Branko Radoš
0036481316

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTA U ZAGREBU
Zavod za automatiku i računalno inženjerstvo

Digitalna obrada i analiza slike

3.6.2018

7: Laboratorisjka vježba - Segmentacija slike

Sadržaj

1. - Uvod

Najjednostavniji i najpoznatiji algoritam grupiranaj jest **algoritam k-srednjih vrijednosti** (engl. k-means algorithm). Algoritmom se primjeri iz neoznačenog skupa primjera $D = {x^{(i)}}_{i=1}^{N}$ grupiranja u K čvrstih grupa, gdje se parametar K zadaje unaprijed. Svaka grupa predstavljena je svojim centroidom ${\mu(i)}_{k=1}^{K}$. Grupiranjem primjera u grupe predstavljen vektorom μ_k nastaje određena pogreška. Pogrešku grupiranja izražava **kriterijska funkcija**:

$$J = \sum_{i=1}^{N} sum_{k=1}^{K} b_k^{(i)} \left\| x^{(i)} - \mu_k \right\|^2 \tag{1-1}$$

gdje je $\|\cdot\|$ **euklidska norma**, tj.

$$\|x - \mu\|^2 = (x - \mu)^T (x - \mu) \tag{1-2}$$

Vrijednost $b_j^{(i)}$ u **** je indikatroska varijabla koja indicira kojoj grupi pripada primjer $x^{(i)}$: ako $b_k^{(i)} = 1$, onda primjer pripada grupi k. Pogreška je jednaka zbroju kvadratnih pogreški, odnosno kvadratima euklidske udaljenosti primjera $x^{(i)}$ od središta μ_k grupe u koju su ti primjeri svrstani. Želimo li minimizirati pogrešku J, svaki primjer $x^{(i)}$ trebamo svrstati, u grupu čije je središte μ_k tom primjeru najbliže, to jest:

$$b_k^{(i)} \begin{cases} 1 & akok = argmin_j \|x^i - \mu_j\| \\ 0 & inae \end{cases}$$

Očito je da bi bilo kakvo drugačije razvrstavanje primjera u grupe rezultiralo samo još većom pogreškom.
Budući da vrijednosti $b_k^{(i)}$ također ovise o μ_k, optimalne vrijednosti za μ_k nije moguće izraziti u zatvorenoj formi. Umjesto toga, algoritam k-srednjih vrijednosti optimizaciju provodi iterativno. Algoritam započinje sa slučajno odabranim srednjim vriejdnostima μ_k . Zatim se u svakoj iteraciji temeljem *** za svaki primjer x^i izračuna vrijednost b_k^i, odnosno svaki se primjer pridjeljuje drupi čijem centroidu najbliži. Nakon toga - budući da sad imamo fiksirane vrijednosti b_k^i - možemo izravno minimizirati pogrešku ***. Postavlajnjem $\nabla_{\mu_k} J = 0$ i rješavanjem po μ_k dobivamo:

$$2 \sum_{i=1}^{N} b_k^i (x^i - \mu_k) = 0 \tag{1-3}$$

iz čega slijedi:

$$\mu_k = \frac{\sum_i b_k^{(i)} x^{(i)}}{\sum_i b_k^{(i)}} \tag{1-4}$$

Vektor μ_k jednak je dakle srednjoj vrijednosti vektora svih primjera koji su stvrstani u grupu k. Budući da je ovime ostvarena promjena vektora μ_k u odnosu na njegovu prethodnu vrijednost , sada treba opet primjeniti izraz *** i ponovno izračunati koji primejri pripadaju grupi k. Ova dva koraka ponavlaju se sve dok se ne dosegne stacionarno stanje, odnosno stanje u kojemu nema daljnjih promjena vrijednosti μ_k.

Algorithm 1 Algoritam 1. Algoritam k-srednjih vrijendosti

inicijaliziraj centroide $\mu_k, k = 1, ..., K$ (npr. na slučajno odabrane $x^{(i)}$);
ponavljaj
za svaki $x^i \in D$;

$$b_k^{(i)} \begin{cases} 1 & akok = argmin_j \|x^i - \mu_j\| \\ 0 & inae \end{cases}$$

za svaki μ_k, k = 1, ..., K
$\mu_k \leftarrow \frac{\sum_{i=1}^{N} b_k^{(i)} x^{(i)}}{\sum_{i=1}^{N} b_k^i}$
dok μ_k ne konvergiraju

Naredba *reshape()* mijenja oblik polja/matrice u želejni uz zadovoljenje nekih uvijeta. na primjer *reshape(A,sz)* mijenja oblik A koristeći vrijednosti vektora sz, konkretnije *reshape(A,[2,3])* mijenja oblik A u 2x3 matricu. sz mora sadržavati barem 2 elementa i *prod(sz)* mora biti jednak *numel(A)*. Gdje je *prod()* umnožak svih vriejdnosi vektora sz, a *numel()* broj elemenata matrice A.

2. - Amplitudna segmentacija

Amplitudna segemntacija je najjednostavnija metoda segmentacije slike. Korisna je kad amplitudne značajke dovoljno precizno definiraju regije scene. Najčešće se za amplitudnu segmentaciju koristi histogram prvog reda.

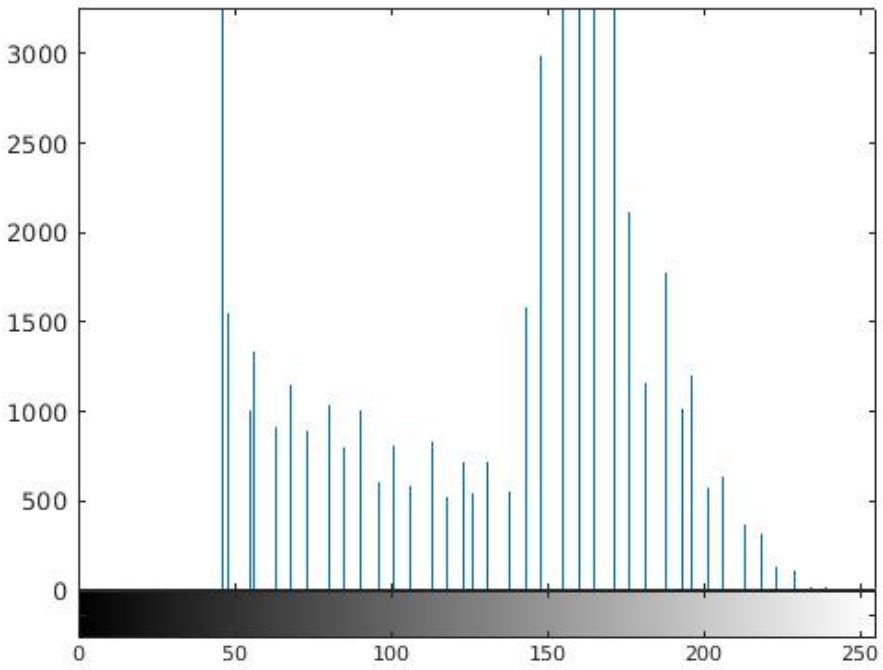
3. - Ručno odabiranje praga

U ovom dijelu vježbe potrebno je na osnovu histograma slike odrediti prag segmentacije. Ukoliko na slici postoji više različitih regija takvih da je svjetlina točaka unutar regije otprilike jednaka, očekujemo bimodalni ili multimodalni histogram. Primjer je prikazan na slici *** . Na tom histogramu vidimo nekoliko skupina, te bi u slučaju segemntacije na dvije regije za prag odabrali vrijednost oko 130.

Za amplitudnu segmentaciju korisitimo funkciju *greyslice()* koja ulazne slike segemntira s zadanim pragovima. Pragovi se navode u jednom vektoru i moraju biti između 0 i 1. Ako izlaz funkcije *greyslice()* nismo pridružili nekoj varijabli, funkciji odmah crta rezultat segemntacije.

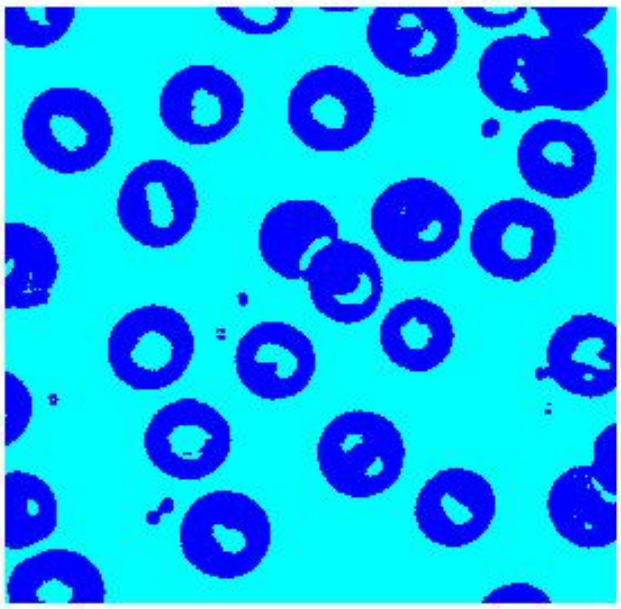
3.1 - Ručno odabiranje praga

Kao optimalan prag obično biramo minimum između dva maksimuma bimodalnog histograma. Ovaj kriterijnekada nazivamo i kriterij srednjeg minimuma.

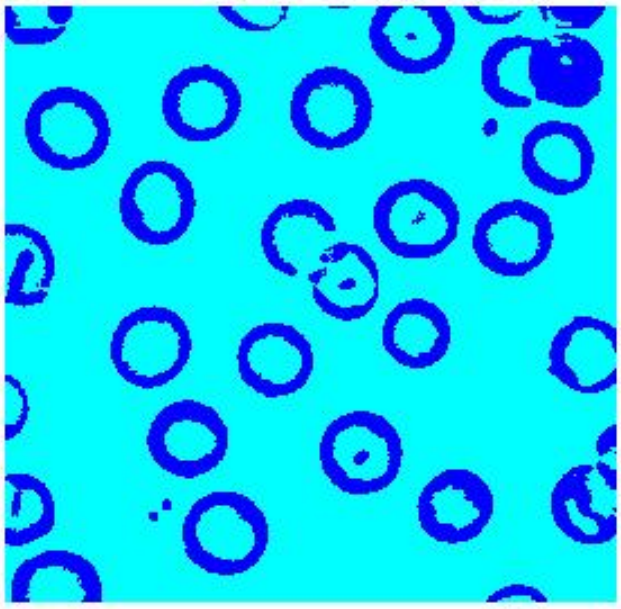


Slika 3-1: Histogram prvog reda slike blood1.tif

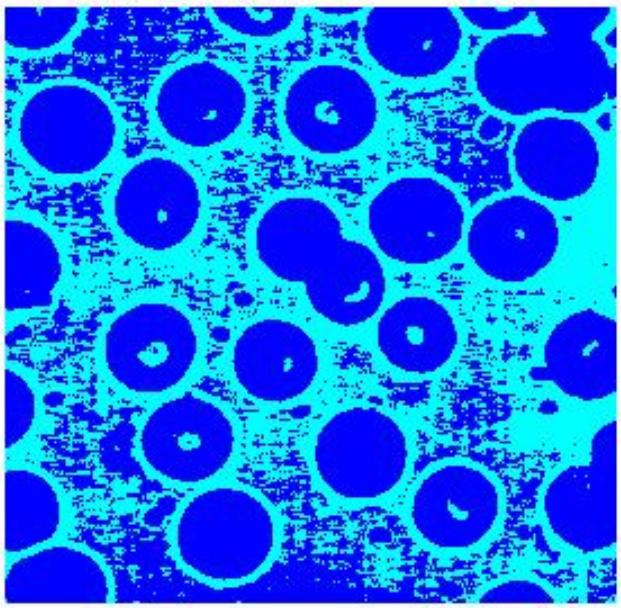
Za segemntaciju s jednim pragom dobivam krvna zrnca , ali ne kristalno čista. Kada odaberem prag 0.2 veći dobivam kao da se nalazi ista materija koja čini krvna zrna i između pojedinih krvnih zrnaca, a za prag manji 0.2 dobivam samo vijenac krvnih zrnaca. Za dva praga slika je bolja nego za dva praga, ali jos uvijek sadrži ne željeni dio (tamno plave točkice na slici).



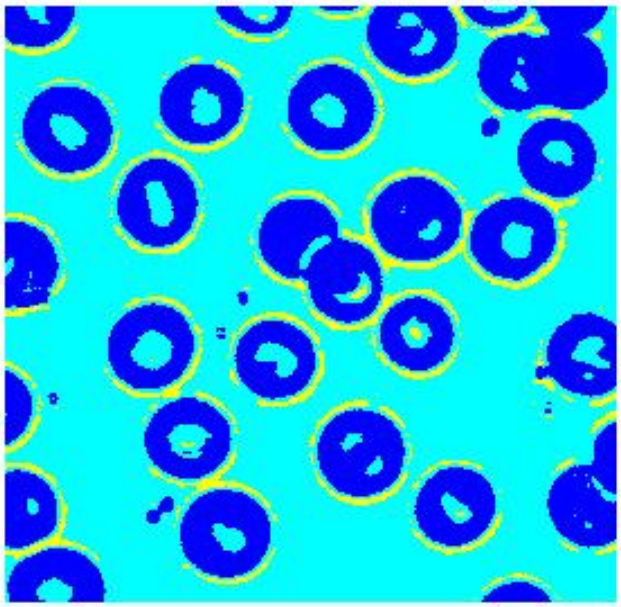
Slika 3-2: Segementacija na dvije regije za prag 107/255 slike blood1.tif



Slika 3-3: Segementacija na dvije regije za prag 56/255 slike blood1.tif



Slika 3-4: Segementacija na dvije regije za prag 158/255 slike blood1.tif

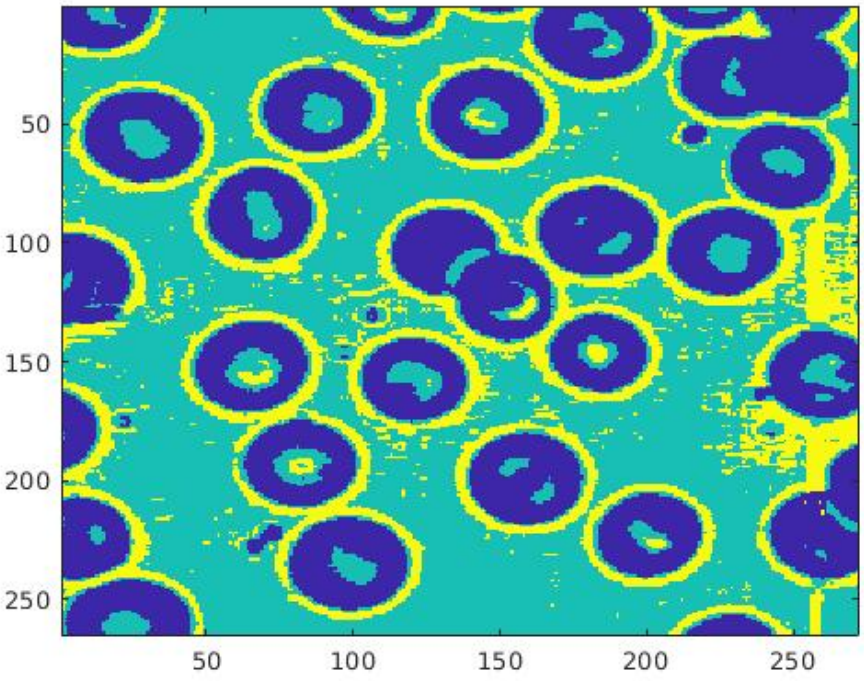


Slika 3-5: segementacija na tri regije za pragove 107/255 i 190/255

4. - Automatsko odabiranje praga

U nastavku vježbe biti će prikazana metoda K-srednjih vrijednosti koja služi za automatsko određivanje praga iz histograma. Za automatsko odabiranje praga koristit ćemo funkciju *kmeans()*.

4.1 - Automatsko odabiranje praga

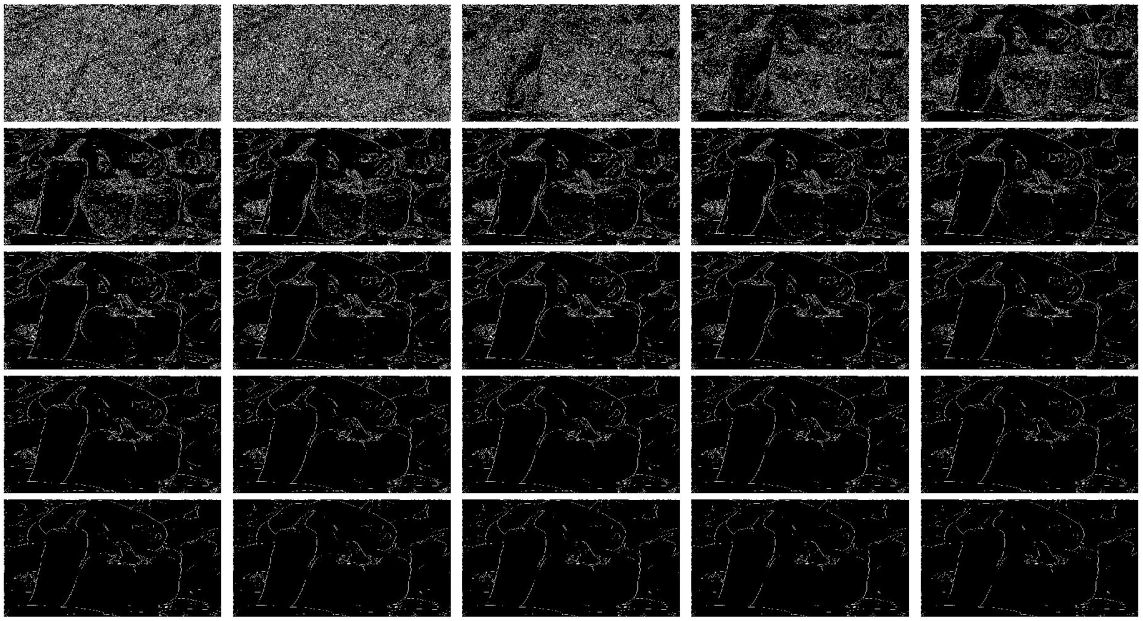


Slika 4-1: Segmentacija slike uzpomoć algoritma *kmeans()*

algoritam *kmeans()* je odradio pedantnije posao, ali pojedini dijelovi su ipak ostali nerješeni.

5. - Određivanje rubova

Za određivanje rubova koristiti ćemo funkciju *edge()*. Podržano je nekoliko različitih metoda određivanja rubova. Funkcija kao rezultat vraća binarnu sliku veličine ulazne slike na kojoj su označeni detektirani rubovi. Podržane metode su *'soble', 'prewitt', 'roberts', 'zerocross', 'log' i 'canny'* . Za metode koje se oslanjaju na procjenu prve derivacije estimacija derivacije se računa za svaku točku te se na tako dobivenoj slici vrši amplitudna segmentacija. Prag za amplitudnu segementaciju moguće je zadati unaprijed. Za metode koje se oslanjaju na procjenu druge derivacije detektira se prolaz kroz nulu. I za te metode je moguće zadati prag, no prag tada određuje strminu ruba.



Slika 5-1: Usporedba dijelovanja sobel operatora za prad od 0 do 0.125 s korakokom 0.005.

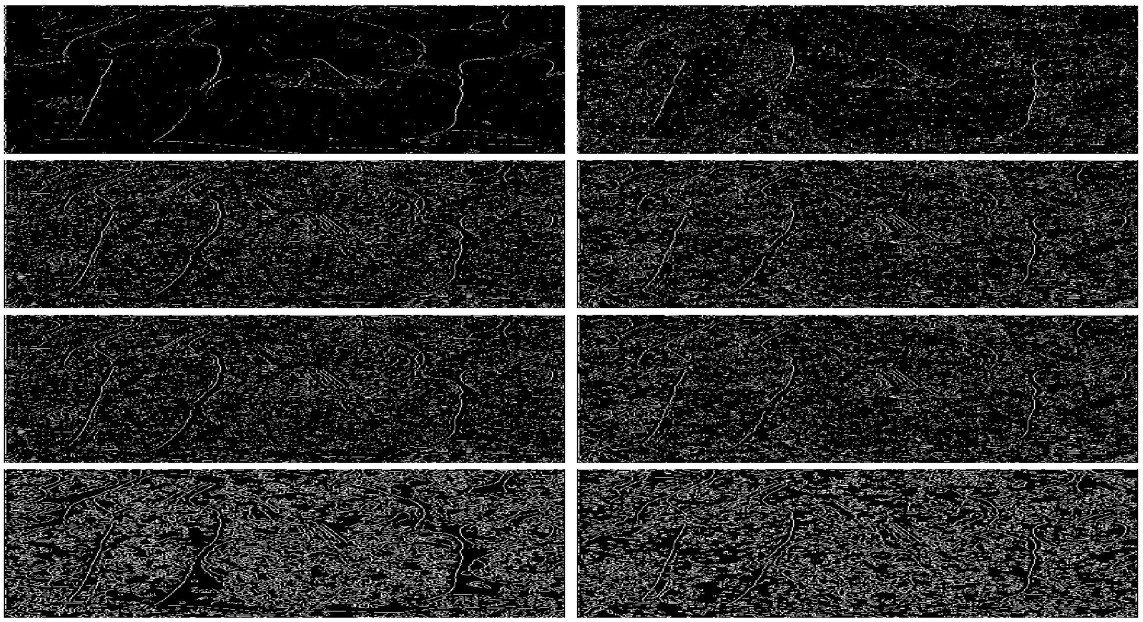
Sobel daje najbolje rezultate za prag 0.085, as automatskim odabirom dobiva se vrijednost 0.0871. Zerocorss i log metode dale su jasnije obrise objekata na slici.

Niti jedna detekcija rubova na impulsni šum dobro ne reagira što je i očekivano jer impulsni šum su maksimumu i minimumu na slici, a na gaussov šum najbolji posao je obavio sobelov operator.

Slika 5-2: Djelovanje sobelovog operatora s automatskim pragom.

Slika 5-3: Djelovanje zerocross operatora s automatskim pragom.

Slika 5-4: Djelovanje log operatora s automatskim pragom.



Slika 5-5: S lijeve strane su slike s gaussovim šumom, a s desne strane su slike s impulsnim šumom. U prvo redu dijelovanje je sobelovog operatora, u drugom dijelu je djelovanje zerocross operatora, u trećem redu je djelovanje log operatora i u četvrtom redu je djelovanje canny operatora.

6. - Segmentacija teksture

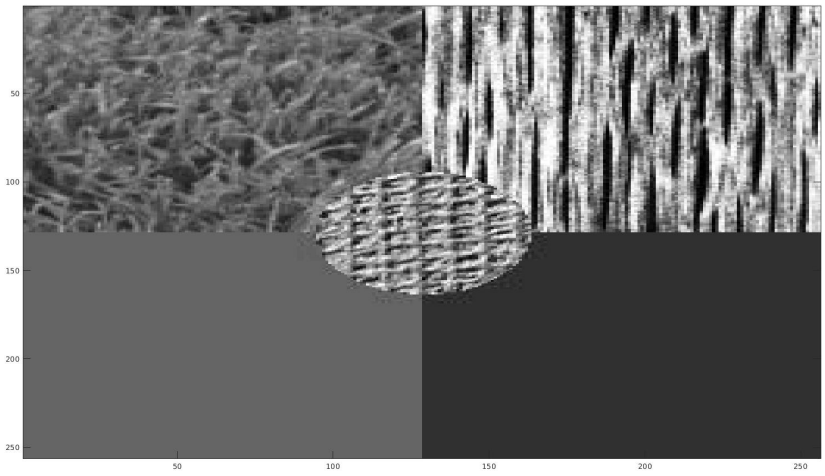
Teksture ćemo segemntirati na temelju nekih odabranih značajki. Najprije ćemo za svaku točku slike izračunati odabranu značajku na odabranoj okolini. Izračunate značajke možemo promatrati kao sliku na kojoj se može izvršiti amplitudna segmentacija. Značajke teksture smo računali koristeći funkciju *colfilt()*. Sada ćemo jednostavno rezultat te funkcije proslijediti funkciji *kmeans()* za automatsku amplitudnu segmentaciju.

Korištene značajke su inercija, kovarijanca i energija bez istosmjerne komponente. Kako same značajke ne preslikavaju 5 struktura u 5 različitih klasa koje se mogu amplitudno odjeliti jedna od druge kmeans nije mogao pronaći 5 grupa koje opisuju pojedinu teksturu.

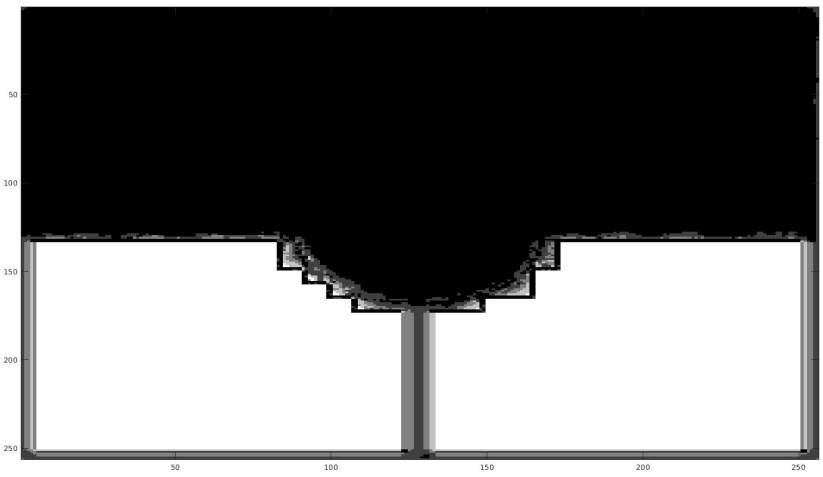
Značajka energija bez istosmjerne komponenete nije pogodna za klasifikaciju jer je 5 različitih tekstura preslikala u dvije klase.

6.1 - Zadatci - Segmentacija teksture

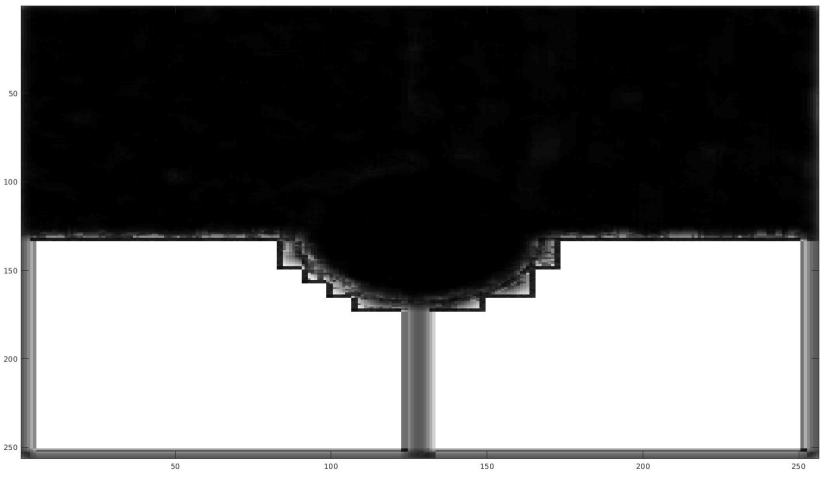
Slika teksture sadrži 5 različitih tekstura. Gore lijevo je slika livade, gore desno je miskroskopski presjek biljke, u sredini je miskrosopski presjek tkanine, dolje lijevo je monokormatska siva boja, a dolje desno akromatska crna boja.



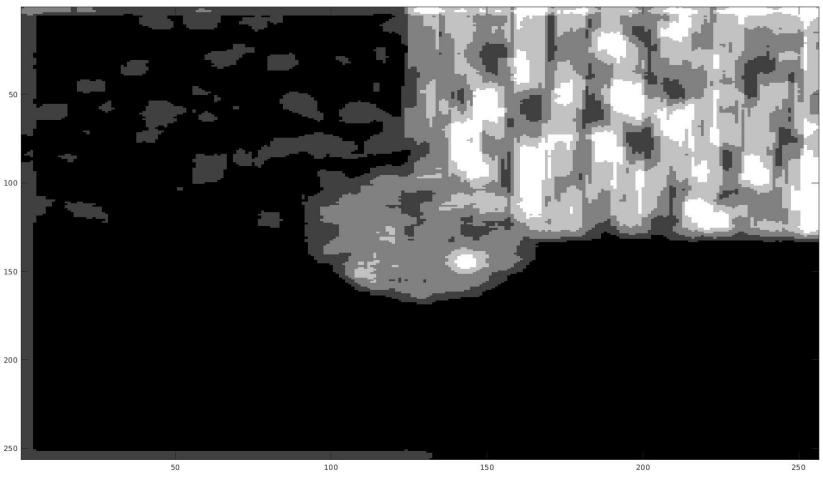
Slika 6-1: Tekstura.



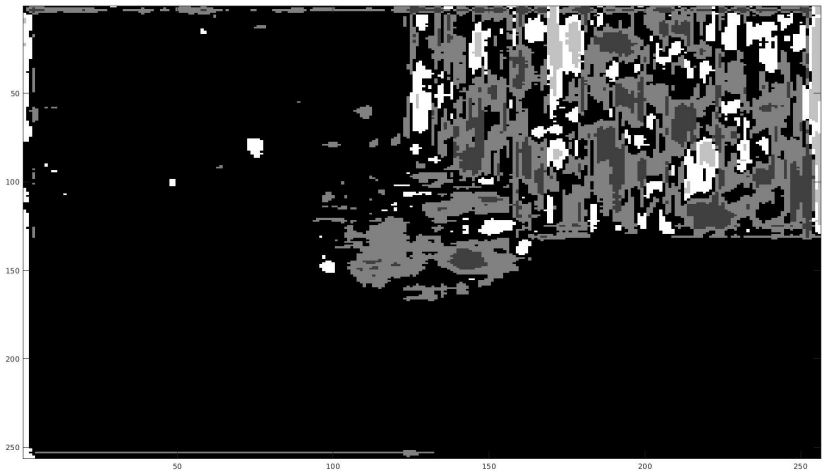
Slika 6-2: Grupacija algoritma *kmeans* u 5 grupa nad slikom kojoj je izvučena značajka energije.



Slika 6-3: Energija spektra bez istosmjerne komponente uz okolinu veličine 12×12.



Slika 6-4: Grupacija algoritma *kmeans* u 5 grupa nad slikom kojoj je izvučena značajka inercije.



Slika 6-5: Grupacija algoritma *kmeans* u 5 grupa nad slikom kojoj je izvučena značajka kovarijance.