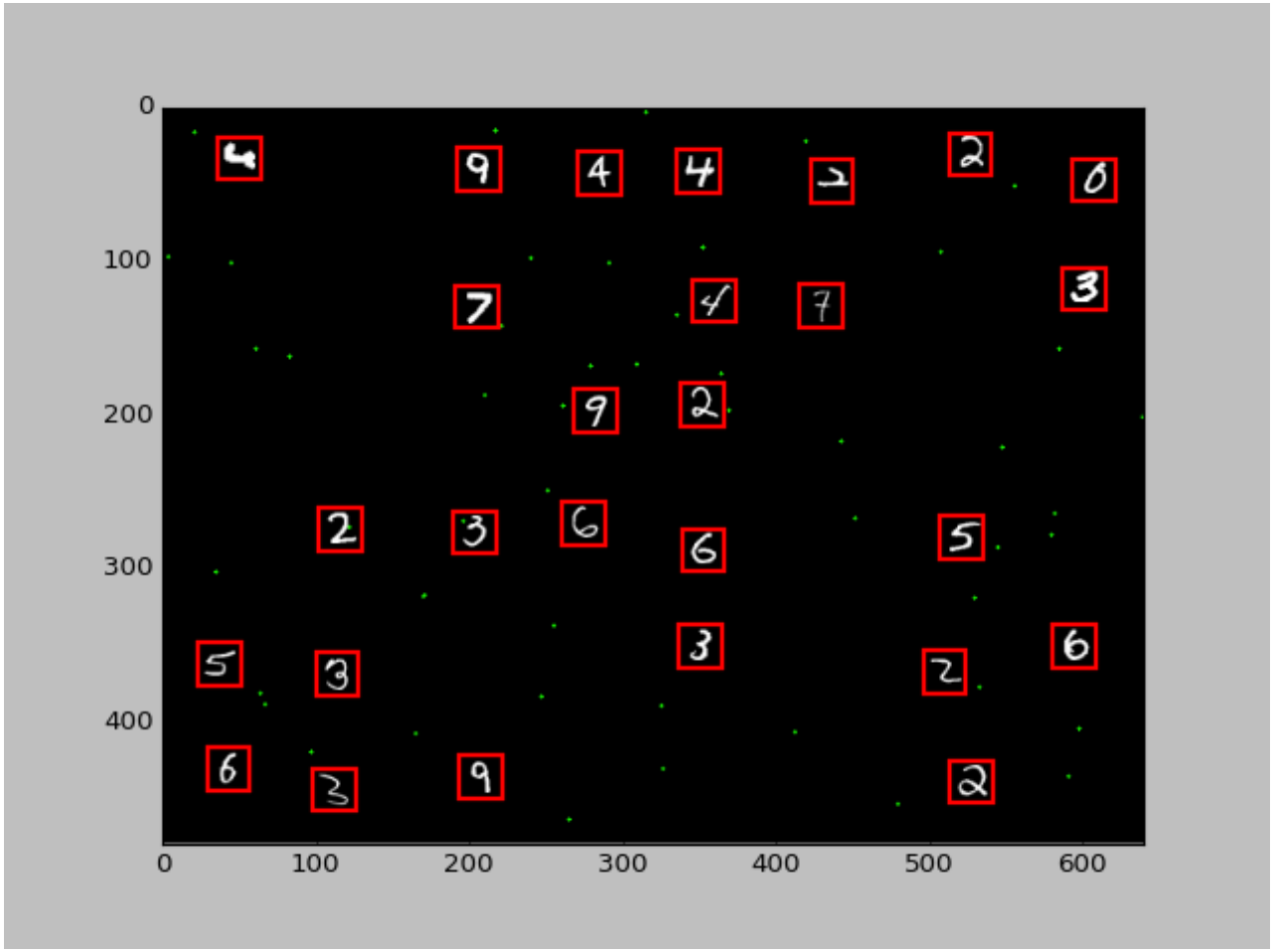
Rešenje: povećanje threshold-a
Problem: uništava brojeve



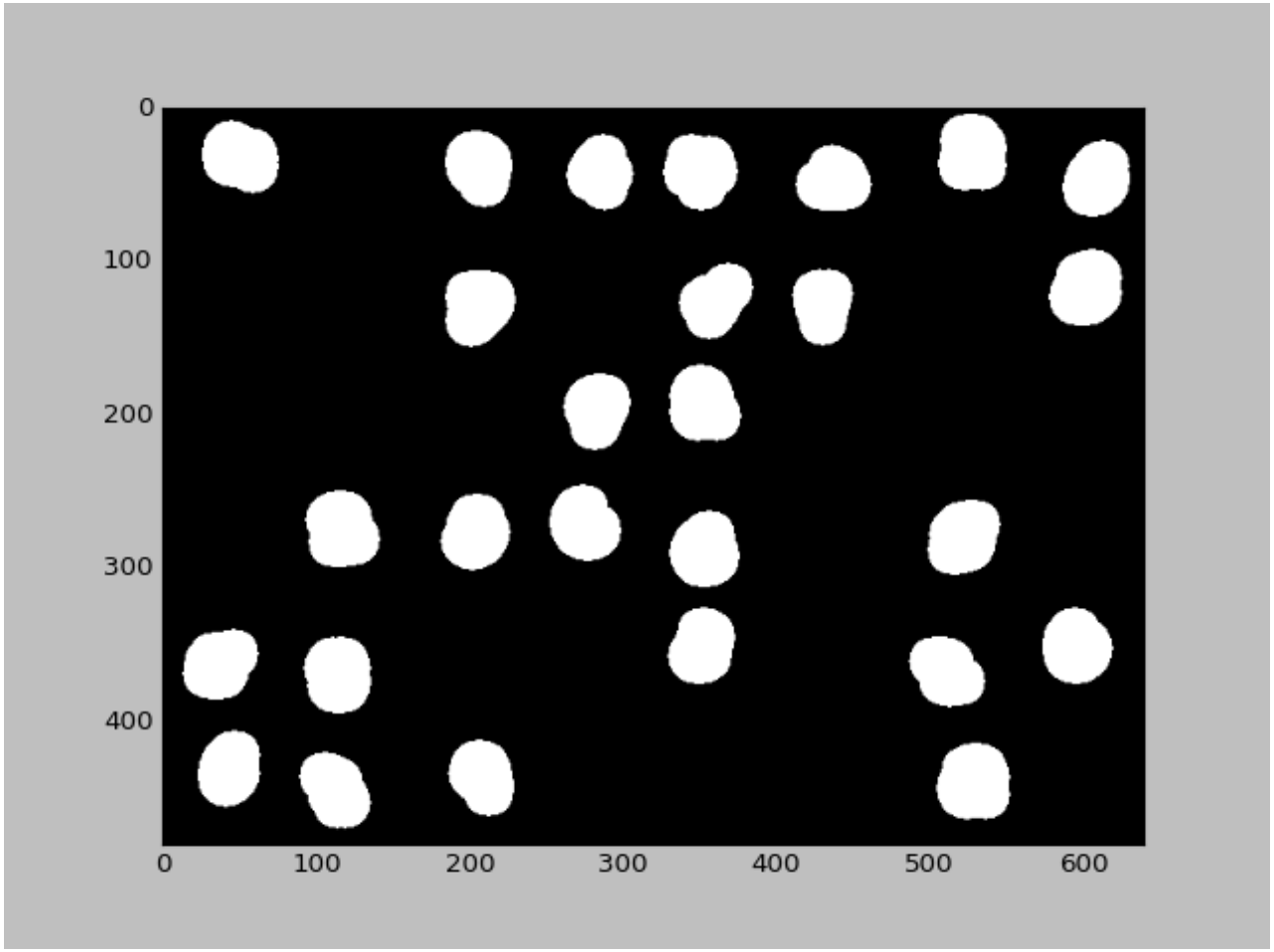
Rešenje 2: opening square(2)



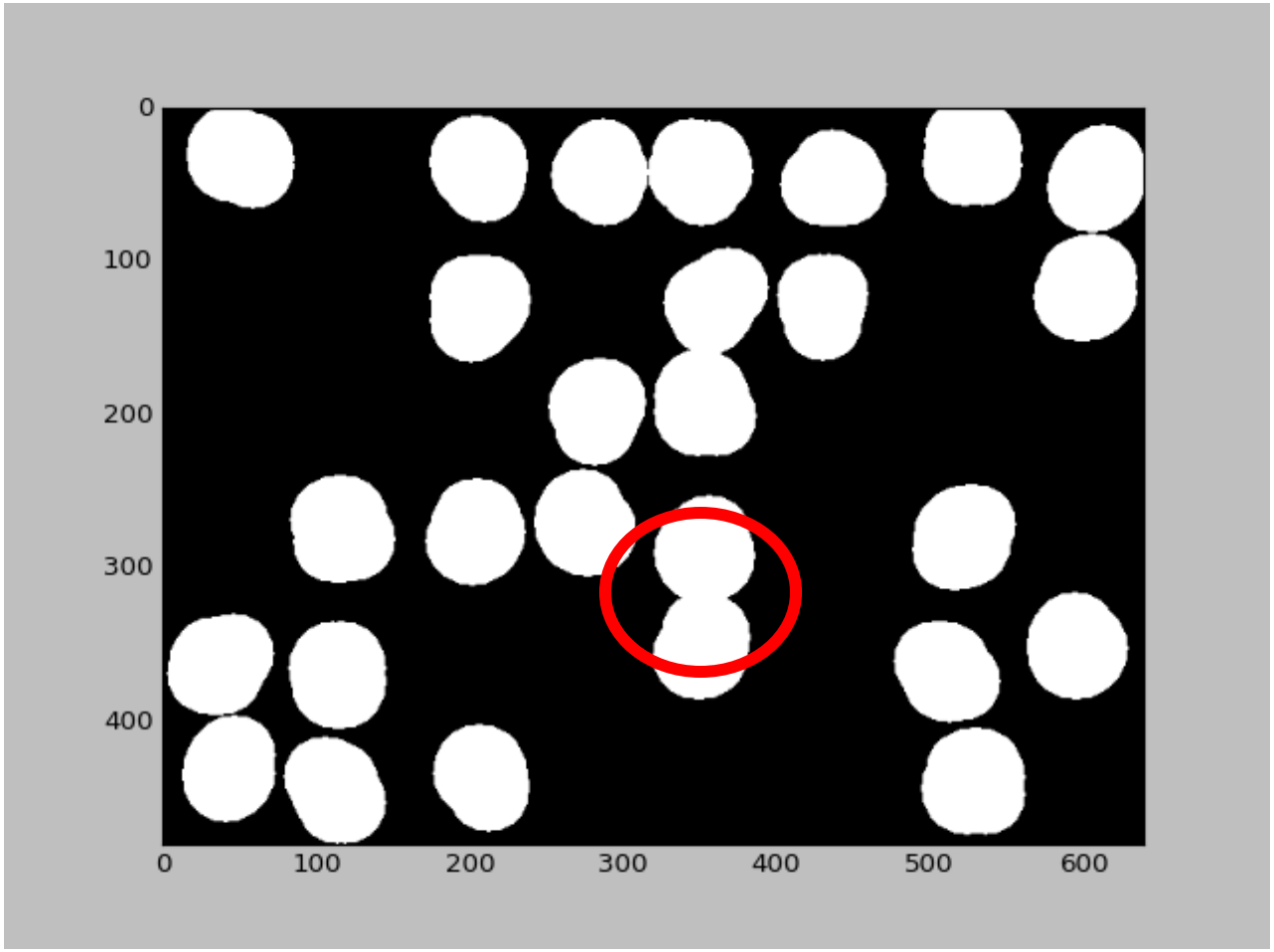
Tačnost: 58%
Da li može da se poboljša?



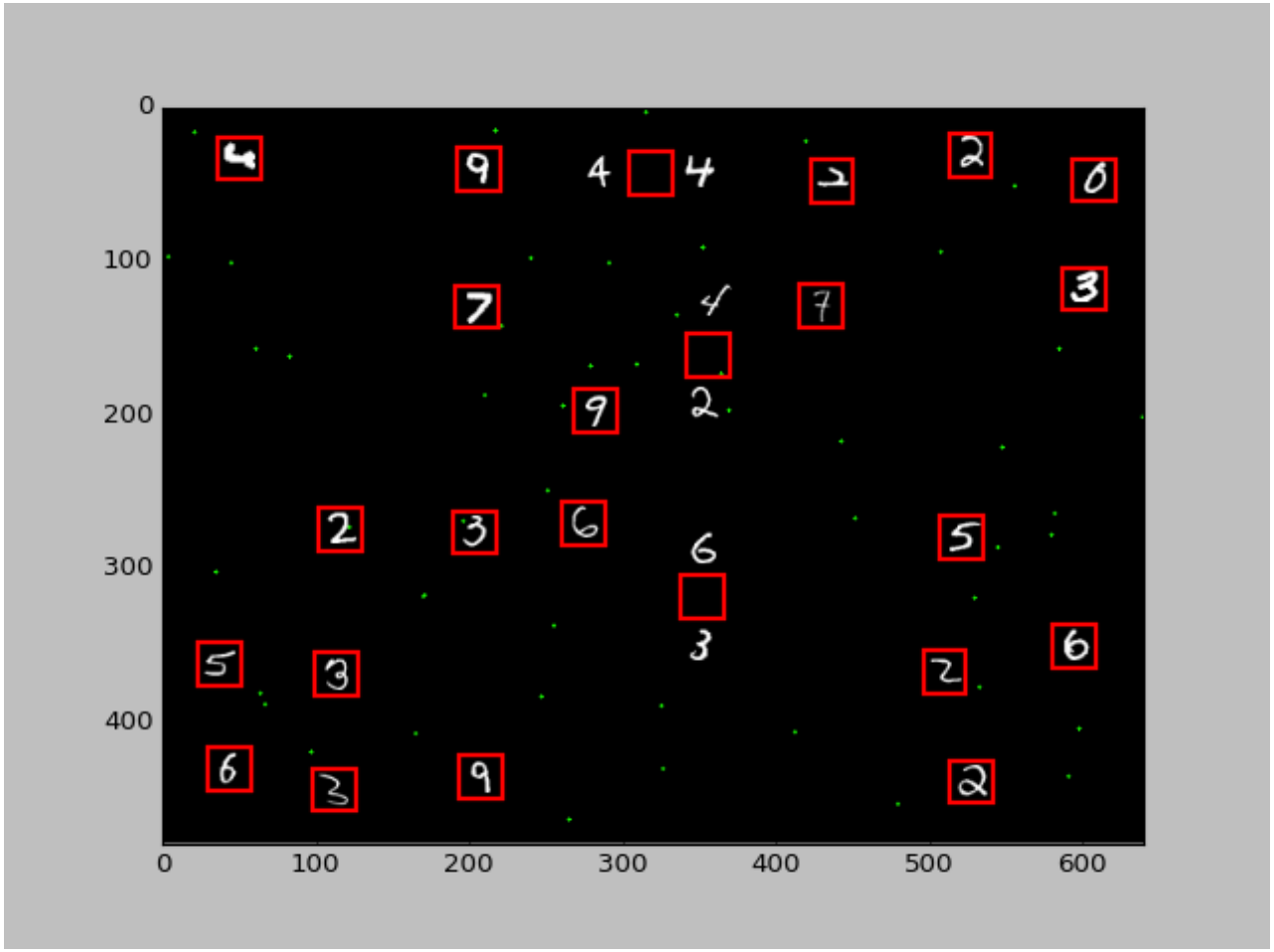
Prethodno: Dilation - disk(15)
Da li se može poboljati određivanje centra?



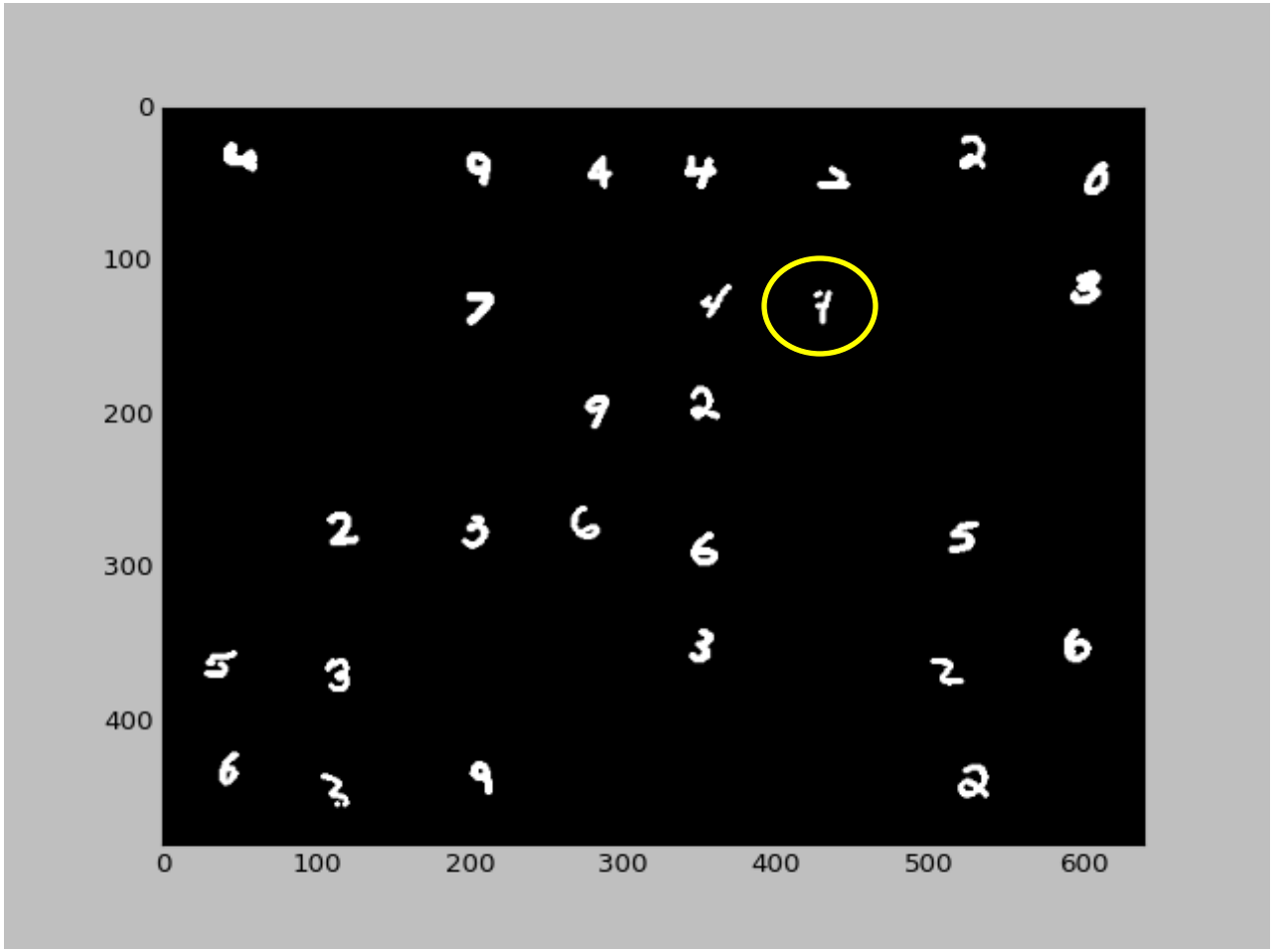
Problem: Dilation
disk(25)



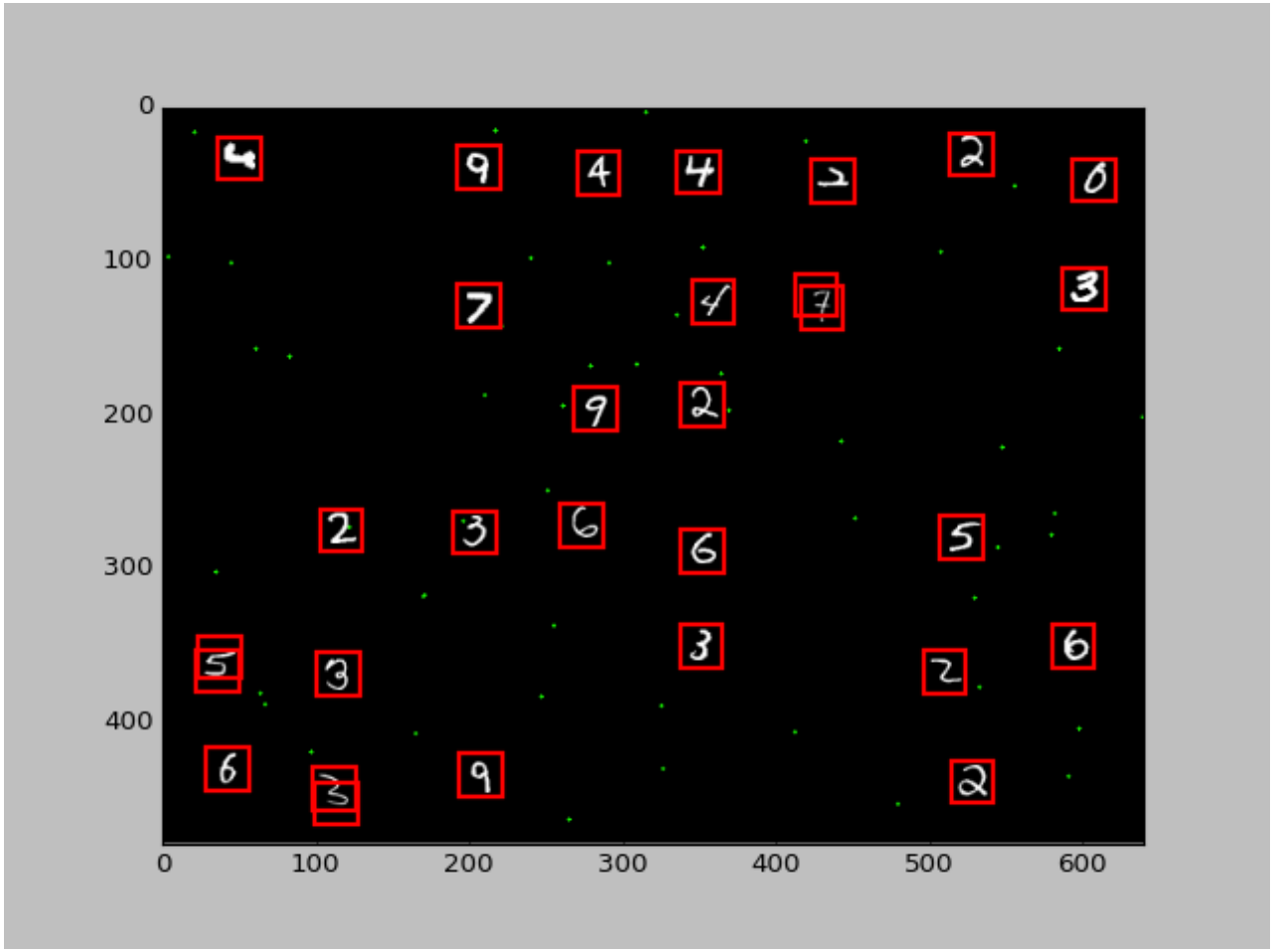
Problem: Dilation - disk(25)
Tačnost: 13%



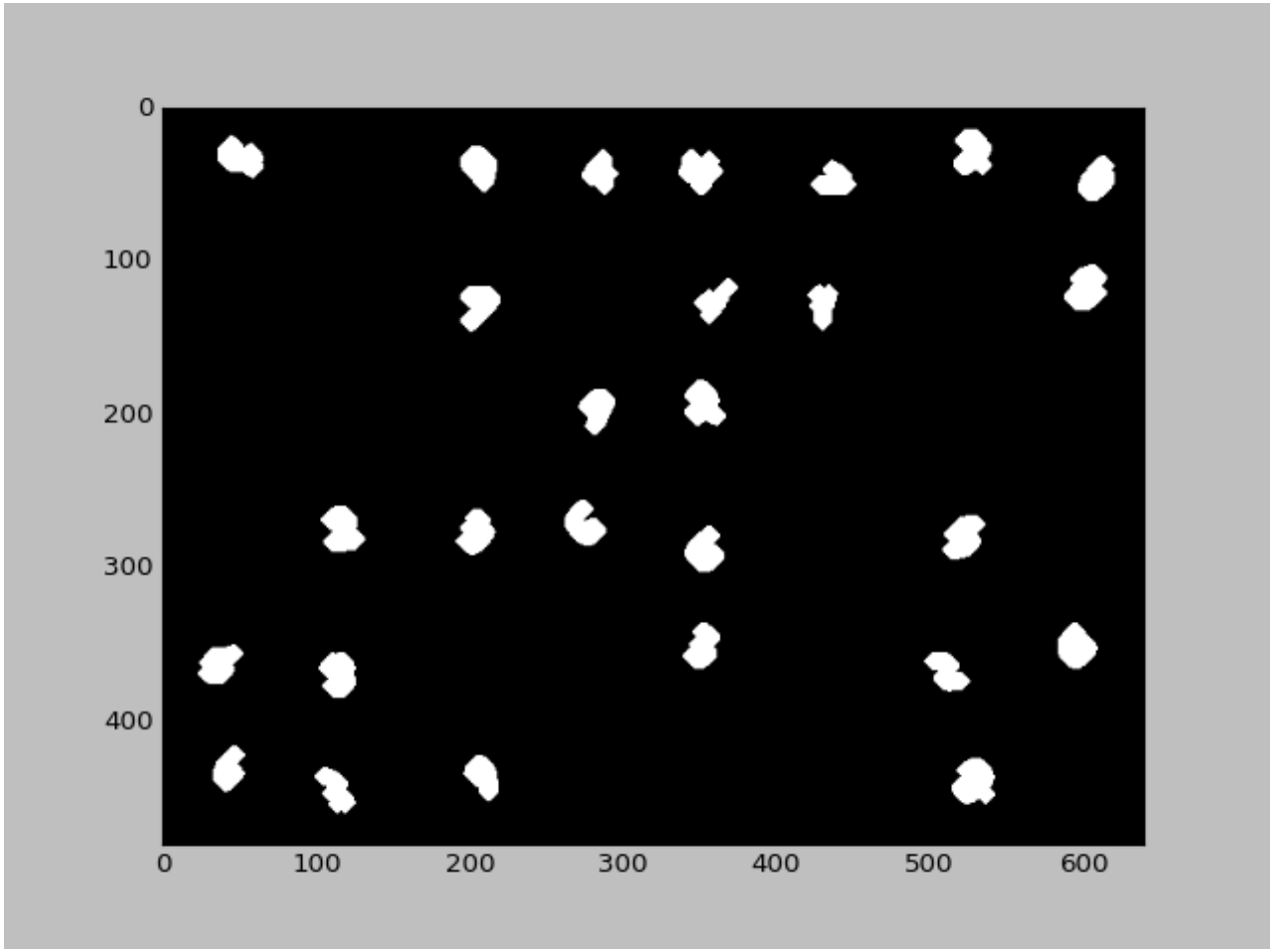
Problem: Dilation
disk(1)



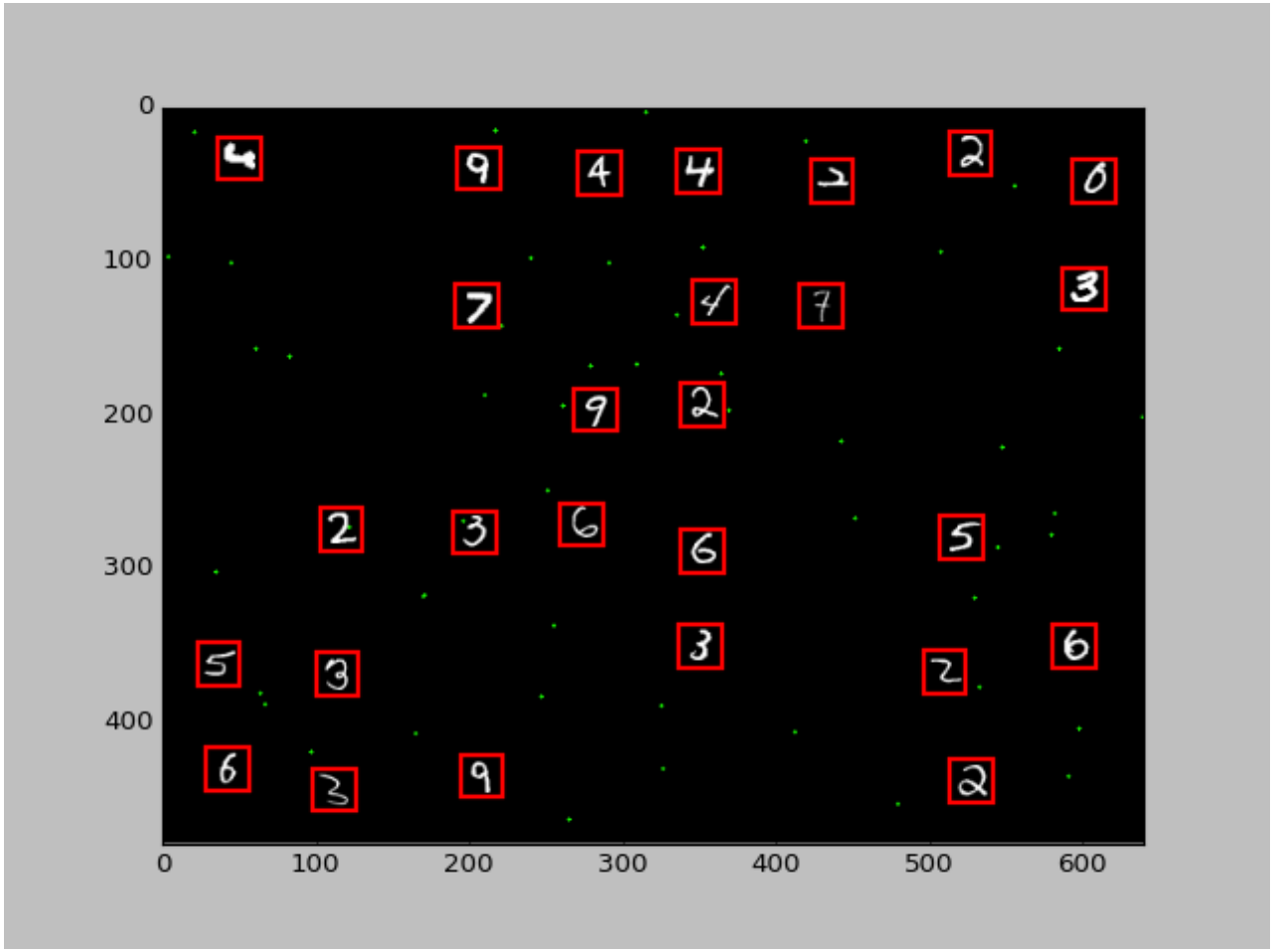
Problem: Dilation - disk(1)
Tačnost: 48%



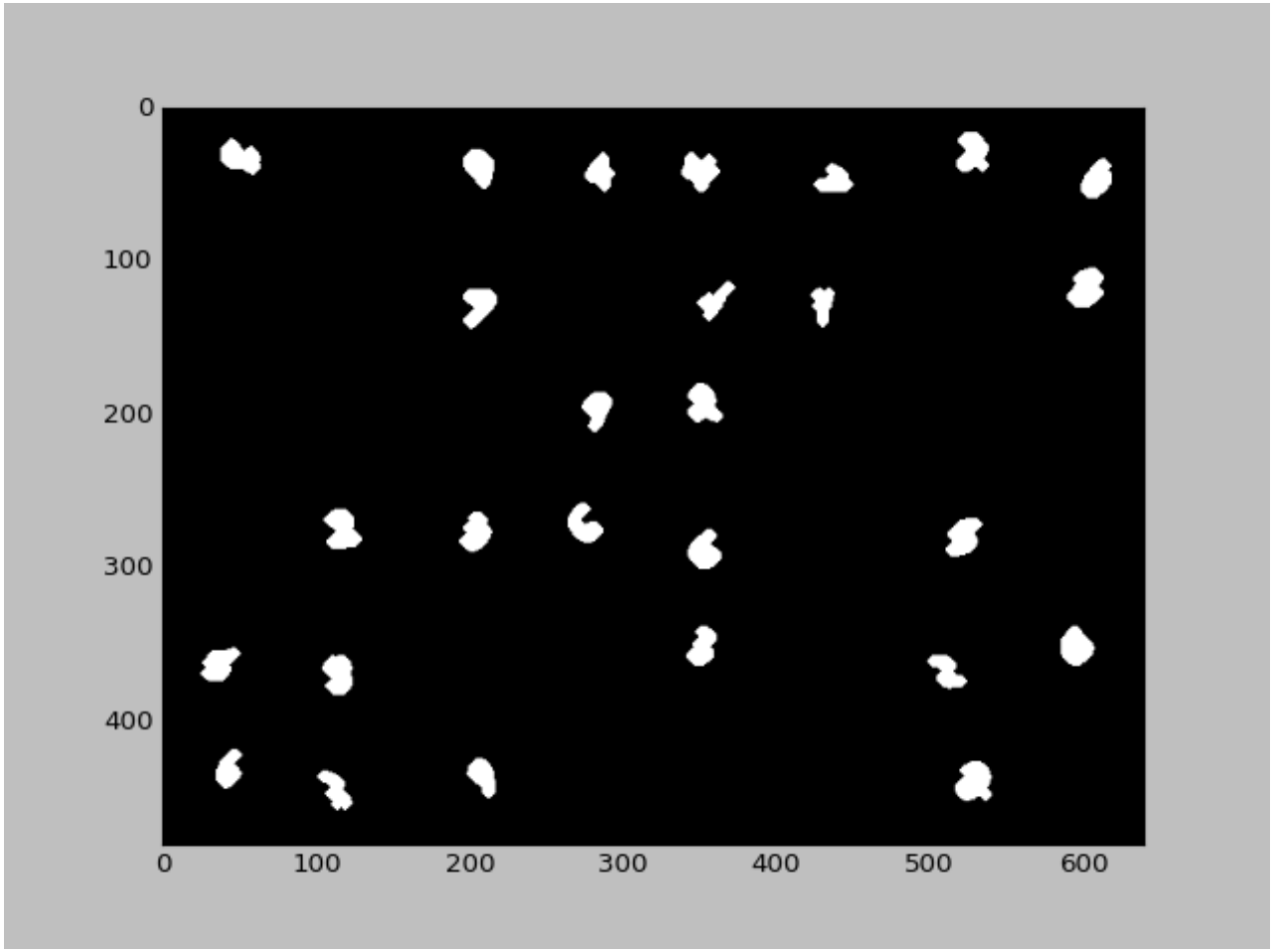
Rešenje: Dilation diamond(5)



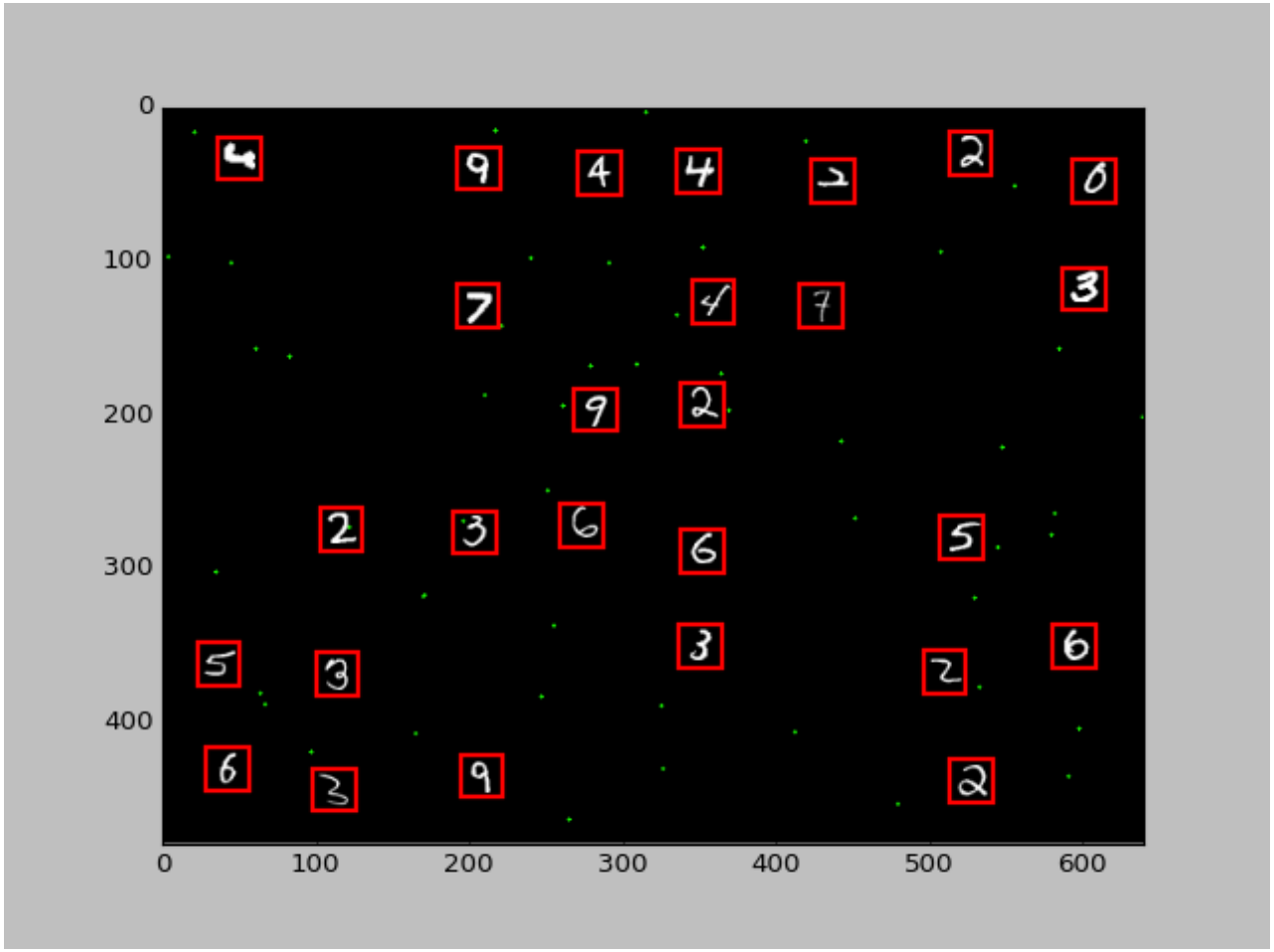
Razlog: oblik brojeva
Tačnost: 65%



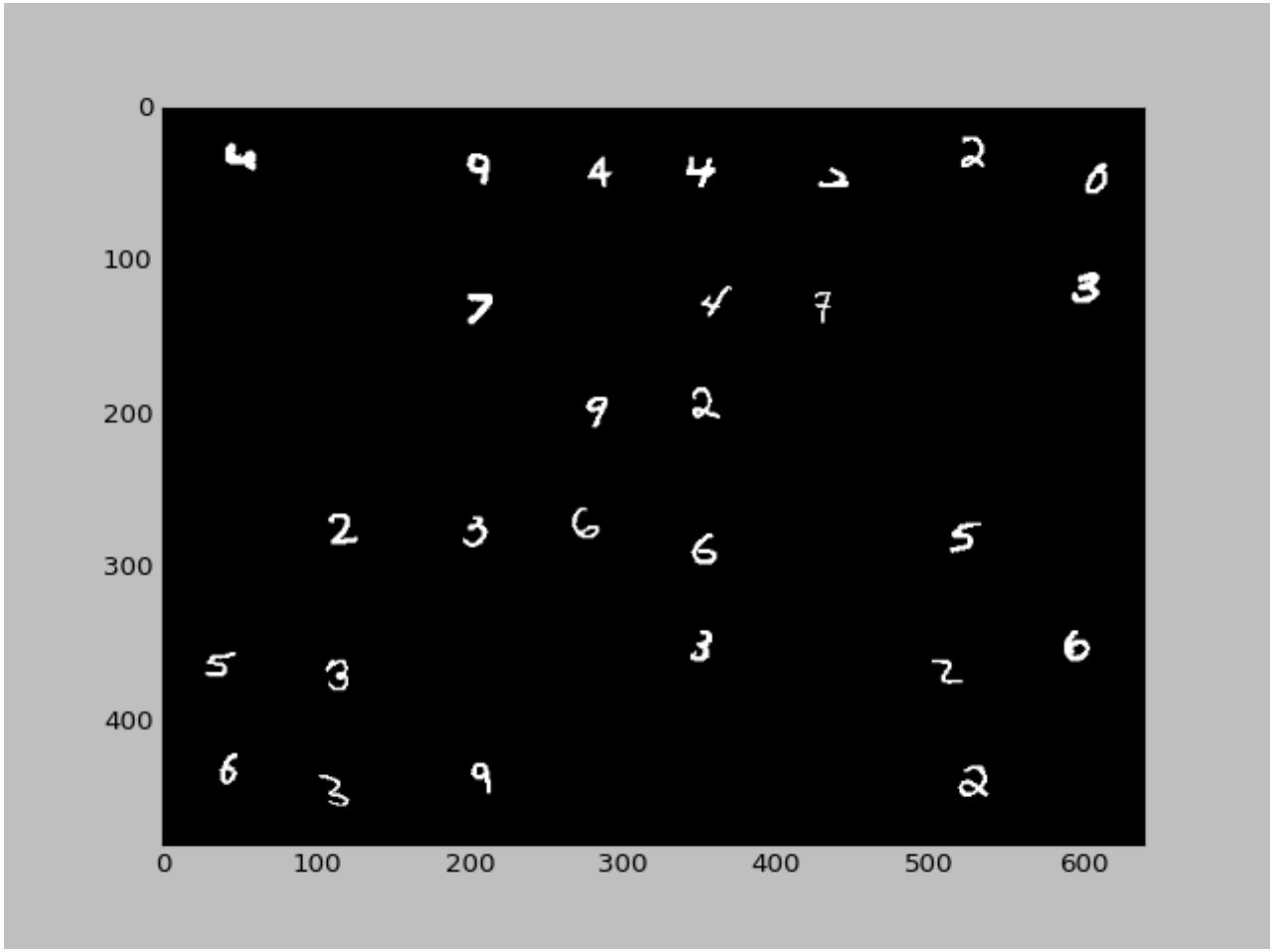
Poboljšanje: Erosion disk(2)



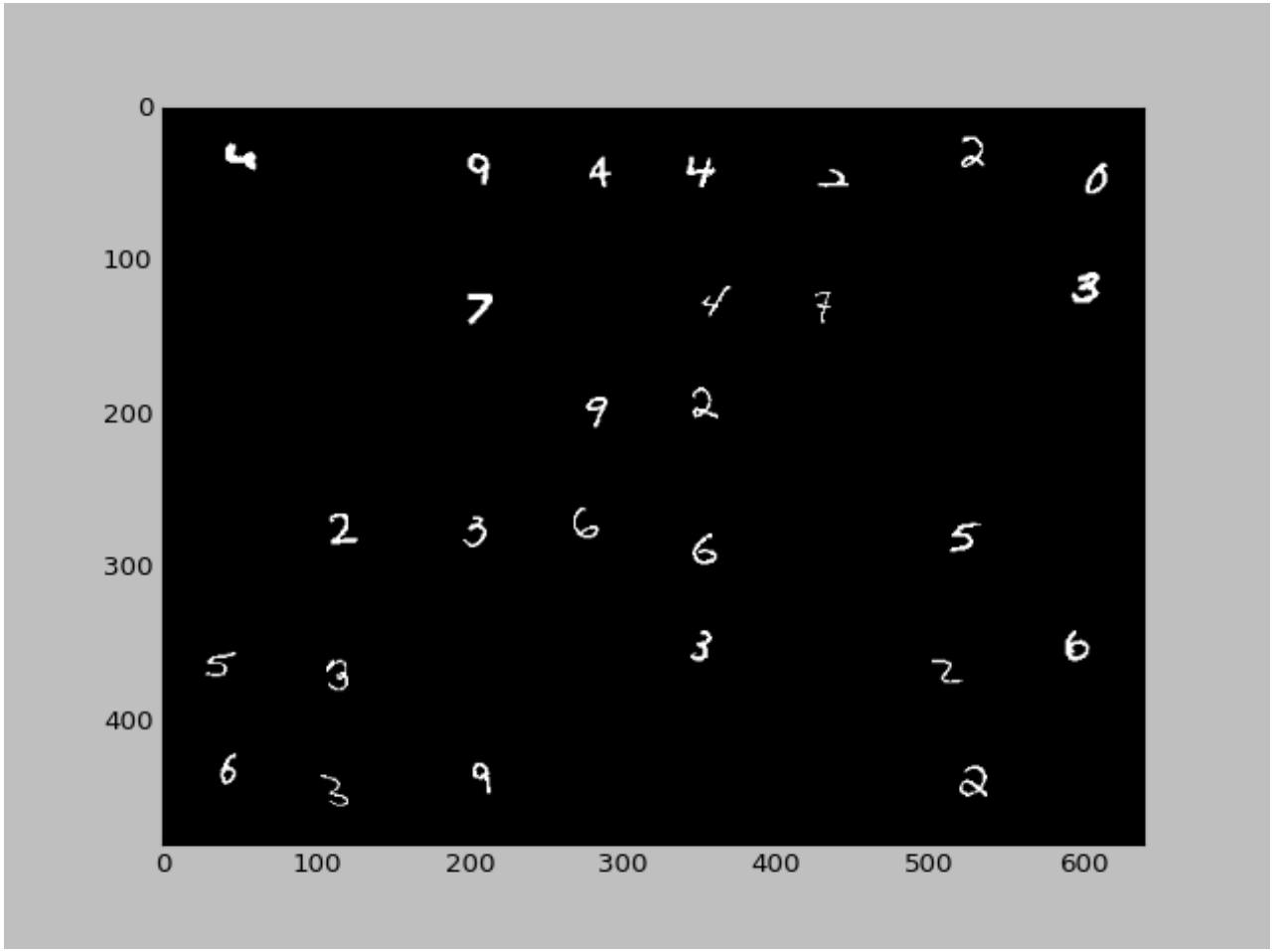
Tačnost: 66%
Šta ako se promeni threshold?



Threshold > 0.03
Tačnost: 63%



Threshold > 0.13
Tačnost: 49%



Poboljšanja

- Za threshold eksperimentalno određeno da bude > 0.054
- Tačnost > 66%

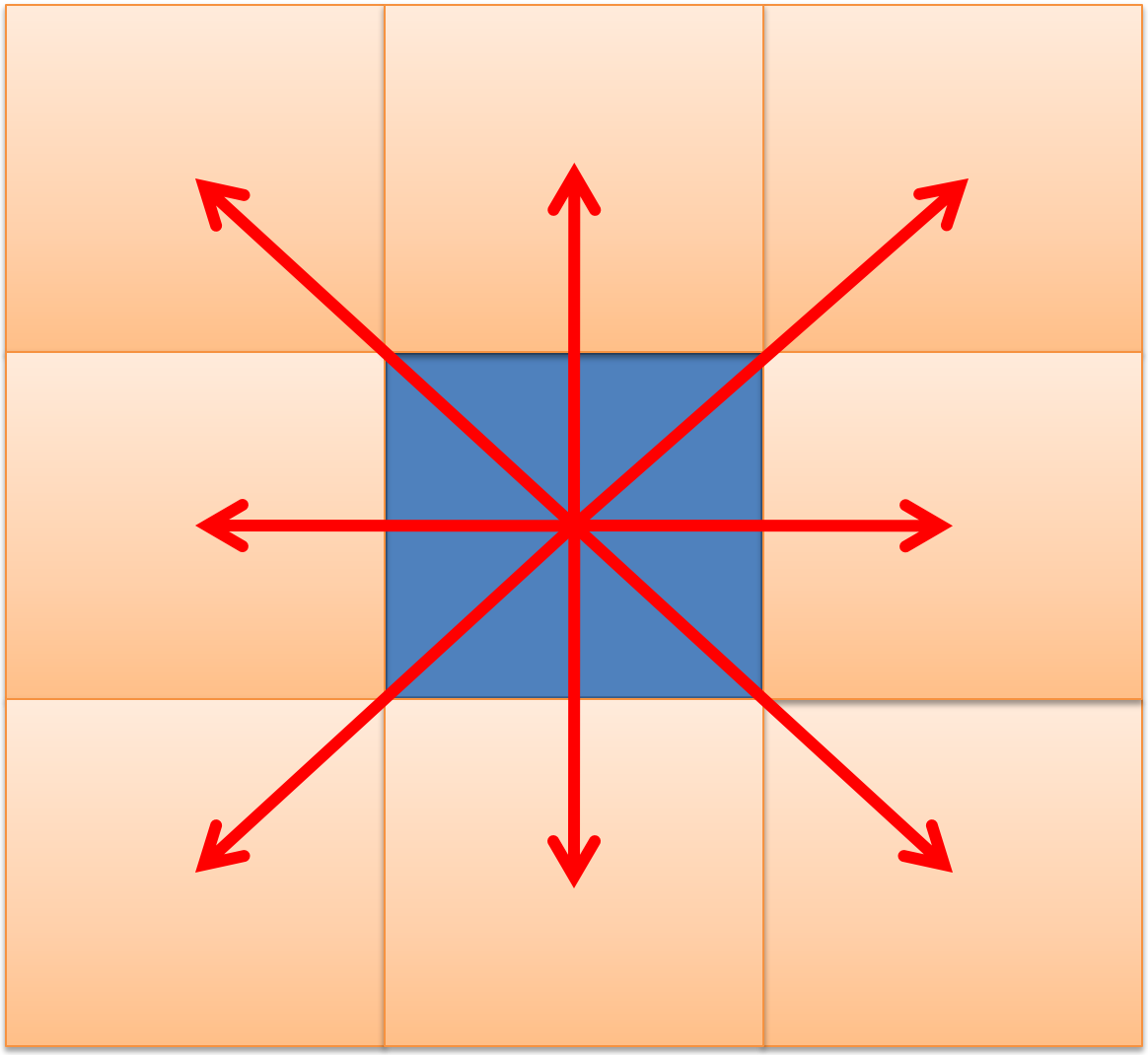
- Koja još poboljšanja da se unesu?
- Nije pomeran okvir oko brojeva
- Potrebno testirati tačnost za pomeranje
- Da li zavisi koliko se pomera okvir?

Pomeranje okvira

- Pomeranje za 2 piksela u svim pravcima daje lošije rezultate od predhodnih: od 4% do 10%
- Pomeranje za 3 piksela u svim pravcima daje još lošije rezultate

- Pomeranje za 1 piksel?

Pomeranje za 1 piksel u svim pravcima

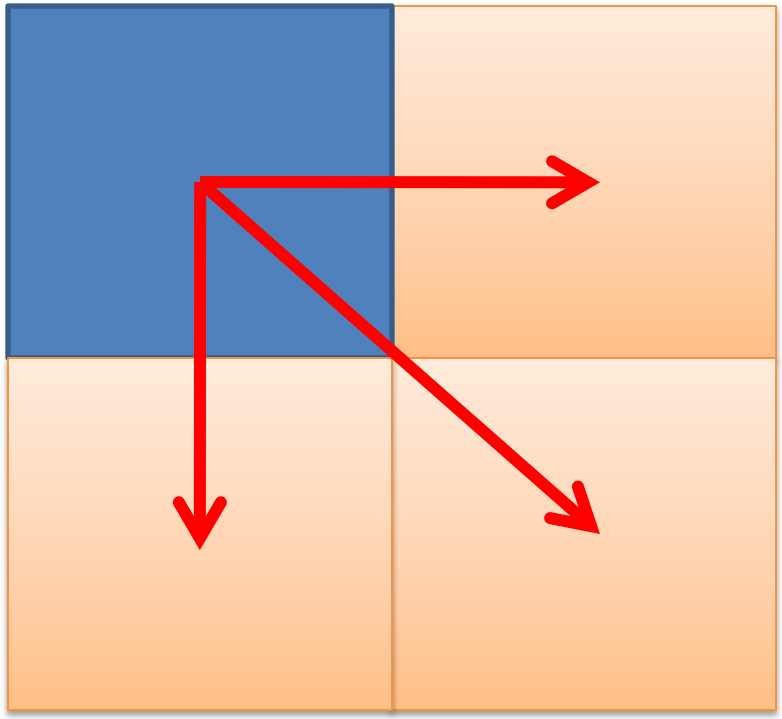


Tačnost: 60%-85%

- U zavisnosti od neuronske mreže, dobija se uglavnom veća tačnost.
- Korišćen algoritam:
  - Traži se najveća verovatnoća broja kod svih slika pomeraja

```
tt = model.predict(np.array([imgB_test]), verbose=0)
t = model.predict_classes(np.array([imgB_test]), verbose=0)
if max_verovatnoca < max(tt[0]):
    max_verovatnoca = max(tt[0])
    max_broj = t[0]
```

Pomeranje za 1 piksel dole-desno



Tačnost: 92%

- Najveća tačnost dobijena je predstavljenom metodom.

Drugi algoritam za određivanje verovatnoće

- Sabiranje svih verovatnoća kod slika pomeraja

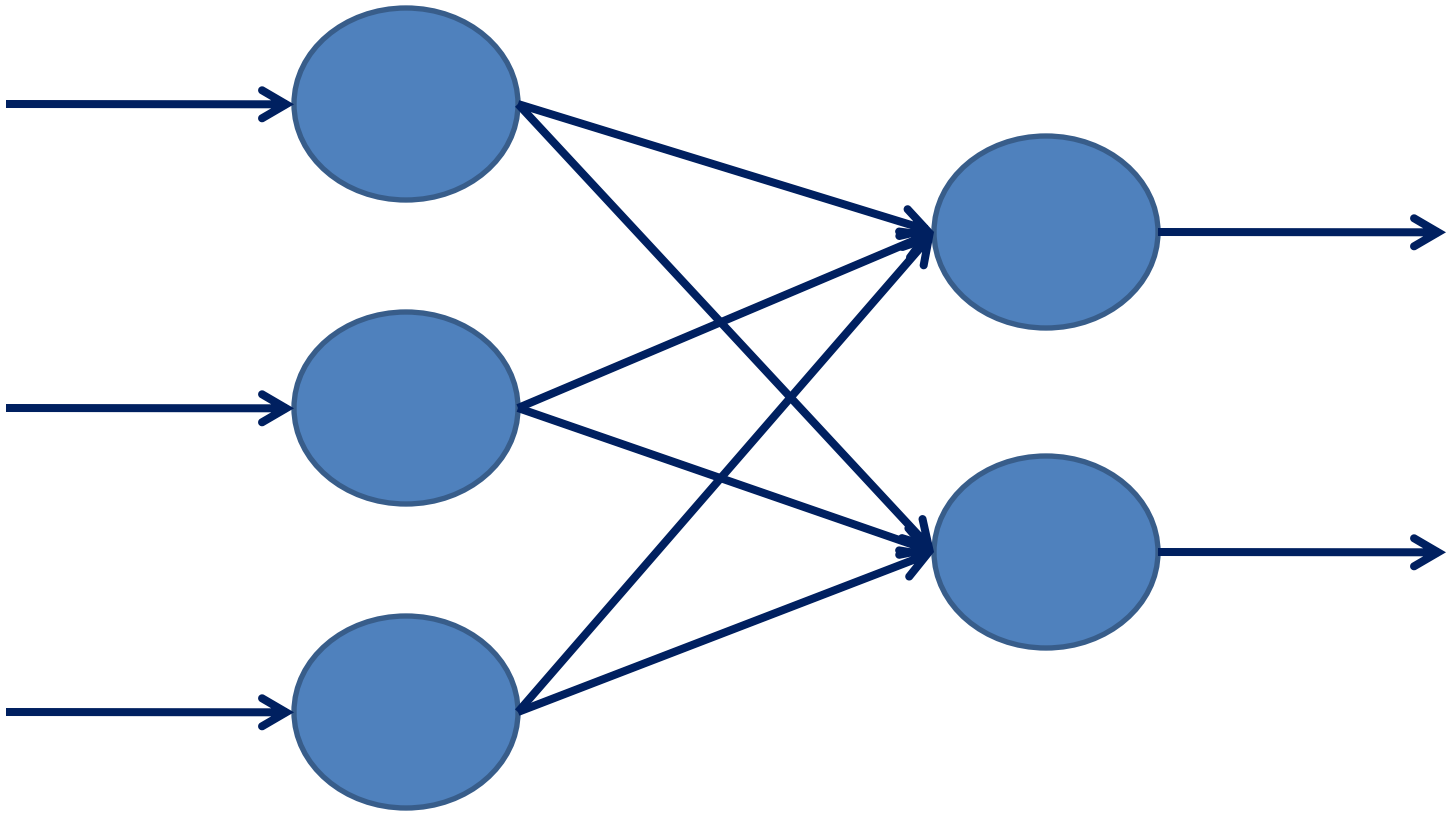
```
        tt = model.predict(np.array([imgB_test]), verbose=0)
        tt2 = tt2 + tt[0]

max_broj = np.argmax(tt2)
```

- Tačnost: 86%

Rezime

- Uklanjanje zelene boje
- Threshold > 0.054
- Opening – square(2)
- Dilation – diamond(5)
- Erosion – disk(2)
- Određivanje regiona
- Pomeraj okvira za 1 piskel dole-desno
- Traži se najveća verovatnoća broja kod svih slika pomeraja

Model neuronske mreže



Model neuronske mreže

- Korišćeni model u prethodnim primerima:
  - Ulaz i izlaz u 1. sloj 784, aktivaciona funckija “relu” (Ractifier)
  - Ulaz u 2. sloj 784, a izlaz 10, aktivaciona funkcija “softmax” (Sigmoidal)

```
#kreiranje modela
model = Sequential()
model.add(Dense(784, input_dim=784, init='normal', activation='relu'))
model.add(Dense(10, init='normal', activation='softmax'))

#compile i fit modela
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(data, train_out, validation_data=(test_data, test_out), nb_epoch=125, batch_size=200, verbose=2)
```

http://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/

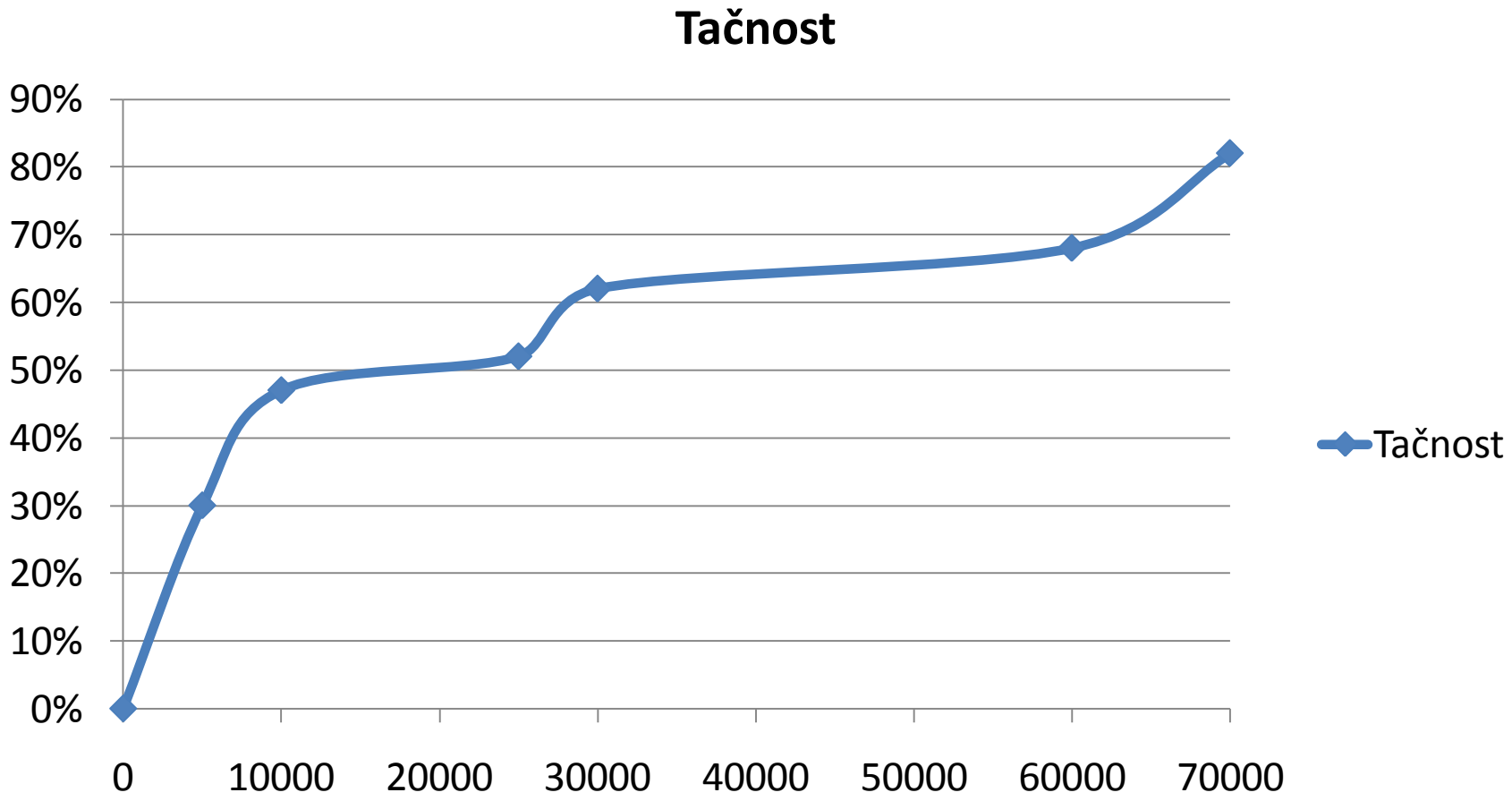
Model neuronske mreže

- Testirana rešenja:
  - Model sa drugim aktivacionim funkcijama – manja tačnost
  - Model sa više slojeva – manja tačnost
  - Model sa različitim izlazima za 1. sloj – manja tačnost
  - Razne kombinacije prethodno navedenog

Model sa 25 epoha

- Obučavajući skup:
  - 5.000 MNIST slika: tačnost: 30%
  - 10.000 MNIST slika: tačnost: 47%
  - 20.000 MNIST slika: tačnost: 52%
  - 35.000 MNIST slika: tačnost: 62%
  - 60.000 MNIST slika: tačnost: 68%
  - 70.000 MNIST slika: tačnost: 82%

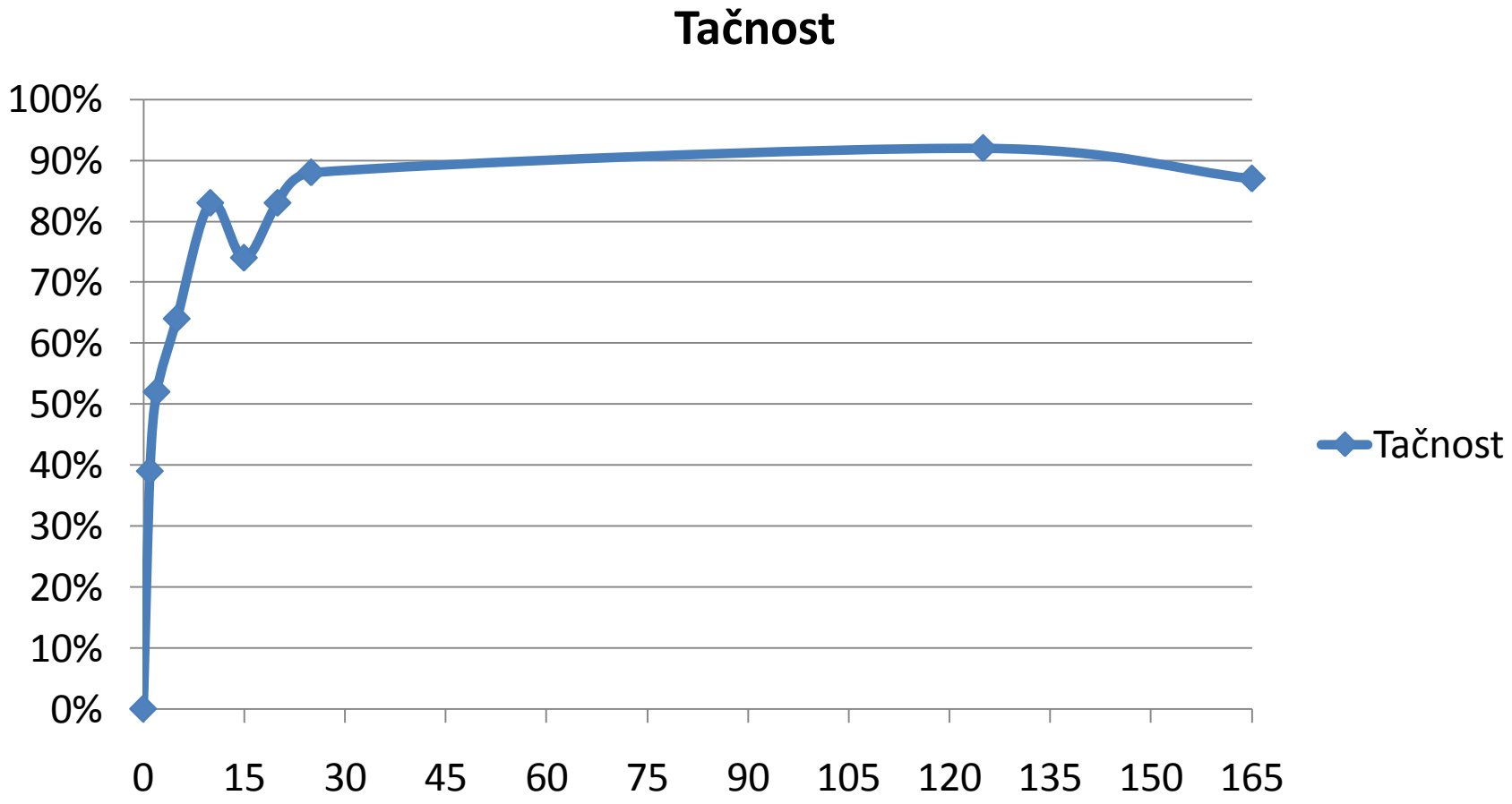
Model sa 25 epoha



Model sa 70.000 MNIST slika

- Broj epoha:
  - 1 epoha: tačnost: 39%
  - 2 epohe: tačnost: 52%
  - 5 epoha: tačnost: 64%
  - 10 epoha: tačnost: 83%
  - 15 epoha: tačnost: 74%
  - 20 epoha: tačnost: 83%
  - 25 epoha: tačnost: 88%
  - 125 epoha: tačnost: 92%
  - 165 epoha: tačnost: 87%

Model sa 70.000 MNIST slika



Napomena

- Dve različite obuke sa istim karakteristikama:
  - 70.000 MNIST slika i 25 epoha
  - Dobijene tačnosti: 82% i 88%
  - Ne moraju uvek iste vrednosti da se dobiju, zavisi od obuke neuronske mreže