UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Jelena Stojiljković

PRIMENA RAZLIČITIH METODA KLASTERIZACIJE NA SEGMENTACIJU SLIKA

master rad

Beograd, 2023.

Mentor:

dr Bojana Milošević, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Vladimir Filipović, redovni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Marko Obradović, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: _____

Porodici

Naslov master rada: Primena različitih metoda klasterizacije na segmentaciju slika

Rezime: Segmentacija slika je proces koji se koristi za razdvajanje slike na različite delove, odnosno segmente, kako bi se pojednostavila analiza i obrada slike. Postoje različite metode za segmentaciju slika, a u ovom radu ćemo se fokusirati na segmentaciju slika primenom metoda klasterizacije.

Klasterizacija je postupak grupisanja sličnih objekata u grupe, poznate kao klasteri, na osnovu njihovih zajedničkih karakteristika. Ova metoda se često koristi u analizi podataka, a u segmentaciji slika se može koristiti za grupisanje piksela sličnih boja ili tekstura u odvojene segmente.

Cilj ovog rada je istražiti različite metode klasterizacije i njihovu primenu u segmentaciji slika. Proučićemo neke od metoda klasterizacije, poput k-sredina, hijerarhijske klasterizacije i klasterizacije pomeranjem sredine, te istražiti njihove prednosti i nedostatke u kontekstu segmentacije slika.

Nakon što smo opisali metode klasterizacije, predstavićemo primer primene ovih metoda na Berkli skupu podataka. Opisaćemo postupak pripreme podataka i implementaciju metoda klasterizacije na slikama ovog skupa podataka. Takođe ćemo uporediti rezultate dobijene primenom ovih metoda klasterizacije i analizirati njihovu efikasnost u segmentaciji slika.

Konačno, u zaključku ćemo sumirati naše nalaze i diskutovati o mogućnostima primene metoda klasterizacije u segmentaciji slika, kao i o mogućim smerovima za buduća istraživanja u ovom području.

Ključne reči: segmentacija, klasterizacija, slika, pikseli, klasteri

Sadržaj

Glava 1

Uvod

Slika se može definisati kao dvodimenzionalna funkcija, $f(x, y)$, gde su x i y prostorne koordinate, a amplituda f u bilo kom paru koordinata (x, y) naziva se intenzitet ili nivo sive u toj tački. Kada su x, y i vrednosti f sve konačne vrednosti, diskretne veličine, sliku nazivamo **digitalnom slikom**. Digitalna slika se sastoji od konačnog broja elemenata, od kojih svaki ima određeni položaj i vrednost. Ovi elementi se nazivaju tačke ili **pikseli**.

Vid je najnaprednije od ljudskih čula, tako da ne čudi slike igraju najvažniju ulogu u ljudskoj percepciji. Međutim, za razliku od ljudi, koji su ograničeni na vizuelni opseg elektromagnetnog (EM) spektra, mašine pokrivaju skoro ceo EM spektar, u rasponu od gama do radio talasa. One mogu da rade na slikama koje su generisali razni izvori uključujući ultrazvuk, elektronsku mikroskopiju kao i kompjuterski generisane slike. Dakle, obrada digitalnih slika obuhvata široko i raznoliko polje primene.

Jedna od prvih primena digitalnih slika bila je u novinskoj industriji, kada su slike prvi put poslate podmorskim kablom između Londona i Nju Jorka. Uvođenjem Bartlejn kablovskog sistema za prenos slike početkom 1920-ih smanjeno je vreme potrebno za transport slike. Specijalizovana oprema za štampanje je kodirala slike za kablovski prenos, s zatim ih rekonstruisala na prijemnom kraju. Slika 1.1 je preneta na ovaj način. Razvoj digitalnih slika ima bogatu istoriju koja seže unazad nekoliko decenija.

- **1950-1960.**: U ovom periodu, istraživači su počeli eksperimentisati sa digitalnom obradom slika koristeći računare. Osnovni koncepti kao što su digitalna reprezentacija slika, kvantizacija i kompresija započeti su u ovom vremenskom

Slika 1.1: Digitalna slika proizvedena 1921. sa kodirane trake telegrafom ([1])

periodu, a jedan od preduslova je bio razvoj programskih jezika visokog nivoa tokom 1950-ih i 1960-ih.

- **1960-1980.**: Prvi računari dovoljno moćni da vrše smislenu obradu slika pojavili su se početkom 1960-ih. Rođenje onoga što nazivamo digitalnom slikom današnje obrade može se pratiti dostupnošću tih mašina i početkom svemirskog programa u tom periodu (Slika 1.2). Takođe, ovaj period obeležen je razvojem prve CCD (Charge-Coupled Device) tehnologije koja je omogućila digitalno snimanje slika. Tada je razvijen i standard za formatiranje i skladištenje digitalnih slika, poznat kao JPEG (Joint Photographic Experts Group) format. Takođe, u ovom periodu su nastali prvi digitalni senzori i kamere.

Slika 1.2: Prva slika Meseca od strane SAD svemirske letelice, 1964. godine ([1])

- **1980-1990.**: U ovom periodu, digitalne slike postaju sve dostupnije širokom auditorijumu. Dolazi do poboljšanja u tehnologiji digitalnih kamera, senzora i softvera za obradu slika. Osnovne tehnike poput filtriranja, kodiranja, dekodiranja i kompresije slika postaju široko rasprostranjene.

- **1990-2000.**: U ovom periodu, digitalna fotografija i obrada slika doživljavaju veliki proboj. Razvijaju se digitalni fotoaparati sa sve boljim senzorima, većom rezolucijom i naprednijim mogućnostima. Takođe, popularizacija računara i interneta omogućava lako deljenje digitalnih slika na globalnom nivou.

- **2000-2010.**: U ovoj deceniji dolazi do eksplozije digitalnih slika zahvaljujući sveprisutnosti digitalnih kamera u mobilnim telefonima. Popularnost društvenih mreža i platformi za deljenje slika dodatno podstiče masovno korišćenje digitalnih slika.

- **2010- danas**: Digitalne slike su postale sastavni deo našeg svakodnevnog života. Napredak u tehnologiji je omogućio visokokvalitetne kamere, senzore i obradu slika u realnom vremenu. Takođe, razvoj veštačke inteligencije i dubokog učenja (eng. deep learning) otvorio je vrata novim mogućnostima u analizi, prepoznavanju i obradi digitalnih slika.

Digitalne slike se matematički predstavljaju kao dvodimenzionalni nizovi piksela, gde svaki piksel nosi informaciju o boji ili intenzitetu svetlosti na određenoj lokaciji u slici. Ova matematička reprezentacija omogućava obradu, analizu i manipulaciju slikama koristeći algoritme i tehnike digitalne obrade slika.

Neki od ključnih pojmova koji se koriste u matematičkoj reprezentaciji digitalnih slika su:

- **Pikseli**: Pikseli su, kao što smo rekli, osnovne jedinice slike. Svaki piksel ima svoje koordinate (x, y) koje određuju njegovu poziciju na slici. Za monohromatsku (crno-belu) sliku, svaki piksel sadrži jednu vrednost koja predstavlja intenzitet ili sivu nijansu piksela, dok multispektralna slika ima vektor vrednosti u svakoj prostornoj tački ili pikselu (ako je slika zapravo slika u boji, onda vektor ima 3 elementa)

- **Rezolucija**: Rezolucija slike određuje broj piksela u horizontalnom i vertikalnom pravcu. Na primer, slika rezolucije 800x600 ima 800 piksela u horizontalnom pravcu i 600 piksela u vertikalnom.

- **Intenzitet**: Intenzitet piksela na crno-beloj slici predstavlja nijansu sive ili svetlost koja se nalazi u toj tački slike. Intenzitet može biti predstavljen brojem iz opsega 0 do 255, gde 0 predstavlja crnu boju, a 255 predstavlja belu boju.
- **Modeli (prostori) boja**: Modeli boja se koriste za predstavljanje boja u slikama. Najčešći model boja je RGB model koji koristi kombinaciju crvene (R), zelene (G) i plave (B) komponente za prikazivanje širokog spektra boja.
- **Histogram**: Histogram predstavlja raspodelu intenziteta piksela na slici. On prikazuje koliko često se određeni intenziteti pojavljuju na slici, što može pružiti informacije o kontrastu, osvetljenju i raspodeli boja na slici.
- **Transformacije slika**: Transformacije slika uključuju različite operacije koje se primenjuju na sliku kako bi se promenio njen izgled ili obezbedile određene informacije. To mogu biti transformacije poput skaliranja, rotiranja, promene veličine, izrezivanja, kao i operacije poput filtriranja, segmentacije, detekcije ivica i mnoge druge.

Matematička reprezentacija digitalnih slika je osnova za širok spektar algoritama i tehnika digitalne obrade slika, uključujući filtriranje, segmentaciju, detekciju oblika, prepoznavanje uzoraka i druge postupke koji omogućavaju analizu i manipulaciju slikama u digitalnom okruženju.

Glava 2

Segmentacija

Segmentacija je podela nečega na različite delove, prema [2]. Ovaj pojam je nastao od latinske reči *segmentum*, što u prevodu znači deo ili sečenje. Segmentacija generalno govoreći je proces podele nekog većeg skupa ili objekta na manje delove, poznate kao segmenti, sa ciljem da se lakše obrade, analiziraju ili razumeju. Segmentacija se primenjuje u različitim oblastima, od obrade signala, preko obrade slika i videa, do obrade teksta i drugih vrsta podataka.

2.1 Segmentacija kod ljudi

Segmentacija po principima geštalt psihologije zasniva se na ideji da ljudski mozak ima prirodnu sklonost da organizuje vizuelne informacije u smislene grupe ili oblike na osnovu nekih univerzalnih principa. Ovi principi su poznati kao zakoni geštalt psihologije i neki od njih su:

Princip sličnosti - objekti koji su vizuelno slični se grupišu zajedno. Na primer, ako imamo skup od nekoliko krugova, kvadrata i trouglova, mozak će ih grupisati zajedno na osnovu njihovih sličnosti.

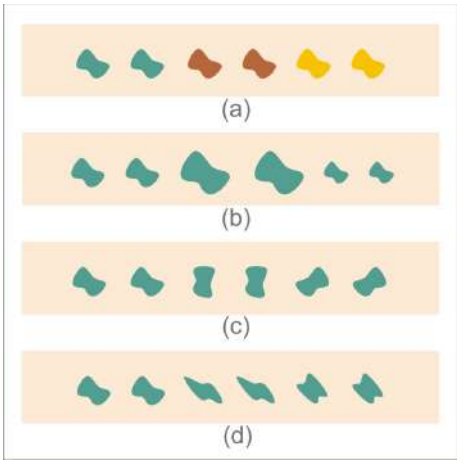
Princip blizine - objekti koji su prostorno blizu se grupišu zajedno. Na primer, ako imamo skup od nekoliko krugova, kvadrata i trouglova koji su raspoređeni u paralelnim redovima, mozak će ih grupisati zajedno po redovima.

Princip dobrog oblika - objekti se grupišu zajedno u smislene oblike. Na primer, ako imamo skup od nekoliko tačaka, mozak će ih grupisati u smislene oblike, kao što su krugovi ili trouglovi.

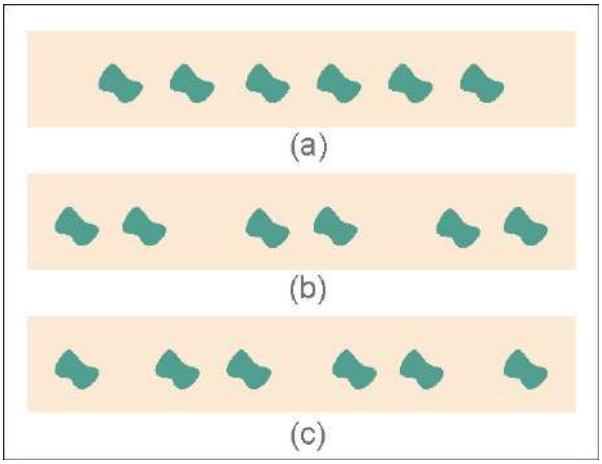
Princip kontinuiteta - objekti koji se nastavljaju jedan na drugi se grupišu zajedno. Na primer, ako imamo skup linija koje se prekidaju na određenim mestima,

mozak će pokušati da ih grupiše zajedno u kontinuirane linije.

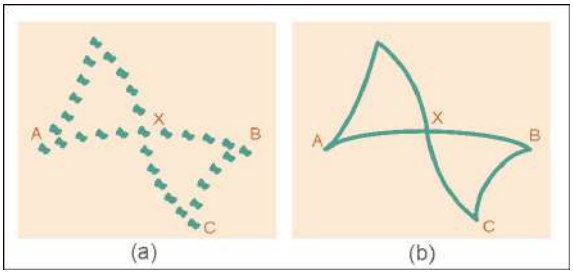
Neki od ovih principa su prikazani na Slikama 2.1, 2.2 i 2.3.



Slika 2.1: Princip sličnosti ([3])



Slika 2.2: Princip blizine ([3])



Slika 2.3: Princip dobrog oblika i kontinuiteta ([3])

2.2 Segmentacija u digitalnoj obradi slika

Geštalt principi se primenjuju i u procesu segmentacije slika tako što se pokušavaju pronaći smislene grupe ili oblici na slici. Na primer, algoritmi segmentacije slika mogu koristiti princip sličnosti da grupišu piksele sličnih boja zajedno u smislene oblike ili princip blizine da grupišu piksele koji su prostorno blizu zajedno u smislene regione.

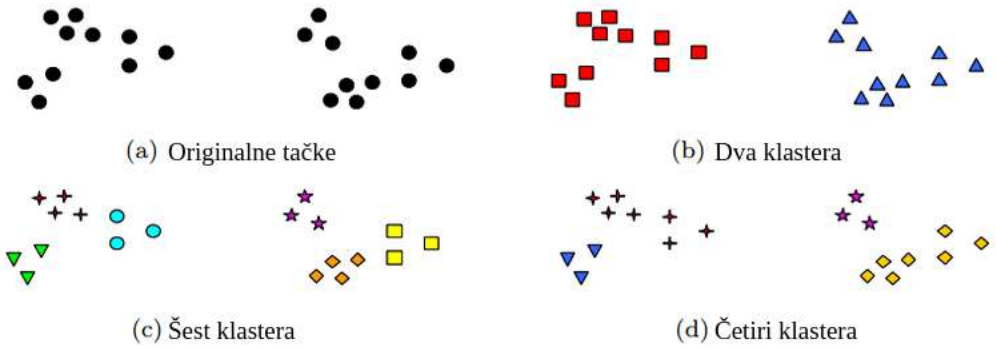
Primena geštalt principa segmentacije u digitalnoj segmentaciji slika može biti složen proces koji uključuje kombinovanje različitih principa i algoritama. Međutim, korišćenjem ovih principa i algoritama moguće je pronaći oblasti na slici koje imaju zajedničke karakteristike i koje se mogu koristiti u različitim aplikacijama, kao što su prepoznavanje uzoraka, detekcija objekata, analiza slika i drugo.

Glava 3

Klasterovanje

3.1 Familije klasterovanja

Pojam klasterovanja nije jednoznačno definisan (Slika 3.1). Zbog raznovrsnosti konteksta u kojima se grupisanje može vršiti i ciljeva koji se pomoću klasterovanja žele postići, postoji više neformalnih definicija i podela klasterovanja.



Slika 3.1: Različiti klasteri ([4])

Najopštije, sve algoritme klasterovanja možemo podeliti u 5 osnovnih familija, mada postoje i neke finije klasifikacije ovih algoritama:

1. Hijerarhijsko klasterovanje (eng. Hierarchical clustering);
2. Particione metode (eng. Centroid-based clustering);
3. Metode zasnovane na gustini (eng. Density-based clustering);

4. Metode zasnovane na raspodeli (eng. Distribution-based clustering);

5. Metode zasnovane na mreži (eng. Grid-based clustering).

U daljem tekstu biće opisana svaka od njih.

Hijerarhijsko klasterovanje

Hijerarhijsko klasterovanje ([5]) je vrsta klasterovanja koja kreira hijerarhiju klastera od početnih podataka. To se postiže tako što se podaci dele na sve manje klastera sve dok se ne dođe do pojedinačnih elemenata. Hijerarhijsko klasterovanje se može podeliti na dve glavne vrste: hijerarhijsko klasterovanje deljenjem (eng. divisive hierarchical clustering) i hijerarhijsko klasterovanje spajanjem (eng. agglomerative hierarchical clustering).

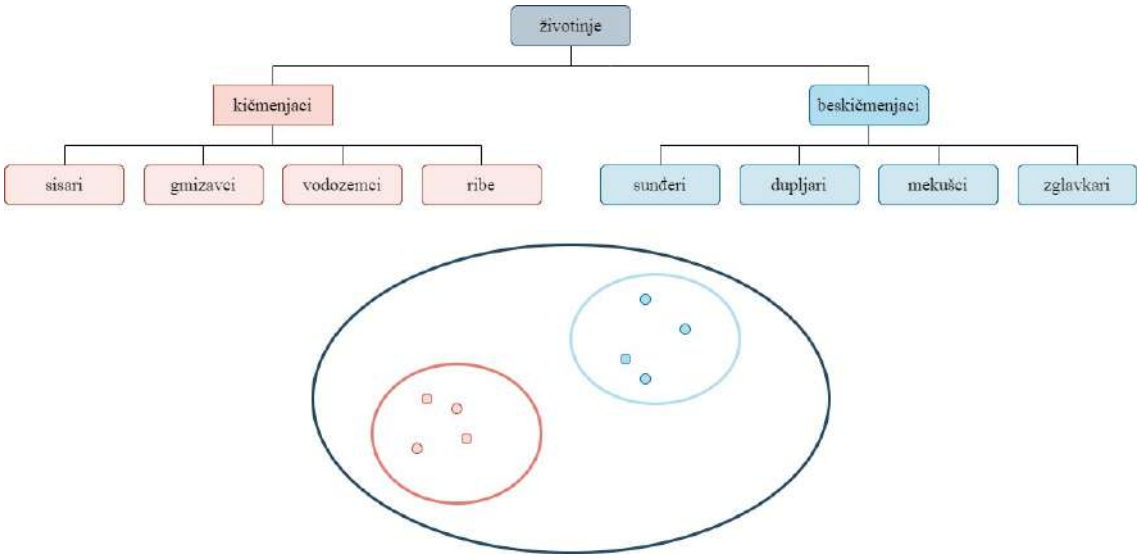
Hijerarhijsko klasterovanje deljenjem počinje sa jedinstvenim skupom podataka i razdvaja ih u sve manje klastera sve dok se ne dođe do željenog broja klastera. Hijerarhijsko klasterovanje spajanjem počinje sa pojedinačnim klasterima za svaki element u skupu podataka i spaja ih u sve veće klastere sve dok se ne dođe do jedinstvenog klastera.

Hijerarhijsko klasterovanje je korisno u situacijama kada nije poznat broj klastera ili kada se želi istražiti struktura klastera na više nivoa. Međutim, ova metoda klasterovanja može biti sporija i zahtevati više računarskih resursa od drugih metoda klasterovanja. Hijerarhijsko klasterovanje takođe može biti manje efikasno za velike skupove podataka. Primer hijerarhijskog klasterovanja prikazan je na Slici 3.2.

Particione metode

Particione metode klasterovanja ([6]) su među najčešće korišćenim metodama u oblasti analize podataka i mašinskog učenja. Ove metode dele skup podataka u predefinisani broj grupa tj. klastera, na osnovu sličnosti među elementima skupa podataka.

Postoje različite particione metode klasterovanja, od kojih su neke k-sredina (eng. k-means), k-medoida (eng. k-medoids), fazi c-sredina (eng. fuzzy c-means) i spektralno klasterovanje (eng. spectral clustering). U k-sredina metodi, broj klastera se unapred definiše, a algoritam se koristi za iterativno preraspoređivanje elemenata u skupove koji minimizuju disperziju unutar klastera. Obično se koristi kada je broj klastera unapred poznat i kada su podaci neprekidni (numerčki). U k-medoida metodi, umesto srednje vrednosti, koristi se medijana klastera kao reprezentativni element



Slika 3.2: Primer hijerarhijskog klasterovanja

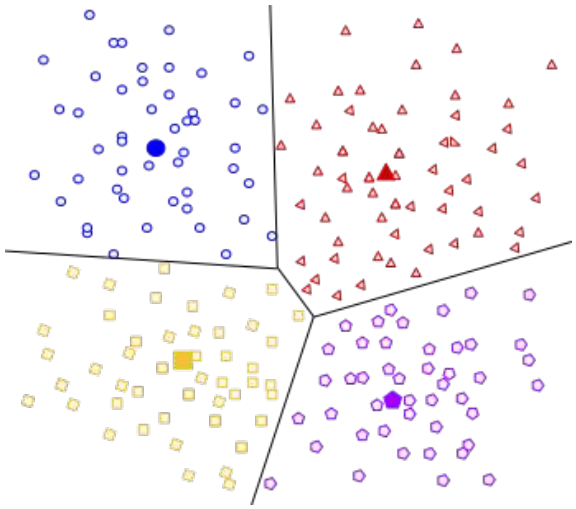
klastera. Takođe koristi se kada je broj klastera unapred poznat, ali je pogodniji kada se podaci ne mogu predstaviti kao kontinuirane vrednosti. Fazi c-sredina metoda generalizuje k-sredina metodu tako što elementi podataka pripadaju klasterima sa određenim stepenom pripadnosti, umesto da pripadaju jednom klasteru sa 100% sigurnošću. Ovaj algoritam se koristi kada se podaci ne mogu jasno klasifikovati u jedan klaster ili drugi, već postoji neka mešavina sličnosti za svaku tačku. Metoda spektralnog klasterovanja koristi matricu sličnosti između elemenata skupa podataka da bi identifikovala klastere i korisna je u situacijama kada je podacima teško pristupiti u njihovom originalnom obliku ili kada su podaci visoko-dimenzionalni.

Particione metode klasterovanja su popularane zbog svoje jednostavnosti, brzine i efikasnosti u radu sa velikim skupovima podataka. Međutim, ove metode mogu da se suoče sa izazovima u identifikovanju broja klastera i u rešavanju problema sa nehomogenim klasterima ili velikim izuzecima u podacima. Primer particionog klasterovanja prikazan je na Slici 3.3.

Metode zasnovane na gustini

Metode zasnovane na gustini ([7]) su vrsta metoda klasterovanja koja se fokusira na identifikaciju podgrupa u podacima na osnovu njihove gustine. Ove metode pretpostavljaju da klasteri predstavljaju područja u prostoru podataka koja su gušća u poređenju sa ostatkom prostora.

Najpoznatije metode zasnovane na gustini su DBSCAN (Density-Based Spatial



Slika 3.3: Primer particionog klasterovanja

Clustering of Applications with Noise) i OPTICS (Ordering Points To Identify the Clustering Structure). Obe metode se fokusiraju na identifikaciji podgrupa koje se razlikuju po gustini podataka.

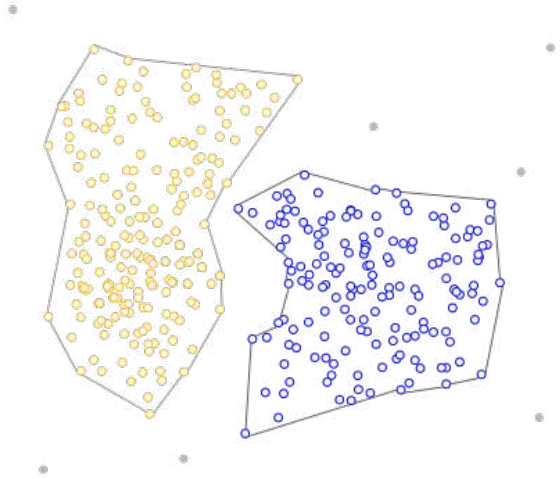
DBSCAN algoritam pronalazi klastere tako što definiše minimalnu gustinu tačaka koja se očekuje da postoji u klasteru. Na ovaj način, tačke u podacima koje nisu dovoljno gusto raspoređene neće biti uključene u klaster. DBSCAN algoritam takođe prepoznaje šum u podacima kao podgrupu koja ne pripada nijednom klasteru.

OPTICS algoritam takođe koristi gustinu podataka za identifikaciju klastera, ali se razlikuje od DBSCAN po tome što ne zahteva unapred definisanje minimalne gustine klastera. OPTICS algoritam gradi hijerarhijski stablo podataka, gde su klasteri sa većom gustinom predstavljeni granama koje se pružaju ka dole, dok se podgrupe sa manjom gustinom predstavljaju krajevima grana.

Metode zasnovane na gustini su efikasne u situacijama kada su podaci raspoređeni u različitim gustinama i kada se klasteri međusobno preklapaju ili imaju različite oblike. Međutim, ove metode mogu biti manje efikasne ako su podaci raspoređeni u potpuno homogene grupe ili ako se klasteri nalaze na velikoj udaljenosti jedan od drugog. Primer metoda zasnovanih na gustini prikazan je na Slici 3.4.

Metode zasnovane na raspodeli

Metode zasnovane na raspodeli ([6]) su vrsta metoda klasterovanja koja se fokusira na identifikaciju podgrupa u podacima na osnovu statističke raspodele. Ove metode pretpostavljaju da podaci u svakom klasteru dolaze iz određene raspode-



Slika 3.4: Primer klasterovanja zasnovanog na gustini

le, poput normalne raspodele. Ove metode su parametarske, za razliku od metoda zasnovanih na gustini gde se nije pretpostavljalo iz koje familije raspodela su podaci.

Najpoznatije metode zasnovane na raspodeli su GMM (Gaussian Mixture Models) i EM (Expectation-Maximization) algoritmi. Obe metode se fokusiraju na pronalaženje skrivenih raspodela koje se mogu koristiti za klasterovanje.

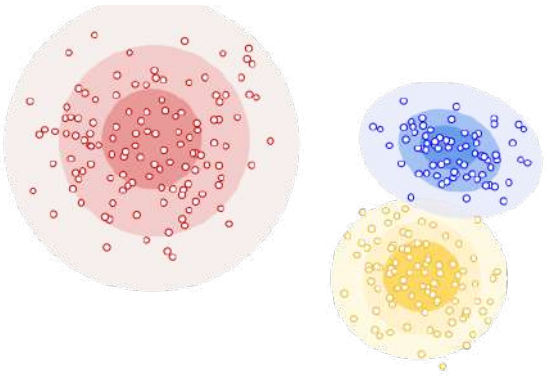
GMM algoritam koristi kombinaciju više Gausovih raspodela kako bi modelovao podatke. GMM koristi EM algoritam kako bi procenio parametre raspodele i odredio verovatnoću da svaka tačka u podacima pripada određenoj raspodeli. Na ovaj način, GMM može identifikovati podgrupe u podacima na osnovu sličnosti u raspodelama.

EM algoritam takođe se može koristiti za klasterovanje zasnovano na raspodeli kada je poznato da podaci dolaze iz određene raspodele. Algoritam radi iterativno, procenjujući parametre raspodele i koristeći ih za određivanje kojoj raspodeli pripada svaka tačka u podacima.

Metode zasnovane na raspodeli su efikasne u situacijama kada su podaci raspoređeni u skupine koje dolaze iz različitih statističkih raspodela. Međutim, ove metode mogu biti manje efikasne ako se podaci u svakom klasteru ne mogu modelovati kao raspodela ili ako klasteri imaju različite oblike i veličine. Primer metoda zasnovanih na gustini prikazan je na Slici 3.5.

Metode zasnovane na mreži

Metode zasnovane na mreži ([7]) su vrsta metoda klasterovanja koja se fokusira na grupisanje podataka na osnovu njihove lokacije u prostoru. Ove metode koriste



Slika 3.5: Primer klasterovanja zasnovanog na raspodeli

podelu prostora na pravilne mreže, a zatim identifikuju klasterovanje na osnovu raspodele tačaka u svakoj ćeliji mreže.

Najpoznatije metode zasnovane na mreži su DBSCAN (Density-Based Spatial Clustering of Applications with Noise) i STING (Statistical Information Grid-based Clustering). Iako je DBSCAN zasnovan na gustini podataka, može se koristiti sa mrežom za efikasnu implementaciju. Pomoću indeksne strukture poput R-drva ili Kd-drva, mogu se brzo pretraživati oblasti gustine i ubrzati proces klasterovanja.

STING koristi statističke informacije o podacima kako bi identifikovao klastere u svakoj ćeliji mreže. Metoda koristi informacije o srednjoj vrednosti, disperziji i korelaciji tačaka u svakoj ćeliji mreže kako bi izračunala statistički značajne klastere.

Metode zasnovane na mreži su efikasne za klasterovanje velikih skupova podataka koji su raspoređeni u pravilnom prostoru. Međutim, ove metode mogu biti manje efikasne ako su podaci raspoređeni u neregularnom ili kompleksnom prostoru ili ako se klasteri razlikuju po veličini ili obliku.

3.2 K-sredina

K-sredina klasterovanje je jedna od najčešće korišćenih tehnika klasterovanja u mašinskom učenju. Ono grupiše n-dimenzionalne podatke u k klastera, gde je k unapred zadati broj klastera.
K-sredina algoritam radi na sledeći način:

1. Unapred se zadaje broj klastera k.

2. Nasumično se inicijalizuje k tačaka u prostoru podataka kao početne centroide klastera.

3. Svaki podatak se dodeljuje klasteru na osnovu toga kojoj centroidi je najbliži. Kako bi se odredilo kojoj centroidi pripada svaki podatak, prvo se računa udaljenost između svakog podatka i svake centroide. Najčešće se koristi euklidska udaljenost, koja se računa formulom:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \tag{3.1}$$

gde su x i y n-dimenzionalni podaci, a $(x_1, x_2, ..., x_n)$ i $(y_1, y_2, ..., y_n)$ vektorske reprezentacije tih podataka. Pored euklidske norme, mogu se koristiti i neke alternativne metrike u zavisnosti od tipa podataka, kao što je kosinusna sličnost. Kosinusna sličnost je metrika koja meri kosinus ugla između dva vektora i daje vrednost od -1 do 1. Veće vrednosti ukazuju na veću sličnost, a negativne vrednosti ukazuju na suprotnost. Kosinusna sličnost je posebno korisna kada radimo sa tekstualnim podacima ([8]).

4. Za svaki klaster se računa nova centroida.
Centroida klastera se računa kao prosečna tačka svih podataka koji pripadaju tom klasteru. Ako imamo k klastera, i-ta centroida (c_i) se računa kao:

$$c_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_j, \tag{3.2}$$

gde je n_i broj tačaka koji pripadaju i-tom klasteru, x_j je tačka i-tog klastera, a $\sum_{j=1}^{n} x_j$ je zbir svih koordinata j-te tačke.

5. Algoritam ponavlja korake 3 i 4 dok se centroida klastera ne stabilizuje.

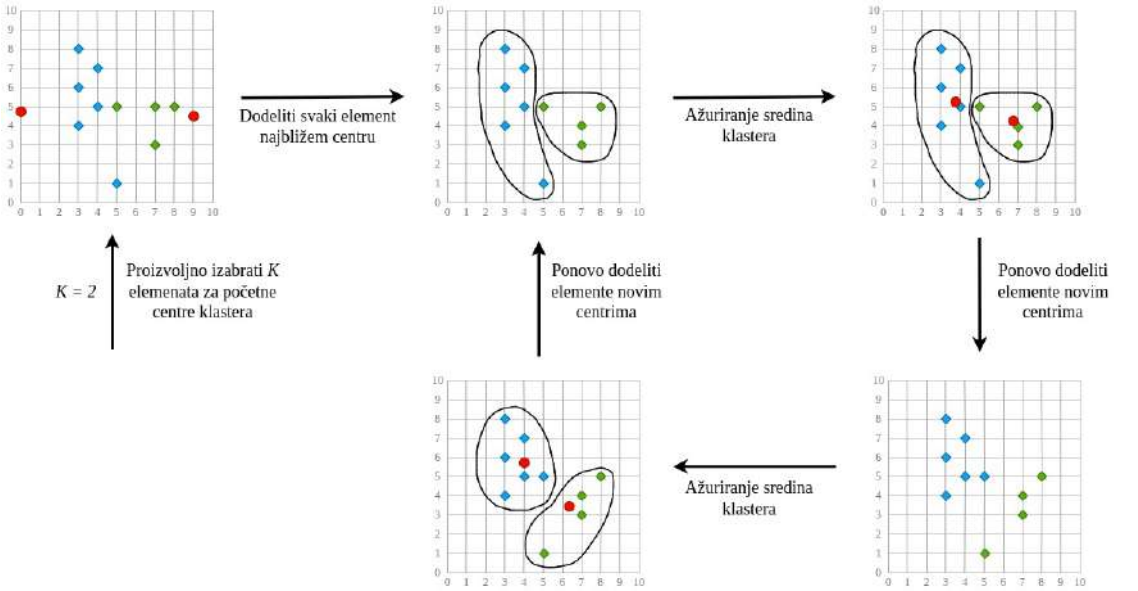
Grafički prikaz toka klasterizacije prikazan je na Slici 3.6.

Konačni rezultat je grupisanje podataka u k klastera. Ovaj algoritam je iterativni, što znači da se može ponoviti više puta kako bi se osiguralo postizanje najboljeg rezultata. K-sredina klasterovanje se često koristi za analizu velikih skupova podataka, kao što su slike, zvuk i tekst.

Ovaj algoritam minimizuje sumu kvadrata rastojanja:

$$\sum_{i=1}^{k}\sum_{x \epsilon C_i} d(x, c_i)^2$$

po c_i, gde je d euklidsko rastojanje. Kako je algoritam zasnovan na minimizaciji euklidskog rastojanja, on teži pronalaženju klastera u obliku lopte. Zbog kvadrata



Slika 3.6: Tok klasterizacije kod k-sredina

rastojanja, algoritam je osetljiv na podatke koji značajno odudaraju od ostalih tj. autlajere i u tom slučaju veća rastojanja utiču na ukupnu grešku nesrazmerno u odnosu na ostala rastojanja, pa samim tim i nesrazmerno utiču na lokaciju centroide. Takođe, u slučaju kada gustina tačaka ne varira drastično i rastojanja među klasterima nisu velika, algoritam će izabrati klastere sa sličnim brojem tačaka u njima, jer u suprotnom veliki klaster bi sadržao i tačke daleko od centroide koje bi značajno povećavale sumu kvadrata rastojanja. Kako ova funkcija ne mora imati globalni minumum, to ni klasterovanje u odnosu na datu sumu kvadrata rastojanja nije jedinstveno, tačnije moguće je da postoji veći broj klasterovanja jednakog kvaliteta. Međutim, mogu postojati i lokalni minimumi slabijeg kvaliteta od globalnog koje algoritam može naći, što nije dobro. Ovaj problem se ublažava tako što se klasterovanje pokreće veći broj puta sa od različitih inicijalnih tačaka i za rezultat se uzima klasterovanje koje ima najmanju vrednosti sume kvadrata rastojanja.

Mana ovog algoritma su različiti rezultati klasterovanja za različit broj klastera, odnosno za razločito k. Takođe, različite vrednosti početnih centroida će rezultirati drugačijim klasterom. Performanse klasterovanja u velikoj meri zavise od ovih parametara, pa postoji mnoštvo metoda za određivanje ovih parametara.

3.2.1 Određivanje broja klastera

Postoje različite tehnike koje se mogu koristiti za određivanje optimalne vrednosti parametra k u algoritmu k-sredina.

Neke od tih tehnika su metoda lakta (eng. elbow method), metoda siluete (eng. silhouette method) koje su najpoznatije, međutim postoje i druge tehnike za određivanje optimalne vrednosti parametra k, kao što su metoda „čuperka" iliti Kalinski-Herabeš (eng. Calinski-Harabasz index), Dejvis-Buldin metoda (eng. Davies-Bouldin index) i druge. Treba imati na umu da nijedna od ovih tehnika nije savršena i da je važno da se koriste u kombinaciji s drugim tehnikama i da se uzmu u obzir i drugi kriterijumi pri određivanju optimalne vrednosti.

Metoda lakta

Ovo pravilo se sastoji u tome da se za različit broj klastera k vrši klasterovanje, i za svako k se računa suma kvadrata grešaka u klasteru k tj. WCSS (Within-Cluster Sum of Squares).

$$WCSS = \sum_{k=1}^{K} WCSS(k), \tag{3.3}$$

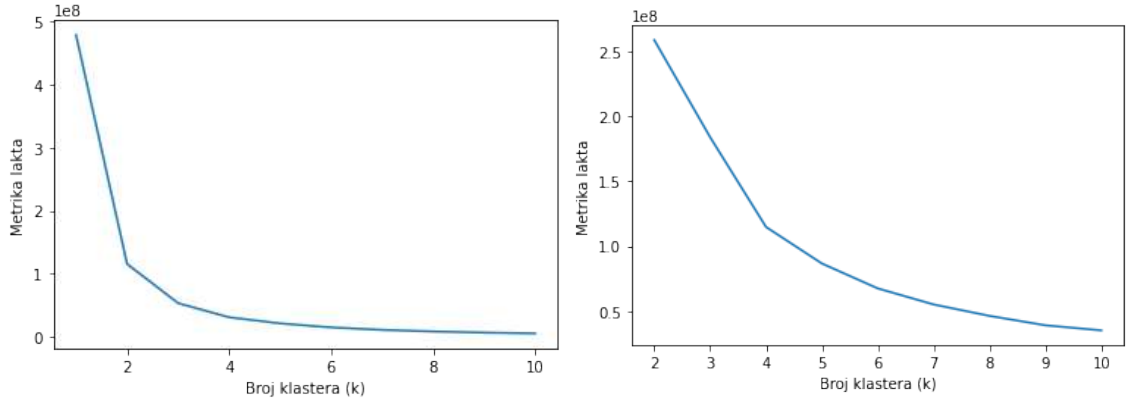
gde je

$$WCSS(k) = \sum_{x_i \in k} (x_i - c_k)^2 \tag{3.4}$$

$WCSS$ za klaster k, a c_k centroida za klaster k.

Na osnovu grafika zavisnosti sume kvadrata rastojanja i broja klastera (Slika 3.7), određuje se optimalno k koje odgovara tački nagle promene brzine opadanja grafika, tj. u tački nakon koje suma kvadrata grešaka prestaje drastično da opada. Što je promena sume kvadrata grešaka u susednim klasterima manja, klasteri su homogeniji.

U slučajevima kada nemamo uvek jasno klasterisane podatke, lakat neće biti "oštar", pa ova metoda postaje nepouzdana.



Slika 3.7: Pravilo lakta

Metoda siluete

Koeficijent siluete je metrika koja se koristi za evaluaciju kvaliteta klastera. On se izračunava za svaku pojedinačnu instancu u skupu podataka i predstavlja meru koliko je ta instanca „dobro" smeštena u svoj klaster u odnosu na ostale klastere. Visok koeficijent siluete znači da je instanca dobro smeštena u svoj klaster, dok niski koeficijent siluete ukazuje da instanca nije dobro smeštena u svoj klaster i da bi mogla bolje da se uklopi u neki drugi klaster. Koeficijent siluete se kreće od -1 do 1. Visoke vrednosti (bliže 1) ukazuju na dobro definisane klastere, dok niske vrednosti (bliže -1) ukazuju na loše definisane klastere. Vrednost 0 označava da je instanca smeštena na granici između dva klastera. Glavne odlike klasterovanja su: kohezija (sve tačke jednog kastera trebaju biti što sličnije) i separacija (tačke različitih klastera trebaju biti što različtije). Pomoću koeficijenta siluete merimo ove dve odlike zajedno i na taj način vidimo koliko su naši klasteri dobri.

Informaciju o koheziji dobijamo računanjem parametara a ($a(i)$ je prosečna udaljenost između i-te instance i svih ostalih tačaka podataka u klasteru kome pripada i-ta instanca), dok separaciju predstavlja parametar b ($b(i)$ je minimalno prosečno rastojanje od i-te instance do svih klastera kojima i-ta instanca ne pripada):

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \epsilon C_I, i \neq j} d(i, j), \tag{3.5}$$

gde je $|C_I|$ kardinalnost klastera C_I, a $d(i, j)$, rastojanje između tačaka i i j klastera C_I,

$$b(i) = \min_{I \neq J} \frac{1}{|C_J|} \sum_{j \epsilon C_J} d(i, j). \tag{3.6}$$

Koeficijent siluete $s(i)$ za i-tu instancu se računa formulom:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}. \tag{3.7}$$

Koeficijent siluete za ceo klaster dobijamo kao srednju vrednost svih koeficijenata siluete $s(i)$ za tačke pridružene tom klasteru.

Kalinski-Herabeš indeks

Metoda „čuperka" je još jedna tehnika koja se koristi za određivanje optimalne vrednosti parametra k u algoritmu k-sredina. Ova metoda se zasniva na računanju razlike između unutrašnje disperzije klastera (eng. within-cluster variance) i međusobne disperzije klastera (eng. between-cluster variance). Optimalna vrednost parametra k je ona koja daje najveću razliku između ove dve disperzije.

Za izračunavanje metode čuperka se definišemo:

- C je skup podataka koji se klasterizuje;
- k je broj klastera;
- n je ukupan broj instanci u skupu podataka C;
- $x(i)$ je i-ta instanca u skupu podataka C;
- μ je prosečna vrednost za sve instance u skupu podataka C;
- $\mu(j)$ je prosečna vrednost za sve instance u j-tom klasteru.

Čuperkova metrika se izračunava kao:

$$CH = \frac{n-k}{k-1}\frac{S(k)}{S(tot)}, \tag{3.8}$$

gde je:

$$S(k) = \sum_j (\mu(j) - \mu)^2 \; za \; sve \; klastere \; j, \tag{3.9}$$

$$S(tot) = \sum_i (x(i) - \mu)^2 \; za \; sve \; instance \; i. \tag{3.10}$$

Visoke vrednosti ukazuju na dobro definisane klastere, dok niske vrednosti ukazuju na loše definisane klastere. Međutim, treba imati na umu da ova metrika može biti subjektivna i da je važno da se koristi u kombinaciji s drugim tehnikama i da se uzmu u obzir i drugi kriterijumi pri određivanju optimalne vrednosti parametra k.

3.3 Hijerarhijsko klasterovanje

Hijerarhijsko klasterovanje je jedna od tehnika klasterovanja u kojoj se elementi grupišu u hijerarhijsku strukturu. Kao što ime sugeriše, ove metode proizvode hijerarhijske reprezentacije u kojima se klasteri na svakom nivou hijerarhije stvaraju spajanjem klastera na sledećem, nižem nivou. Na najnižem nivou, svaki klaster sadrži jednu tačku. Na najvišem nivou postoji samo jedan klaster koji sadrži sve podatke.

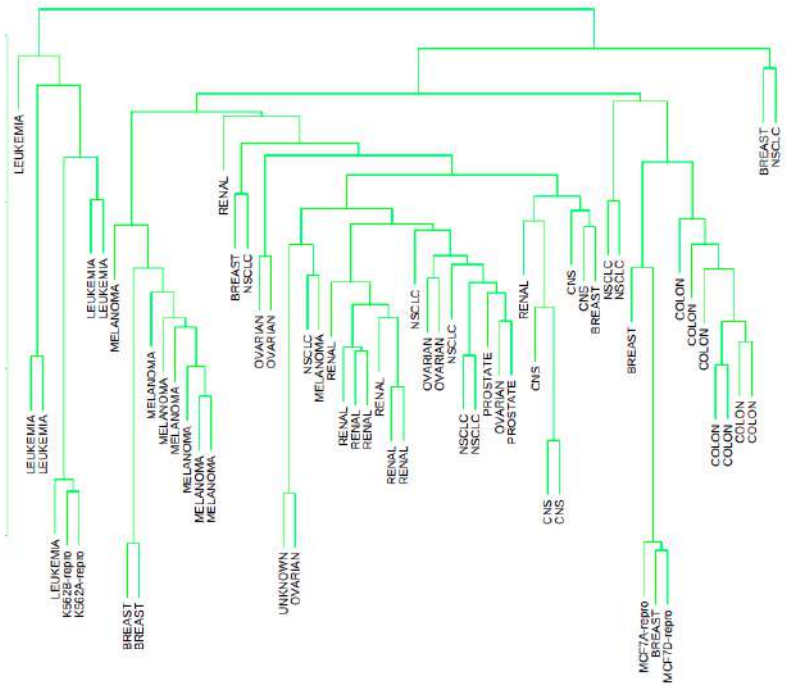
Strategije za hijerarhijsko klasterovanje se dele na dve osnovne paradigme: **hijerarhijsko klasterovanje spajanjem** (od dna prema gore) i **hijerarhijsko klasterovanje deljenjem** (od vrha prema dnu). Klasterovanje spajanjem počinje od dna i na svakom nivou rekurzivno spaja odabrani par klastera u jedan klaster. To stvara grupisanje na sledećem višem nivou sa jednim manje klasterom. Par koji se bira za spajanje čine dva klastera sa najmanjom međuklasterskom različitošću. Klasterovanje deljenjem počinje od vrha i na svakom nivou rekurzivno deli jedan od postojećih klastera na dva nova klastera. Par koji će se deliti se bira tako da se dobiju dva nova klastera sa najvećom međuklasterskom različitošću. U obe paradigme postoji $N - 1$ nivo u hijerarhiji gde je N ukupan broj podataka. Na korisniku je da odluči koji nivo (ako postoji) zaista predstavlja „prirodno" grupisanje u smislu da su tačke unutar svakog klastera dovoljno slične međusobno u odnosu na tačke dodeljene različitim klasterima na tom nivou.

Rekurzivno binarno deljenje/spajanje može se prikazati pomoću binarnog stabla sa korenom. Čvorovi stabla predstavljaju klastere. Korenski čvor predstavlja ceo skup podataka. N terminalnih čvorova predstavlja svaku pojedinačnu tačku (klaster sa jednim elementom). Svaki čvor koji nije terminalni ima dva čvora-deteta. Za klasterovanje deljenjem, ove dva deteta predstavljaju dva klastera koji nastaju iz deljenja roditelja; za klasterovanje spajanjem, deca predstavljaju dva klastera koja su spojena kako bi se formirao roditelj.

3.3.1 Dendrogram

Sve metode klasterovanja spajanjem i neke metode klasterovanja deljenjem (kada se posmatraju od dna prema vrhu) poseduju svojstvo monotonosti. To znači da je različitost između spojenih klastera monotono povećava s nivoom spajanja. Stoga se binarno stablo može prikazati na način da je visina svakog čvora proporcionalna vrednosti međuklasterske različitosti između njegove dva deteta. Terminalni čvorovi

koji predstavljaju pojedinačne tačke prikazuju se na visini 0. Ovakav tip grafičkog prikaza naziva se **dendrogram**. Dendrogram pruža visoko interpretabilan i potpun opis hijerarhijskog klasterovanja u grafičkom formatu. Primer dendrograma prikazan je na Slici 3.8.



Slika 3.8: Dendrogram iz hijerarhijskog klasterovanja spajanjem sa prosečnom vezom sa podacima mikromreža tumora ([5])

Horizontalno sečenje dendrograma na određenoj visini deli podatke na odvojene klastere koji su predstavljeni presečenim vertikalnim linijama. To su klasteri koji bi bili formirani prekidanjem postupka kada optimalna međuklasterska različitost premašuje odgovarajuću unapred zadatu vrednost. Dobri kandidati za prirodne klastere su grupe koje se spajaju pri visokim vrednostima, u odnosu na vrednosti spajanja podklastera sadržanih unutar njih niže u stablu.

3.3.2 Hijerarhijsko klasterovanje spajanjem

Algoritmi klasterovanja spajanjem počinju sa svakom tačkom koja predstavlja jednočlani klaster. Na svakom od $N - 1$ koraka, najbliža dva (najmanje različita) klastera se spajaju u jedan klaster, što rezultira jednim manje klasterom na sledećem,

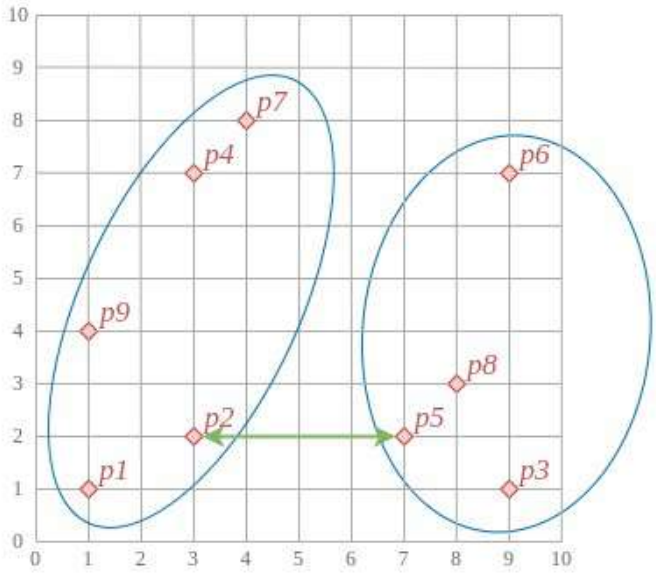
višem, nivou. Stoga je potrebno definisati meru različitosti između dva klastera (grupe posmatranja).

Neka G i H predstavljaju dve takve grupe. Različitost $d(G, H)$ između G i H se izračunava na osnovu skupa različitosti parova posmatranja $d_{i,j}$ gde jedan član para, i, pripada grupi G, a drugi, j, grupi H. U hijerarhijskom klasterovanju spajanjem najčešće se koriste metode poput jednostrukog povezivanja, potpunog povezivanja i prosečnog povezivanja. Ove metode se razlikuju u načinu na koji se računa sličnost između klastera.

Kod hijerarhijskog klasterovanja spajanjem sa jednostrukim povezivanjem (eng. Single linkage, SL, Slika 3.9) međuklasterska različitost je predstavljena kao različitost najbližeg (najmanje različitog) para:

$$d_{SL}(G, H) = \min_{i \epsilon G, j \epsilon H} d_{ij}. \tag{3.11}$$

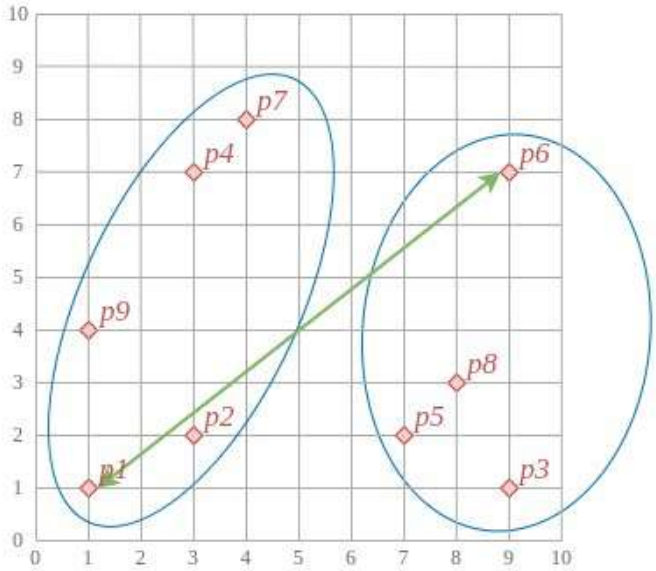
Ovo se često naziva tehnika najbližeg suseda.



Slika 3.9: Klasterovanje sa jednostrukim povezivanjem

Hijerarhijsko klasterovanje spajanjem sa potpunim povezivanjem (eng. Complete linkage, CL, Slika 3.10) uzima međuklastersku različitost kao različitost najudaljenijeg (najrazličitijeg) para:

$$d_{CL}(G, H) = \max_{i \epsilon G, j \epsilon H} d_{ij}. \tag{3.12}$$

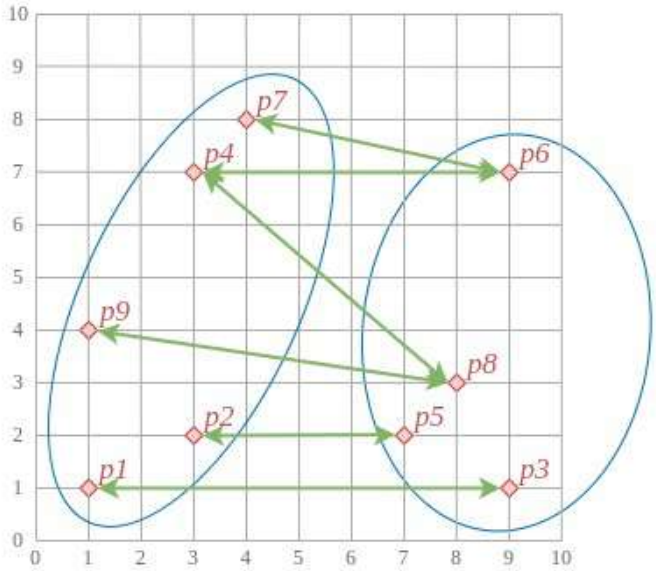


Slika 3.10: Klasterovanje sa potpunim povezivanjem

Hijerarhijsko klasterovanje spajanjem sa prosečnim povezivanje (eng. Average linkage, AL,Slika 3.11) koristi prosečnu različitost između klastera:

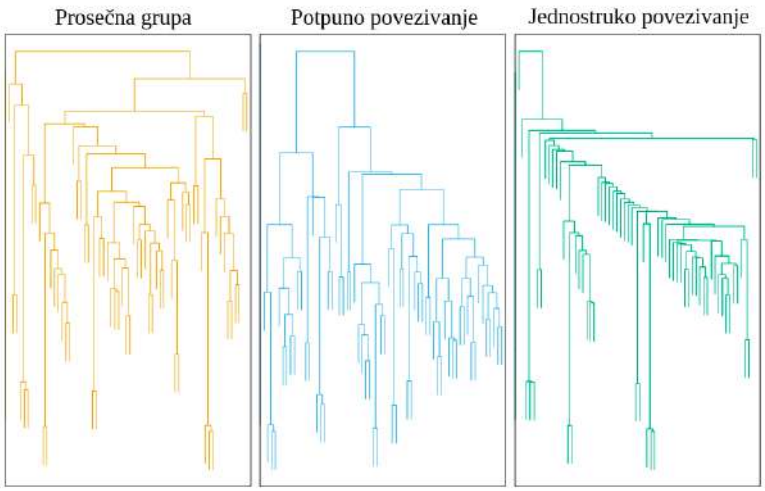
$$d_{AL}(G,H) = \frac{1}{N_G N_H} \sum_{i \epsilon G} \sum_{J \epsilon H} d_{ij}, \tag{3.13}$$

gde su N_G i N_H predstavljaju broj tačaka u svakoj grupi.



Slika 3.11: Klasterovanje sa prosečnom grupom

Na Slici 3.12 prikazani su dendrogrami dobijenim različitim povezivanjima.



Slika 3.12: Dendrogrami iz hijerarhijskog grupisanja spajanjem podataka mikromreža tumora ([5])

Još jedna metoda koja se često koristi u klasterovanju spajanjem je Vardovo povezivanje (eng. Ward linkage) [9]. Ova metoda određuje međuklastersku različitost na osnovu toga koliko će se povećati unutar-klasterska disperzija kada se te dve grupe spoje. Ideja iza Vardovog povezivanja je da se minimizuje disperzija unutar klastera nakon spajanja, što se smatra merom kompaktnosti klastera. Spajanjem grupa sa manjom disperzijom unutar njih, stvaraju se klasteri koji su homogeniji u smislu sličnosti između posmatranja. Međuklasterska različitost je:

$$d_{WL}(G,H) = \frac{N_G \cdot N_H \cdot (s_G^2 + s_H^2)}{N_G + N_H}, \tag{3.14}$$

gde su s_G^2 i s_H^2 disperzije za klastere G i H:

$$s^2 = \frac{\sum_{k=1}^{N}(x_k - \overline{x})^2}{N},\ \overline{x} = \frac{1}{N}\sum_{k=1}^{N} x_k. \tag{3.15}$$

3.3.3 Hijerarhijsko klasterovanje deljenjem

Algoritmi klasterovanja deljenjem počinju sa celokupnim skupom podataka kao jednim klasterom, i rekurzivno dele jedan od postojećih klastera na dva deteta klastera u svakoj iteraciji od vrha prema dnu. Ovaj pristup nije toliko detaljno proučavan kao metode spajanjem u literaturi o klasterovanju. U kontekstu klasterovanja, potencijalna prednost metoda deljenjem u odnosu na metode spajanjem može se javiti kada je interes fokusiran na particionisanje podataka u relativno mali broj klastera.

Postoji nekoliko pristupa koji se mogu koristiti u klasterovanju deljenjem, kao što su k-sredina, k-medoida ili neki drugi algoritmi za particionisanje podataka. U svakoj iteraciji, jedan od postojećih klastera se bira za deljenje, a zatim se koristi odabrani algoritam za particionisanje kako bi se podelio u dva nova klastera, podešavanjem parametra $k = 2$. Međutim, ovakav pristup bi zavisio od početne konfiguracije koja je određena u svakom koraku. Takođe, ne bi nužno proizveo sekvencu podela koja poseduje monotonost potrebnu za reprezentaciju dendrograma.

Algoritam koji izbegava ove probleme je predložen od strane Meknatn Smita u [10]. On počinje tako što sve tačke smešta u jedan klaster G. Zatim bira onu tačku čija je prosečna različitost u odnosu na sve ostale tačke najveća. Ta tačka čini prvi član drugog klastera H. U svakom sledećem koraku, da bi odlučio koju tačku iz G treba premeštati, algoritam radi sledeće: računa prosečnu različitost između svih tačaka u G i svih tačaka u H, a zatim računa prosečnu udaljenost između svih tačaka u G koje su ostale u G. Konačno, algoritam bira tačku u klasteru G koja ima najveću razliku izmedju gore navedenih vrednosti. Proces se nastavlja sve dok odgovarajuća razlika u proseku ne postane negativna. Drugim rečima, više nema tačaka u G koja su, u proseku, bliža onima u H. Rezultat je podela originalnog klastera na dva deteta klastera, posmatranja premeštena u H, i ona koja ostaju u G. Ova dva klastera predstavljaju drugi nivo hijerarhije. Svaki sledeći nivo se dobija primenom ovog postupka podela na jedan od klastera sa prethodnog nivoa. Kofmen i Rusju [11] predlažu odabir klastera na svakom nivou sa najvećim prečnikom za podelu. Alternativa bi bila odabrati onaj sa najvećom prosečnom različitošću među članovima. Rekurzivno deljenje se nastavlja sve dok svi klasteri postanu samostalni ili svi članovi svakog klastera imaju nula različitosti jedni od drugih.

3.4 Klasterovanje pomeranjem sredine

Klasterovanje pomeranjem sredine (eng. Mean-Shift) je algoritam za grupisanje zasnovan na gustini koji se može koristiti za identifikaciju klastera u skupu podataka. Posebno je korisno za skupove podataka gde klasteri imaju proizvoljne oblike i nisu dobro razdvojeni linearnim granicama.

Osnovna ideja iza klasterovanja pomeranjem sredine je da se svaka tačka podataka pomeri prema modi (tj. najvećoj gustini) raspodele tačaka unutar određenog radijusa. Algoritam iterativno izvodi ove pomake sve dok tačke ne konvergiraju lokalnom maksimumu funkcije gustine. Ovi lokalni maksimumi predstavljaju klastere

u podacima.

Algoritam se sastoji od sledećih koraka:

1. Inicijalizacija: Biramo početne proizvoljne tačke u skupu podataka, kao inicijalne centre klastera.

2. Pravljenje prozora: Odaberemo okolinu određenog radijusa, za svaku od tačaka.

3. Izračunavanje središta mase: Računamo središte mase tačaka unutar prozora. Ovo je prosečna vrednost svih tačaka unutar prozora, ponderisana udaljenošću od centra prozora (tačke bliže centru dobijaju veću težinu).

4. Pomeranje prozora: Pomeramo prozor tako da je centar na središtu mase koje smo izračunali.

5. Iteracija: Ponavljamo korake 3 i 4 dok prozor ne prestane da se pomera tj. dok se ne postigne konvergencija (dok središte mase unutar prozora ne postane centar prozora) ili dok se ne dostigne maksimalan broj iteracija.

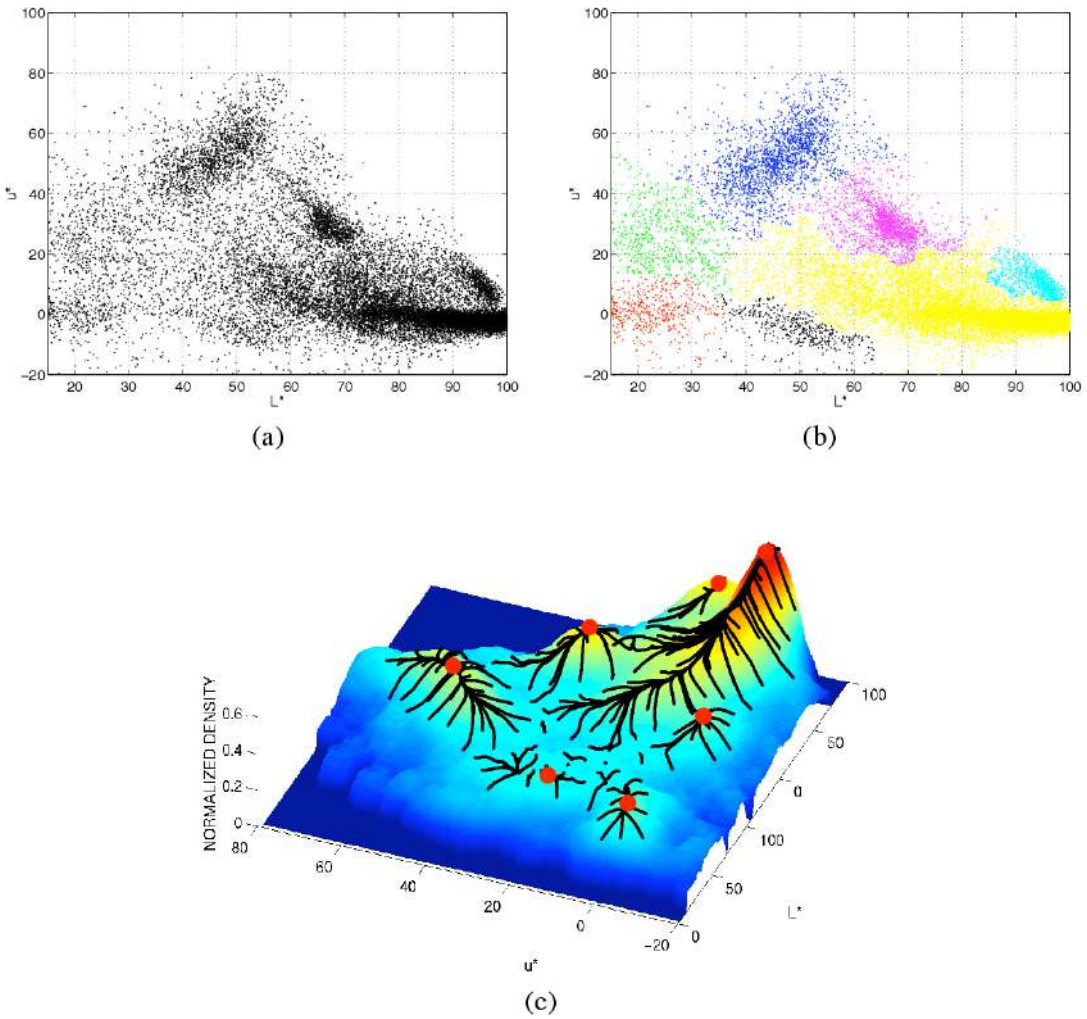
Na kraju ovog postupka, različiti prozori će se stabilizovati oko različitih centara klastera. Tada možemo dodeliti svaku tačku podataka klasteru centra najbližeg kojem je ta tačka.

Primena ovog algoritma nad podacima prikazana je na Slici 3.13.

Izbor radijusa u algoritmu pomeranjem sredine zavisi od prirode podataka i ciljeva analize. Postoje različite metode za određivanje radijusa ([12]):

- Prvi pristup ima statističku motivaciju. Optimalni radijus definisan je kao radijus koji postiže najbolju kompromis između pristrasnosti i disperzije ocene, odnosno, minimizira AMISE[1]. U višedimenzionalnom slučaju, rezultirajuća formula za radijus ([13]) ima malo praktične upotrebe, jer zavisi od Laplasijana nepoznate gustine koja se procenjuje. Za jednodimenzionalni slučaj, pouzdana metoda za selekciju radijusa je pravilo uvrštene ocene ([14]) koje je dokazano da je superiorno u odnosu na druge metode (kao što je npr. metoda najmanjih kvadrata za unakrsnu validaciju). Jedina pretpostavka je glatkoća osnovne gustine.

[1]AMISE (Asymptotic Mean Integrated Squared Error) - asimptotskaintegralna srednje kvadratna greška je funkcija kriterijuma optimalnosti koja meri performanse ocene gustine jezgra



Slika 3.13: Primer 2D analize prostora obeležja. (a) Dvodimenzionalni skup podataka od 110400 tačaka. (b) Razlaganje dobijeno pokretanjem 159 procedura algoritma pomeranjem sredine sa različitim inicijalizacijama. (Različite boje predstavljaju različite klastere)(c) Trajektorije procedure nacrtane preko Epanješnjikovljeve ocene gustine izračunate za iste podatke. Vrhovi zadržani za konačnu klasifikaciju su označeni crvenim tačkama. ([12])

- Drugi pristup izboru radijusa vezan je za stabilnost klasterovanja. Radijus se određuje kao centralna tačka najšireg raspona u kojem se za dati skup podataka postiže isti broj klastera.
- Što se tiče treće tehnike, najbolji radijus maksimizuje ciljnu funkciju koja izražava kvalitet klasterovanja. Ciljna funkcija obično upoređuje među- i unutarklastersku varijabilnost.

3.4.1 Jezgro

Pojam jezgra odnosno kernela u mašinskom učenju, se razlikuje od konteksta u kom se koristi. U kontekstu PCA (Principal Component Analysis) kernel se koristi za nelinearnu transformaciju podataka, metodi potpornih vektora kernel omogućava preslikavanje podataka za bolju linearnu separaciju, dok se u algoritmu pomeranjem sredine koristi za definisanje smera pomeraja tačnije smera povećanja gustine podataka.

Formalno, u kontekstu ocenjivanja gustine, jezgro je nenegativna, integrabilna funkcija K. U većini slučajeva, poželjno je da funkcija K zadovolji dva dodatna uslova:

- $\int_{-\infty}^{\infty} K(u)du = 1$ što omogućava ocenjivanje gustine jezgrom;
- $K(-u) = K(u)$ za svako u.

Algoritam pomeranjem sredine se oslanja na korišćenje jezgra kako bi se formirala okolina ili „prozor" oko svake tačke podataka. Kroz iterativni postupak, svaka tačka se pomera prema središtu mase okoline, određenom pomoću jezgra.

Različita jezgra (Slika 3.14) mogu se koristiti u ovom algoritmu, uključujući:

1. Ravno jezgro (eng. flat kernel): Ovo je najjednostavniji oblik jezgra. Sve tačke unutar prozora su jednako važne za izračunavanje središta mase.
2. Gausovo jezgro: Ovo jezgro daje veću težinu tačkama koje su bliže centru prozora. Ovo se postiže primenom Gausove funkcije (normalne raspodele) na udaljenosti tačaka od centra.
3. Epanješnjikovljevo jezgro: Ovo jezgro je efikasnije od Gausovog i često daje slične rezultate. Takođe daje veću težinu tačkama koje su bliže centru, ali na način koji je efikasniji za računanje od Gausovog jezgra.

Ravno jezgro je najjednostavnije u praksi, Gausovo jezgro se često koristi zbog svoje efikasnosti i performansi. Međutim, druga jezgra mogu biti korisna u određenim situacijama, zavisno od prirode podataka i specifičnih zahteva problema.

Ravno jezgro se definiše kao:

$$K(t) = 1 \cdot I(|t| \leq \lambda), \tag{3.16}$$

gde je λ konstanta koja kontroliše širinu jezgra.

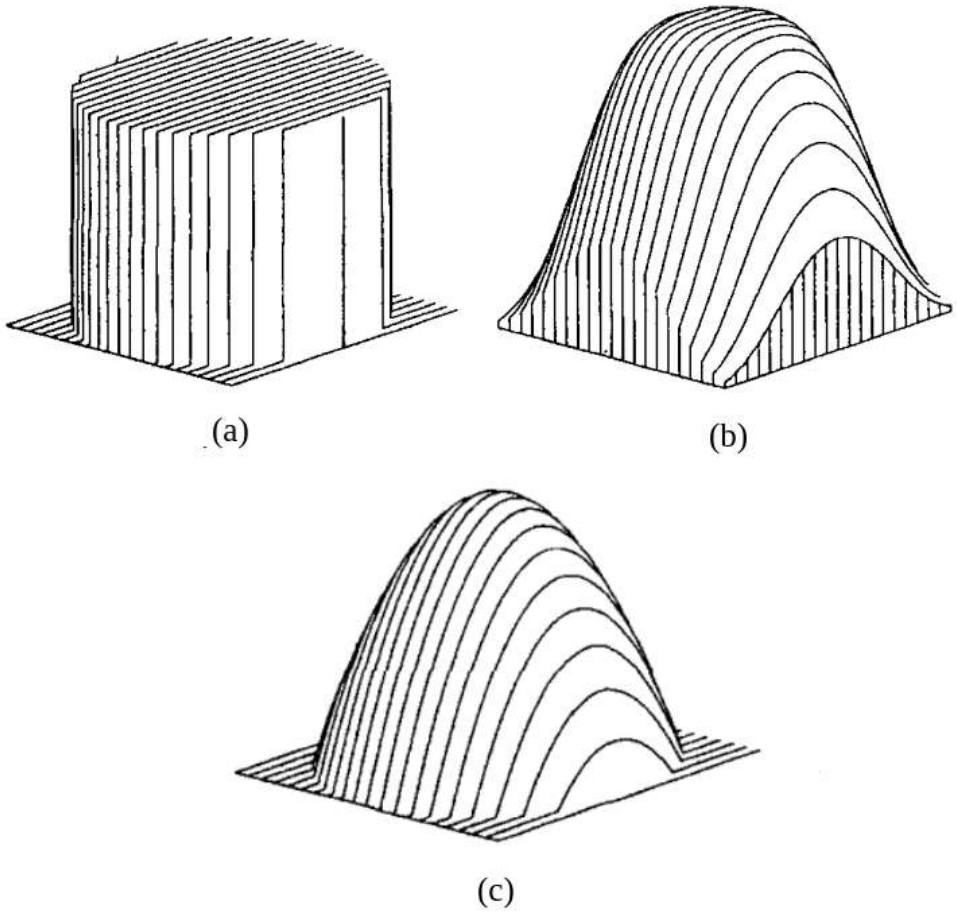
Gausovo jezgro se često koristi u ovom algoritmu zbog svoje sposobnosti da daje veću težinu tačkama koje su bliže centru „prozora" ili okoline koju razmatramo. Ovo omogućava efikasno i efektivno klasterovanje podataka.

Gausovo jezgro, često se definiše kao:

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}. \tag{3.17}$$

Epanješnjikovljevo jezgro se definiše kao:

$$K(t) = \frac{3}{4}\left(1 - t^2\right) \cdot I(|t| \leq 1). \tag{3.18}$$



Slika 3.14: (a) Ravno jezgro, (b) Gausovo jezgro, (c) Epanješnjikovljevo jezgro ([15])

3.4.2 Ocena gustine

Neka je $\{\mathbf{x}_i\}_{i=1,\ldots,n}$ proizvoljan skup tačaka u d-dimenzionalnom euklidskom prostoru R^d. Višedimenzionalna ocena gustine jezgrom dobijena sa jezgrom $K(x)$ i radijusom prozora h, izračunata u tački x, je definisana kao:

$$\widehat{f}(\mathbf{x}) = \frac{1}{n\,h^d}\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right). \tag{3.19}$$

Optimalno jezgro koje daje minimalnu integralnu srednje kvadratnu grešku[2] je Epanješnjikovljevo jezgro, koji možemo predstaviti kao:

$$K_E(\mathbf{x}) = \frac{1}{2}c_d^{-1}(d+2)(1-\mathbf{x}^T\mathbf{x})I(\mathbf{x}^T\mathbf{x}\leq 1). \tag{3.20}$$

Upotreba diferencijabilnog jezgra omogućava definisanje ocene gradijenta gustine kao gradijenta ocene gustine jezgrom:

$$\widehat{\bigtriangledown} f(\mathbf{x}) = \bigtriangledown \widehat{f}(\mathbf{x}) = \frac{1}{n\,h^d}\sum_{i=1}^{n} \bigtriangledown K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right). \tag{3.21}$$

Za Epanješnjikovljevo jezgro, ocena gradijenta gustine postaje:

$$\widehat{\bigtriangledown} f(\mathbf{x}) = \frac{1}{n(h^d c_d)}\frac{d+2}{h^2}\sum_{\mathbf{x}_i \epsilon S_h(\mathbf{x})}[\mathbf{x}_i-\mathbf{x}] = \frac{n_{\mathbf{x}}}{n(h^d c_d)}\frac{d+2}{h^2}\left(\frac{1}{n_{\mathbf{x}}}\sum_{\mathbf{x}_i \epsilon S_h(\mathbf{x})}[\mathbf{x}_i-\mathbf{x}]\right), \tag{3.22}$$

gde je region $S_h(\mathbf{x})$ hipersfera poluprečnika h koja ima zapreminu $h^d c_d$, centriranu oko tačke $\mathbf{x}$ i sadrži $n_{\mathbf{x}}$ tačaka podataka. Poslednji član u jednačini 3.22:

$$M_h(\mathbf{x}) = \frac{1}{n_{\mathbf{x}}}\sum_{\mathbf{x}_i \epsilon S_h(\mathbf{x})}[\mathbf{x}_i-\mathbf{x}] = \frac{1}{n_{\mathbf{x}}}\sum_{\mathbf{x}_i \epsilon S_h(\mathbf{x})}\mathbf{x}_i-\mathbf{x} \tag{3.23}$$

se naziva pomerajem srednje vrednosti uzorka. Korišćenje jezgra drugačijeg od Epanješnjikovljevog rezultira izračunavanjem ponderisane srednje vrednosti. Vrednost $\frac{n_{\mathbf{x}}}{n(h^d c_d)}$ je ocena gustine jezgra $\widehat{f}(\mathbf{x})$ izračunata sa hipersferom S_h (ravno jezgro) i stoga ocenu gradijenta gustine Epanješnjikovljevog jezgra možemo zapisati kao:

$$\widehat{\bigtriangledown} f(\mathbf{x}) = \widehat{f}(\mathbf{x})\frac{d+2}{h^2}M_h(\mathbf{x}) \tag{3.24}$$

[2]MISE (Mean Integrated Squared Error) - integralna srednje kvadratna greška ([16]) je data formulom:

$$E\int \|f_n - f\|_2^2 = E\int (f_n(x)-f(x))^2\,dx,$$

gde je f nepoznata gustina, a f_n njena ocena.

odakle je

$$M_h(\mathbf{x}) = \frac{h^2}{d+2} \frac{\widehat{\bigtriangledown} f(\mathbf{x})}{\widehat{f}(\mathbf{x})}. \tag{3.25}$$

Ovaj izraz pokazuje da se ocena normalizovanog gradijenta može dobiti računanjem uzorka pomaka srednje vrednosti u uniformnom jezgru centriranom na $\mathbf{x}$. Vektor pomaka srednje vrednosti ima smer gradijenta ocene gustine u tački $\mathbf{x}$ kada se ova ocena dobija Epanješnjikovljevim jezgrom.

Pošto vektor pomaka srednje vrednosti uvek pokazuje u smeru maksimalnog porasta gustine, on može definisati putanju koja vodi do lokalnog maksimuma gustine, odnosno do mode gustine. Postupak pomaka srednje vrednosti koji se dobija sledećim uzastopnim koracima:

- računanje vektora pomaka srednje vrednosti $M_h(\mathbf{x})$
- translacija prozora $S_h(\mathbf{x})$ pomoću $M_h(\mathbf{x})$

konvergira.

Označimo sa $\{\mathbf{y}_i\}_{i=1,2,...}$ niz uzastopnih tačaka dobijenih tokom postupka srednjeg pomeraja. Prema definiciji, za svako $k = 1, 2, ...$:

$$y_{k+1} = \frac{1}{n_k} \sum_{\mathbf{x}_i \epsilon S_h(\mathbf{y}_k)} \mathbf{x}_i, \tag{3.26}$$

gde je y_1 centar početnog prozora, a n_k je broj tačaka koje se nalaze u prozoru $S_h(\mathbf{y}_k)$ centriranom oko y_k. Konvergencija pomaka srednje vrednosti je opravdana kao posledica jednačine $M_h(\mathbf{x}) = \frac{h^2}{d+2} \frac{\widehat{\bigtriangledown} f(\mathbf{x})}{\widehat{f}(\mathbf{x})}$. Međutim, iako je tačno da vektor pomaka srednje vrednosti $M_h(\mathbf{x})$ ima smer gradijenta ocene gustine u tački $\mathbf{x}$, nije očigledno da je ocena gustine na lokacijama $\{\mathbf{y}_i\}_{i=1,2,...}$ monoton rastući niz. Kretanje u smeru gradijenta garantuje penjanje uz brdo samo za beskonačno male korake. Sledeća teorema pokazuje konvergenciju za diskretne podatke:

Teorema 1 *Neka je* $\widehat{f}_E = \left\{\widehat{f}_k(\boldsymbol{y}_k, K_E)\right\}_{k=1,2,...}$ *niz ocena gustine dobijenih koristeći Epanješnjikovljevo jezgro, izračunatih u tačkama* $\{\boldsymbol{y}_i\}_{i=1,2,...}$ *definisanim uzastopnim lokacijama algoritma pomaka srednje vrednosti sa uniformnim jezgrom. Ovaj niz konvergira.*

Dokaz se može naći u [17].

Bitno je napomenuti da algoritam pomeranjem sredine ne zahteva unapred definisan broj klastera, što ga čini posebno korisnim za podatke gde broj klastera nije

poznat unapred. Međutim, izbor radijusa prozora može imati veliki uticaj na rezultate klasterovanja. Takođe ne pravi nikakve pretpostavke o raspodeli podataka i može da obrađuje proizvoljne oblike i veličine klastera, pa se ovo grupisanje se može primeniti na različite tipove podataka, uključujući obradu slike i videa, praćenje objekata i bioinformatiku.

3.5 Mere kvaliteta modela

Procena da li je određeno grupisanje dobro ili ne je problematično i kontroverzno pitanje. U stvari, Boner ([18]) je bio prvi koji je tvrdio da ne postoji univerzalna definicija šta je dobro grupisanje. Ipak, u literaturi je razvijeno nekoliko kriterijuma za evaluaciju. Ovi kriterijumi se obično dele u dve kategorije: interne i eksterne.

3.5.1 Interni kriterijumi

Interni kriterijumi ocenjuju kvalitet klasterizacije na osnovu samih podataka, bez korišćenja spoljašnjih informacija ili oznaka. Ove mere se fokusiraju na ocenjivanje kompaktnosti, kohezije i razdvajanja klastera.

Neki uobičajeni interni kriterijumi uključuju:

- Danov indeks (eng. Dunn index): Meri kompaktnost klastera i razdvajanje između klastera. Izračunava se kao minimum udaljenosti između klastera podeljen sa maksimalnim dijametrom klastera. Dijametar klastera je maksimalna udaljenost između tačaka unutar klastera.

  Računa se kao:

  $$D = \min_{i,j} \frac{d(i,j)}{\max_k diam(K_k)}, \tag{3.27}$$

  gde je $d(i,j)$ razdaljina između klastera i i j, $diam(K_k)$ je dijametar klastera K_k, $\min\limits_{i,j}$ predstavlja minimum po svim klasterima, a $\max\limits_{k}$ predstavlja maksimum preko svih klastera.

  Optimalna vrednost je maksimalna.

- Dejvis-Buldin indeks: Meri prosečnu sličnost između klastera. Izračunava se kao prosečna sličnost između klastera, gde je sličnost izračunata kao razmera između unutrašnjeg rastojanja klastera (kohezije) i međusobnog rastojanja klastera (razdvajanja).

Računa se kao:

$$SB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left(\frac{S_i + S_j}{d(C_i, C_j)} \right). \tag{3.28}$$

Optimalna vrednost je minimalna.

- Koeficijent siluete: Opisan u glavi 3.2.1.
- Kalinski-Herabeš indeks: Opisan u glavi 3.2.1.

3.5.2 Eksterni kriterijumi

- Randov indeks (eng. Rand Index): Randov indeks meri sličnost između dve particije podataka, gde jedna particija predstavlja stvarnu strukturu podataka, a druga rezultat klasterizacije. RI se kreće između 0 i 1.

  Računa se kao:

  $$RI = \frac{a + b}{a + b + c + d}, \tag{3.29}$$

  gde je a broj parova instanci koje su u istom klasteru u stvarnoj i dobijenoj particiji podataka, b broj parova instanci koje su u različitim klasterima u stvarnoj i dobijenoj particiji podataka, c broj parova instanci koje su u istom klasteru u stvarnoj particiji, ali u različitim klasterima u dobijenoj particiji, a d broj parova instanci koje su u različitim klasterima u stvarnoj particiji, ali u istom klasteru u dobijenoj particiji.

  Optimalna vrednost je maksimalna.

- Preciznost i odziv (eng. Precision and Recall): Preciznost meri koliko tačno klaster sadrži istinski slične instance, dok odziv meri koliko istinski sličnih instanci je grupisano u isti klaster. F1 skor (eng. F1-score) se često koristi kao jedinstvena mera za ocenjivanje klasterizacije, kombinujući preciznost i odziv.

  Računaju se kao:

  $$\text{Precision} = \frac{TP}{TP + FP}, \tag{3.30}$$

  $$\text{Recall} = \frac{TP}{TP + FN}, \tag{3.31}$$

  $$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{3.32}$$

  gde su TP (True Positive) broj tačno pozitivnih instanci, odnosno broj instanci koje su ispravno grupisane u isti klaster kao i u stvarnoj particiji, FP

(False Positive) broj lažno pozitivnih instanci, odnosno broj instanci koje su grupisane u isti klaster, ali ne pripadaju istom klasteru u stvarnoj particiji, FN (False Negative) broj lažno negativnih instanci, odnosno broj instanci koje pripadaju istom klasteru u stvarnoj particiji, ali su razdvojene u dobijenoj particiji.

Optimalne vrednosti su maksimalne.

- Žakarov indeks (eng. Jaccard index): Žakarov indeks meri sličnost između stvarne i dobijene particije podataka kao intersekciju nad unijom instanci koje pripadaju istom klasteru. Žakarov indeks varira od 0 do 1.

  Računa se kao:
  $$J = \frac{|A \cap B|}{|A \cup B|}, \tag{3.33}$$
  gde su A i B dva skupa instanci za poređenje, gde A predstavlja skup instanci u stvarnoj particiji, a B skup instanci u dobijenoj particiji,$|A \cap B|$ broj instanci koje se nalaze u oba skupa (presek) i $|A \cup B|$ ukupan broj instanci u oba skupa (unija).

  Optimalna vrednost je maksimalna.

- Normirana obostrana informacija (eng. Normalized Mutual Information, NMI): NMI meri koliko informacija se deli između stvarne i dobijene particije podataka. NMI varira od 0 do 1.

  Računa se kao:
  $$NMI(X, Y) = \frac{MI(X, Y)}{\sqrt{H(X) \cdot H(Y)}}, \tag{3.34}$$
  gde su X i Y dva skupa instanci za poređenje, gde X predstavlja skup instanci u stvarnoj particiji, a Y skup instanci u dobijenoj particiji,$MI(X, Y)$ obostrana informacija između dva skupa X i Y, koja meri koliko informacija se deli između dva skupa, a $H(X)$ i $H(Y)$ predstavljaju neodređenost skupova X i Y. Neodređenost se računa kao $H(X) = -\sum_i P(x_i) \log_2 P(x_i)$, gde je $P(x_i)$ verovatnoća pojave elementa x_i u skupu X.

  Optimalna vrednost je maksimalna.

3.6 Pretprocesiranje

Pretprocesiranje i klasterovanje su dve različite faze u analizi podataka i mašinskom učenju. Pretprocesiranje se bavi pripremom podataka, dok klasterovanje se bavi identifikacijom sličnih grupa u tim podacima. Dakle, iako se nalaze na različitim tačkama analitičkog procesa, ove dve faze su neraskidivo povezane, tačnije dobro pretprocesiranje je osnova za uspešno klasterovanje. Iz tog razloga ćemo se u ovoj glavi ukratko osvrnuti na tehnike pretprocesiranja podataka.

Pretprocesiranje slika obuhvata niz tehnika i operacija koje se primenjuju na slikama pre nego što se primeni sam postupak klasterovanja. Cilj pretprocesiranja je poboljšanje kvaliteta slika, uklanjanje šuma, isticanje važnih karakteristika i priprema slike za efikasno i tačno klasterovanje.

Odabir prostora boja

Prostor boja je matematički model koji opisuje boje na način koji omogućava njihovo reprezentovanje i manipulisanje u digitalnom obliku. Oni se koriste u digitalnoj obradi slika, grafici, fotografiji, video produkciji i drugim srodnim oblastima.

Svaki prostor boja se sastoji od tri glavna elementa: primarnih boja, bele tačke i funkcije prenosa boja. Primarne boje su tri boje koje se mogu koristiti da se stvori bilo koja druga boja u tom prostoru. Bela tačka se koristi za definisanje osvetljenja u kojem se boje posmatraju. Funkcija prenosa boja određuje kako se boje predstavljaju u digitalnom obliku, kao i kako se prenose između različitih uređaja, npr. između kamere i monitora.

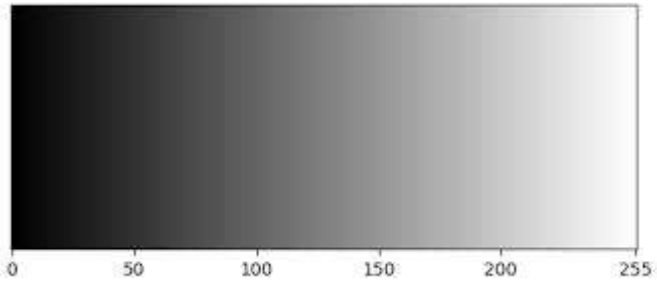
Postoje mnogi prostori boja, ali najčešće korišćeni su **RGB** (crveno-zeleno-plavo), **CMYK** (cijan-magenta-žuta-crna), **HSL** (nijansa-zasićenje-osvetljenje), **HSV** (nijansa-zasićenje-vrednost), **LAB** (svetlina-a-b), **YUV/YCbCr** (luma-croma) i **CIE Lab*** (svetlina-crveno-zeleno-plavo). Svaki od ovih prostora boja ima svoje prednosti i nedostatke, i koristi se u određenim aplikacijama u zavisnosti od potreba. Odabir odgovarajućeg prostora boja veoma je važan za proces segmentacije slike.

Crno-beli (eng. grayscale) ili monohromatski prostor boja, je poseban slučaj prostora boja u kojem se boje predstavljaju samo nijansama sive. Crno-beli prostor boja, za razliku od RGB ili HSV prostora boja, koristi samo jednu komponentu koja

predstavlja intenzitet nijanse sive. Vrednosti u crno-belom prostoru boja se takođe obično kreću u opsegu od 0 do 255, gde 0 predstavlja crnu boju (najniži intenzitet) a 255 predstavlja belu boju (najviši intenzitet), što je prikazano na Slici 3.15.

Crno-bele slike su korisne u mnogim situacijama, posebno kada je fokus na naglašavanju oblika, tekstura ili kontrasta, a ne na boji. Ove slike su jednostavnije za obradu i zahtevaju manje prostora za skladištenje u odnosu na RGB slike. Takođe, crno-bele slike su pogodne za primene poput medicinske dijagnostike, analize slika i mnogih drugih oblasti gde boja nije presudna.

Crno-bele slike mogu dobiti konverzijom RGB slika, gde se intenzitet svakog piksela izračunava kao srednja vrednost komponenti crvene, zelene i plave boje.



Slika 3.15: Vizuelna reprezentacija crno-belog prostora boja sa vrednostima piksela

Uklanjanje šuma

Uklanjanje šuma sa slike važan je korak u obradi slika jer šum može značajno uticati na kvalitet slike i otežati dalju obradu. Šum na slici može biti uzrokovan različitim faktorima, kao što su loš kvalitet kamere, loše osvjetljenje, preniska ili previsoka ISO vriednost, loša kompresija slike itd.

Kako šum na slici može otežati prepoznavanje objekata na slici i otežati primenu algoritama za obradu slika, uklanjanje šuma sa slike je važan korak u obradi slika kako bi se poboljšale performanse algoritama za obradu slika.

Postoje različiti načini za uklanjanje šuma sa slike. Neki od njih su:

- Gausov filter: Ovaj filter koristi Gausovu funkciju za uklanjanje šuma slike.
- Medijalni filter: Ovaj filter koristi vrednost medijane piksela u jezgru za uklanjanje šuma. Velika prednost ovog filtera je što zadržava ivice slike bolje od Gausovog filtera.

- Bilateralni filter: Ovaj filter koristi funkciju težine koja uzima u obzir udaljenost piksela i razliku u vrednosti piksela. On bolje održava ivice slike od drugih filtera i pogodan je za uklanjanje šuma sa slike bez gubitka oštrine ivica.

Normalizacija slike

Normalizacija boja je važan korak u pretprocesiranju slika. Postoje različiti načini na koje se mogu normalizirati boje slike, ali u osnovi se radi o skaliranju vrednosti boja u raspon od 0 do 1 ili -1 do 1, zavisno od korišćene skale boja. Jedan od načina normalizacije boja slika je primena standardne normalizacije (eng. Z-score normalization) na vrednosti piksela slike. Ovaj postupak podrazumeva skaliranje vrednosti piksela tako da imaju srednju vrednost 0 i standardnu devijaciju 1. Ova normalizacija pomaže u uklanjanju efekta različitih osvetljenja i kontrasta na slikama i sprečavanju zasićenja i eksplozije gradijenata tokom treniranja modela.

Drugi način normalizacije boja slika je min-max normalizacija, koja skalira vrednosti piksela slike u raspon od 0 do 1. Ova normalizacija pomaže u uklanjanju razlika u rasponu vrednosti piksela između različitih slika.

Glava 4

Eksperiment

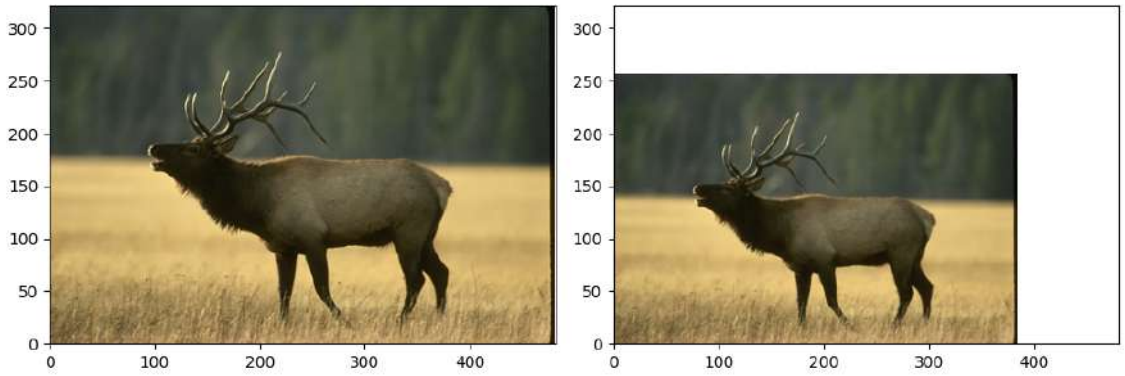
Skup podataka koji je korišćen je *Berkeley Segmentation Dataset* ([19]) koji se sastoji od 500 slika različitih veličina i rezolucija. Skup podataka je kreiran na Univerzitetu Kalifornije kako bi se omogućilo testiranje i upoređivanje različitih algoritama za segmentaciju slika. BSD sadrži tri podskupa podataka: BSDS300, BSDS500 i BSDS1000, a svaki od njih ima različite veličine skupa podataka i metrike za poređenje performansi algoritama segmentacije.
U ovom istraživanju biće korišćen BSDS500 koji je proširenje BSDS300 i sastoji se od 300 slika za trening i validaciju u 200 slika za testiranje. Deo skupa podataka prikazan je na Slici 4.1.

Slika 4.1: Primer skupa podataka BSDS500

4.1 Pretprocesiranje slika

Dimenzije slike

Kako bi se ubrzao proces obrade slike i smanjili zahtevi za memorijom, potrebno je smanjiti veličinu slike. U našem slučaju, slike se kreću od 128 kB, pa ćemo smanjiti i visinu i širinu za 20%. Na Slici 4.2 prikazana je originalna i skalirana slika.



Slika 4.2: Promena veličine slike

Odabir prostora boja

Odabran je crno-beli prostor boja. Originalna slika je konvertovana u ovaj prostor boja, što je prikazano na Slici 4.3.

Slika 4.3: Slika u HSV prostoru boja (desno)

Uklanjanje šuma

U našem slučaju, za uklanjanje šuma je korišćen bilateralni filter. Slika nakon primene ovog filtera je prikazana na Slici 4.4.

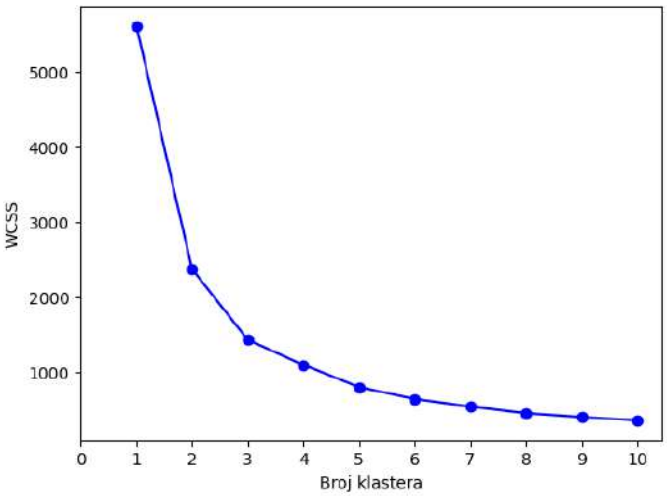
Slika 4.4: Slika nakon uklanjanja šuma (desno)

Normalizacija slike

Kada se radi o klasterizaciji slika u crno-belom prostoru boja, kao u našem slučaju, jedna od uobičajenih metoda normalizacije boja jeste skaliranje vrednosti na raspon od 0 do 1. Ovo se može učiniti primenom min-max normalizacije na svakoj od komponenti, tako da vrednosti u svakoj komponenti budu skalirane na raspon od 0 do 1.

4.2 K-sredina

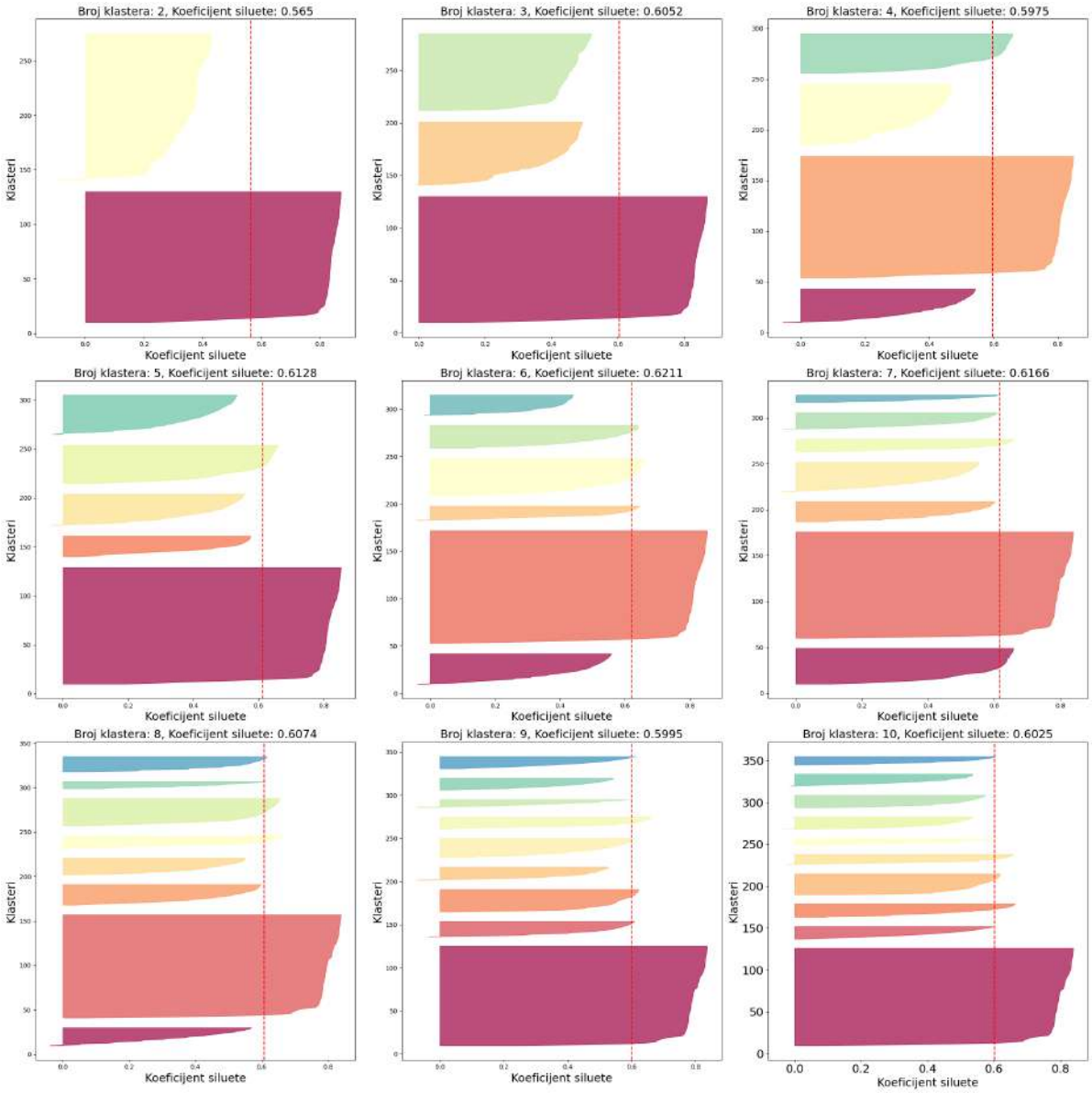
Nakon pretprocesiranja slike možemo primeniti algoritam. Međutim, pre tog koraka, odredićemo optimalni broj klastera, metodom lakta. Ramatraćemo rezultate za $k = 2$ do $k = 10$. Rezultat koji dobijamo ne daje nam jasnu sliku o optimalnom broju klastera (Slika 4.5).



Slika 4.5: Metoda lakta za određivanje broja klastera

Na Slici 4.6 prikazani su grafici za različit broj klastera $k = 2, ..., 10$ i odgovarajući koeficijent siluete. Vertikalnom crvenom linijom označen je prosečni koeficijent siluete. Svaki klaster na grafiku je predstavljen različitom bojom kako bi se ilustrovalo kako su tačke unutar klastera grupisane u odnosu na ostale klastere. Širina obojene oblasti za svaki klaster predstavlja gustinu tačaka unutar tog klastera. Šira oblast sugeriše na veću gustinu tačaka, dok uža oblast sugeriše na manju gustinu.

Koeficijent siluete je najveći za $k = 6$, tako da ćemo nadalje vršiti klasterizaciju na 6 klastera.



Slika 4.6: Metoda siluete za određivanje broja klastera

K-sredina klasterovanje je izvršeno podešavanjem određenih parametra:

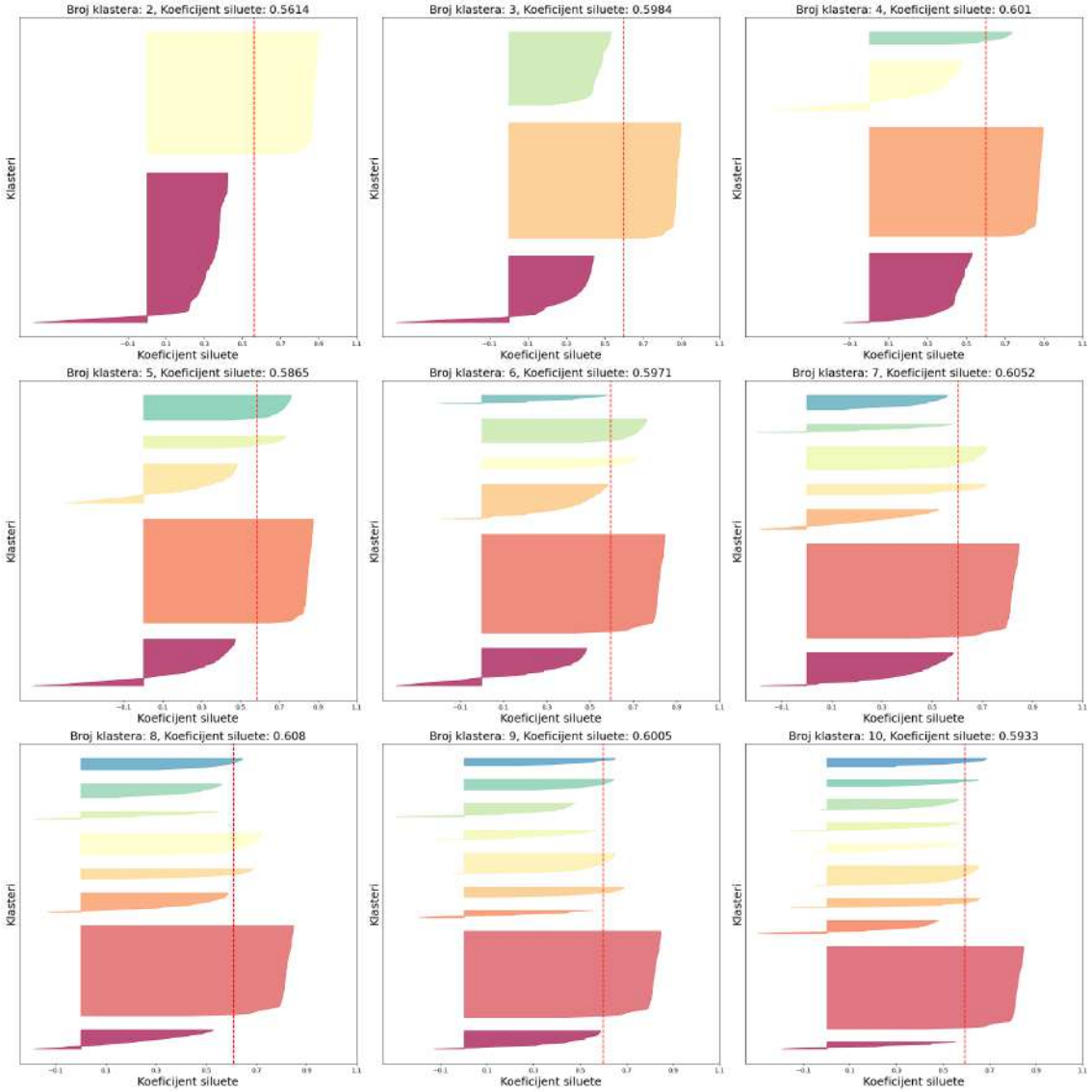
1. broj klastera, k=6
2. kriterijum zaustavljanja algoritma klasterovanja je kombinacija maksimalnog broja iteracija i željene tačnosti: algoritam će se zaustaviti kada dostigne maksimalno 10 iteracija ili kada promena u centrima klastera padne ispod 1.0
3. broj puta koliko će se pokrenuti algoritam sa nasumičnim početnim centrima klastera, $n = 10$. Nakon više pokretanja, najbolje rezultate će biti izabrano kao konačni rezultat.
4. inicijalizacija centara klastera: u našem slučaju, centri klastera će biti inicijalizovani nasumično.

Nakon primene algoritma k-sredina, na Slici 4.7 možemo videti rezultate klasterovanja.

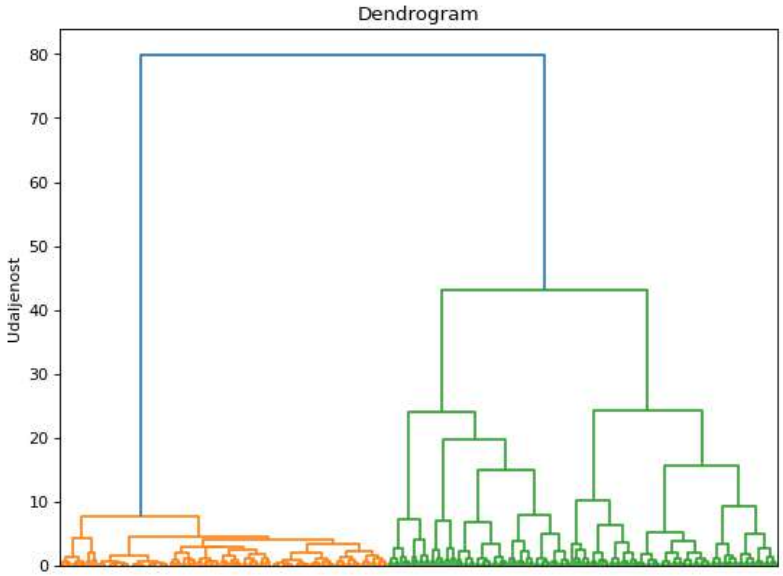
Slika 4.7: Originalna slika i slika nakon primene k-sredina klasterovanja

4.3 Hijerarhijsko klasterovanje

Algoritam hijerarhijskog klasterovanja koji smo izabrali je klasterovanje spajanjem sa Vardovim povezivanjem i $k = 8$. Parametar k je izabran pomoću metode siluete (Slika 4.8). Dendrogram piksela slike prikazan je na Slici 4.9, a rezultat klasterovanja na Slici 4.10.



Slika 4.8: Metoda siluete za određivanje broja klastera



Slika 4.9: Dendrogram piksela slike

Slika 4.10: Originalna slika i slika nakon primene klasterovanja spajanjem

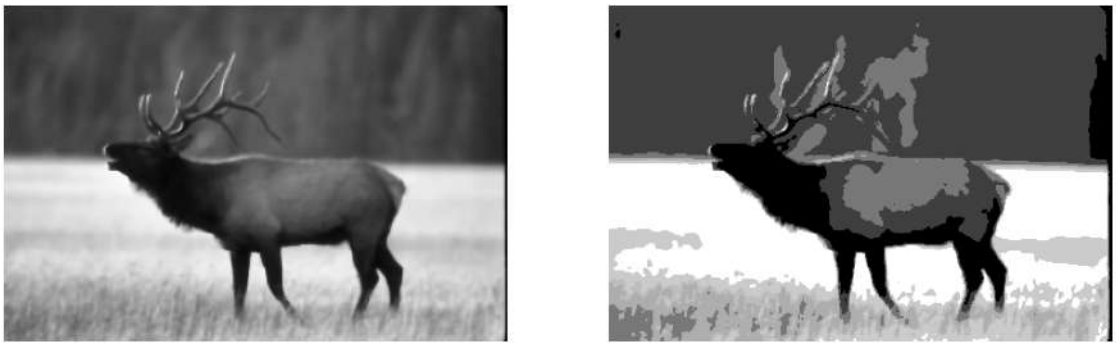
4.4 Klasterovanje pomeranjem sredine

Prilikom klasterovanja pomeranjem sredine (u daljem tekstu KPS), potrebno je odrediti koje jezgro ćemo koristiti, sto je u našem slučaju ravno jezgro i koja je veličina radijusa prozora. Što se tiče radijusa prozora, njega ćemo proceniti pomoću funkcije koja radi na osnovu kvantilne metode, koja se temelji na interkvartilnom opsegu podataka. Za N-dimenzionalne podatke, radijus se ocenjuje kao:

$$r = q \cdot m(\|X_i - X_j\|_2),$$

gde je r ocena radijusa prozora, q vrednost kvantila[1], koja se koristi kao mera skale, X_i i X_j nasumično izabrane tačke iz skupa podataka, $\|X_i - X_j\|^2$ kvadrat Euklidske udaljenosti između tačaka X_i i X_j, $m(\|X_i - X_j\|^2)$ medijana[2] kvadrata Euklidskih udaljenosti između svih parova tačaka u skupu podataka.

Vrednost kvantila između 0.2 i 0.3 je empirijski odabrana da funkcioniše dobro za različite vrste podataka. Ovako procenjen radijus je 0.03. Nakon primene algoritma za radijus prozora 0.03 i ravno jezgro, dobijamo i procenjen broj klastera 7. Rezultat klasterovanja je prikazan na Slici 4.11.



Slika 4.11: Originalna slika i slika nakon primene klasterovanja pomeranjem sredine

4.5 Mere kvaliteta modela

U ovom eksperimentu, kao mere kvaliteta modela korišćene su:

- Kvadratni koren iz srednje kvadratne greške (RMSE): Smatra se standardom merenja performansi izlazne slike. U tom slučaju, pikseli slike se grupišu u klastere na osnovu sličnosti njihovih boja. Nakon što se izvrši grupisanje piksela, može se izračunati RMSE između stvarnih boja piksela i boja klastera kojima su dodijeljeni. Konkretnije, za svaki piksel se određuje kojem klasteru pripada, a zatim se za svaki klaster izračunava prosečna boja. RMSE se tada računa kao kvadratni koren prosečne kvadratne greške između stvarne boje piksela i prosečne boje klastera kojem pripada. U ovom slučaju, niže vrijednosti RMSE

[1]Kvantil je vrednost koja deli uzorak na dva dela koji su u odnosu $q : (1 - q)$.

[2]Medijana je mera centralne tendencije skupa podataka. To je vrednost koja deli skup podataka na dva jednaka dela, tako da polovina podataka ima vrednosti manje od medijane, a druga polovina ima vrednosti veće od medijane. Ako skup podataka ima neparan broj elemenata, medijana je srednja vrednost kada se podaci sortiraju po veličini. Ako skup podataka ima paran broj elemenata, medijana je aritmetička sredina dve srednje vrednosti.

ukazuju na bolji kvalitet klasterizacije, odnosno na to da su pikseli u svakom klasteru sličniji po boji, a razlike između klastera su veće.

- Maksimalni odnos signala i šuma (PSNR): PSNR je metrika koja se koristi za merenje kvaliteta rekonstruisanog signala u odnosu na originalni signal. Definiše se kao odnos između maksimalne moguće snage signala i kvadrata srednjekvadratne greške (MSE) između originalnog i rekonstruisanog signala. PSNR se izražava u decibelima (dB) i veće PSNR vrednosti su bolje. Uobičajeno se smatra da su vrednosti PSNR-a veće od $30dB$ dobre za segmentaciju crno-belih slika. Međutim, optimalne vrednosti mogu varirati u zavisnosti od specifične primene i zahteva.

  Matematički, PSNR se može definisati kao:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right).$$

- Kalinski-Herabeš indeks (CH): Opisan u glavi 3.2.1.

Rezultati klasterovanja prikazani su u Tabeli 4.1 i na Slici 4.12.

algoritam	k	RMSE	PSNR	CH
k-sredina	6	0.051668	25.73553	2419.9569
hijerarhijsko k.	8	0.093477	20.58588	1254.2249
KPS	7	0.032416	29.78475	3709.0759

algoritam	k	RMSE	PSNR	CH
k-sredina	3	0.05127	25.80248	3498.9388
hijerarhijsko k.	2	0.08041	21.89433	4059.8312
KPS	6	0.03389	29.39809	4293.9736

algoritam	k	RMSE	PSNR	CH
k-sredina	2	0.13934	17.11865	723.0138
hijerarhijsko k.	2	0.13820	17.18967	730.5012
KPS	10	0.03451	29.24137	2664.2891

algoritam	k	RMSE	PSNR	CH
k-sredina	8	0.04577	26.78891	2608.3471
hijerarhijsko k.	8	0.03122	30.11154	4524.9018
KPS	11	0.03792	28.42346	3115.0828

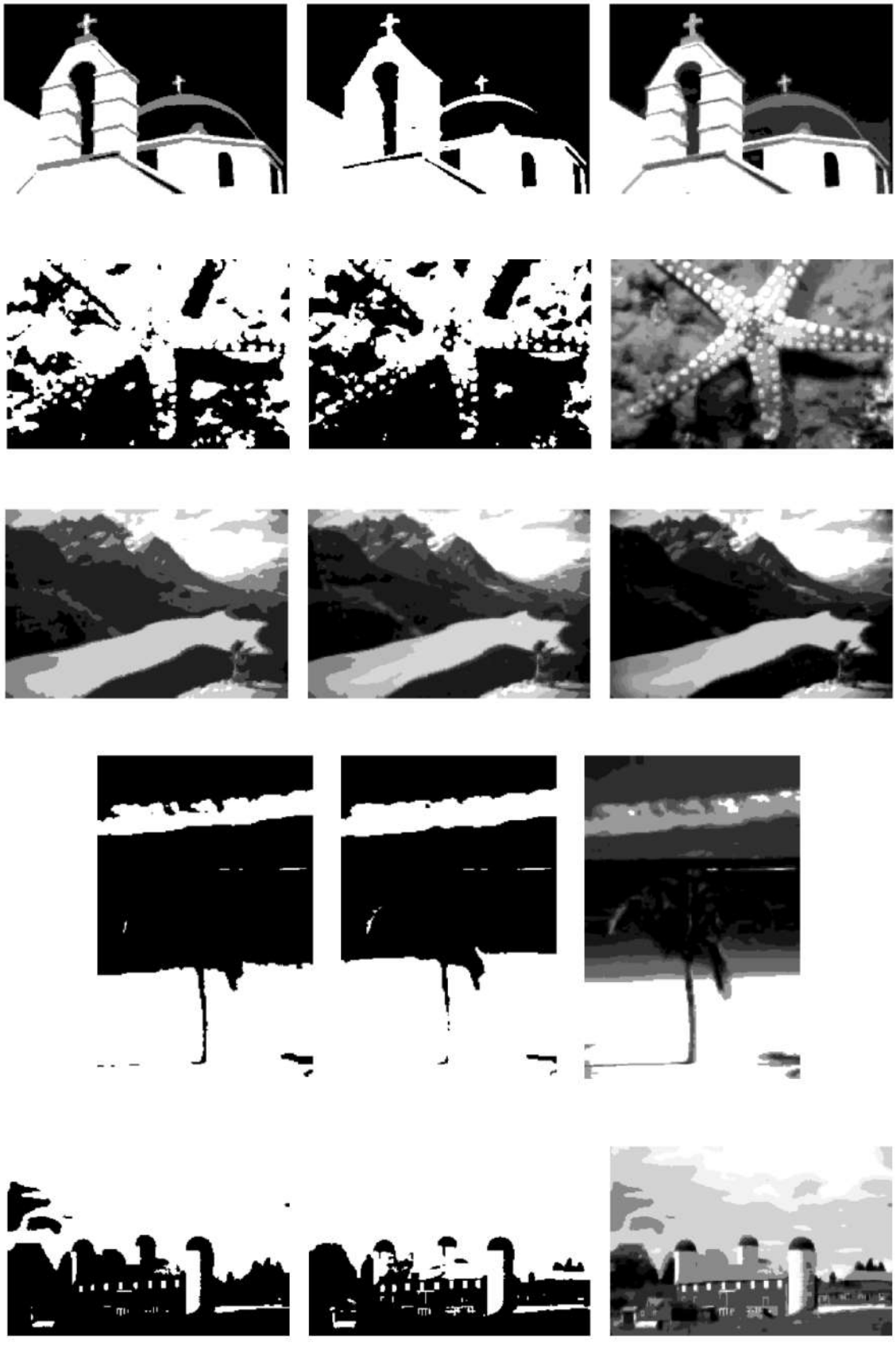
algoritam	k	RMSE	PSNR	CH
k-sredina	2	0.11642	18.67943	2214.2731
hijerarhijsko k.	2	0.13576	17.34447	1451.3552
KPS	7	0.03736	28.55234	3860.4762

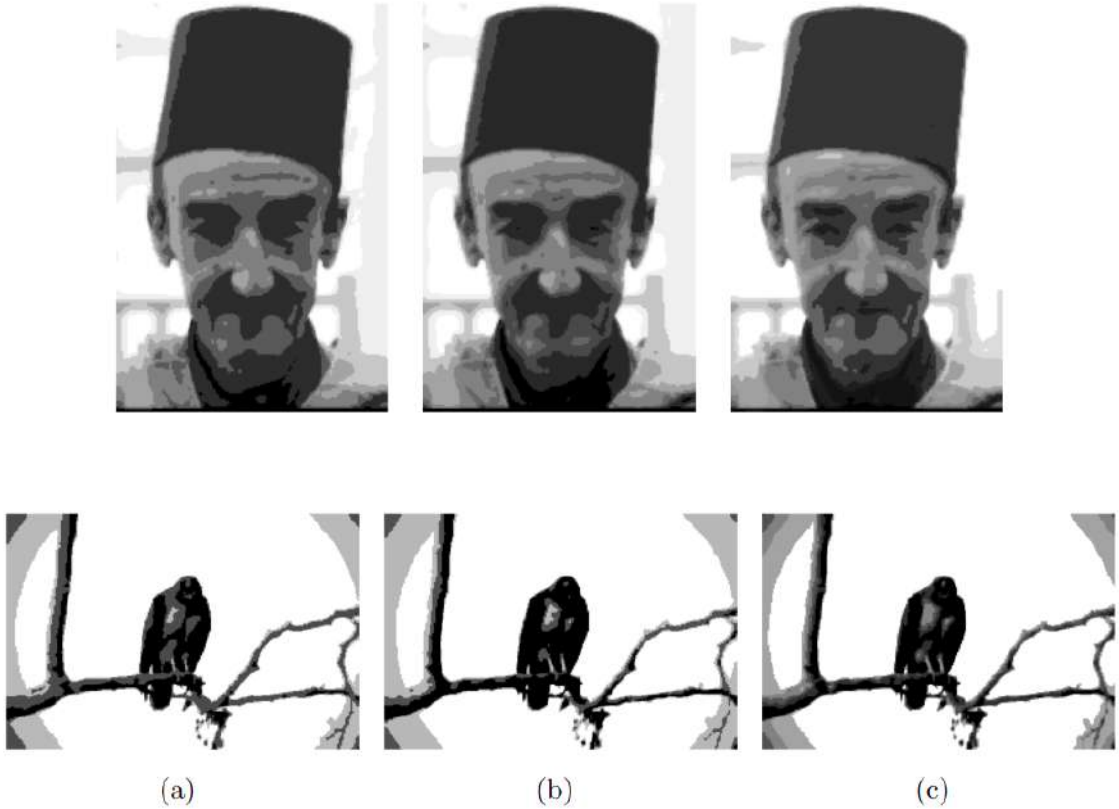
algoritam	k	RMSE	PSNR	CH
k-sredina	2	0.10498	19.57752	1462.1038
hijerarhijsko k.	2	0.11046	19.13622	1456.3103
KPS	8	0.04294	27.34277	1865.4641

algoritam	k	RMSE	PSNR	CH
k-sredina	10	0.03782	28.44474	2349.6489
hijerarhijsko k.	9	0.03109	30.14875	5339.3528
KPS	11	0.03027	30.38043	4080.7093

algoritam	k	RMSE	PSNR	CH
k-sredina	4	0.05565	25.09038	2128.7701
hijerarhijsko k.	4	0.14041	17.05209	1972.1456
KPS	6	0.04967	26.07753	2914.7038

Tabela 4.1: Rezulatati klasterovanja





Slika 4.12: Rezultati klasterovanja primenom: a) k-sredina, b) hijerarhijskog klasterovanja, c) klasterovanja pomeranjem sredine

Glava 5

Zaključak

Na osnovu rezultata analize, zaključujemo da se klasterovanje pomeranjem sredine izdvaja kao najpouzdaniji i najefikasniji algoritam za segmentaciju slika u ovom konkretnom istraživanju. Ova zaključna tvrdnja potkrepljuje se rezultatima koje su dobijeni kroz poređenje RMSE, PSNR i Kalinski-Herabeš indeksa, gde je klasterovanje pomeranjem sredine ostvarilo najbolje rezultate u smislu smanjenja greške i povećanja kvaliteta slike.

Iako su i algoritmi k-sredina i hijerarhijsko klasterovanje pružili zadovoljavajuće rezultate, klasterovanje pomeranjem sredine se istaklo svojom sposobnošću da prilagodi strukturu podataka bez potrebe za unapred definisanim brojem klastera. Ova fleksibilnost algoritma pokazuje se ključnom osobinom u situacijama gde je struktura podataka kompleksna i varijabilna.

Sa druge strane, algoritam k-sredina je privukao pažnju svojom visokom brzinom izvođenja i minimalnog zahteva za memorijom, što ga čini efikasnim izborom za segmentaciju slika i druge analize velikih skupova podataka. Međutim, zbog osetljivosti na početni izbor centara klastera, rezultira manje optimalnim klasterovanjem.

Hijerarhijsko klasterovanje omogućava formiranje hijerarhije klastera, ali zahteva više vremena za izvođenje i izazovno za velike skupove podataka, što je kod nas slučaj, pa nije najbolji izbor.

Zaključno, ovo istraživanje daje dragocen doprinos razumevanju različitih algoritama za segmentaciju slika i njihovih performansi u odnosu na navedene metrike. Upotreba klasterovanja pomeranjem sredine kao preferiranog algoritma pruža obećavajuće rezultate za segmentaciju slika sa ciljem povećanja kvaliteta i tačnosti. U budućem radu, dalje istraživanje može se fokusirati na optimizaciju parametra ovog algoritma i ispitivanje njegove primene u različitim scenarijima segmentacije

slika. Takođe, istraživanje može proširiti analizu upoređivanjem klasterovanja pomeranjem sredine sa drugim naprednim tehnikama segmentacije kako bi se bolje razumela njegova potencijalna primena u raznovrsnim praktičnim situacijama.

Literatura

[1] R. E. Woods i R. C. Gonzalez. *Digital image processing*. Pearson Education Ltd., 2008.

[2] S. Wehmeier, C. McIntosh, J. Turnbull, M. Ashby i A.S. Hornby. *Oxford Advanced Learner's Dictionary: Of Current English*. Oxford University Press, 2005.

[3] D. Todorović. „Gestalt principles". U: *Scholarpedia* (2008). URL: `http://www.scholarpedia.org/article/Gestalt_principles?__hstc=77520074.36a0ddae8e24bce7`.

[4] M. Nikolić i A. Zečević. „Mašinsko učenje". `https://ml.matf.bg.ac.rs/readings/ml.pdf`. 2019.

[5] D. Ruppert. *The elements of statistical learning: data mining, inference, and prediction*. Taylor & Francis, 2004.

[6] C. M. Bishop i N. M. Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

[7] J. Han, M. Kamber i J. Pei. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011.

[8] J. Leskovec, A. Rajaraman i J. D. Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

[9] J. H. Ward Jr. „Hierarchical grouping to optimize an objective function". U: *Journal of the American statistical association*. Taylor & Francis, 1963.

[10] P. Macnaughton-Smith, W. Williams, M. Dale i L. G. Mockett. „Dissimilarity analysis: a new technique of hierarchical sub-division". U: *Nature*. Nature Publishing Group UK London, 1964.

[11] L. Kaufman i P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. Wiley, 1990.

[12] D. Comaniciu i P. Meer. „Mean shift: A robust approach toward feature space analysis". U: *IEEE Transactions on pattern analysis and machine intelligence*. IEEE, 2002.

[13] M.P. Wand i M.C. Jones. *Kernel Smoothing*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1994.

[14] A. Kostić. „Ocenjivanje gustine raspodele". master rad. Matematički fakultet, Beograd, 2017.

[15] Y. Cheng. „Mean shift, mode seeking, and clustering". U: *IEEE transactions on pattern analysis and machine intelligence*. IEEE, 1995.

[16] D. Subotić. „Ocenjivanje gustine raspodela u pristustvu greške merenja". master rad. Matematički fakultet, Beograd, 2017.

[17] D. Comaniciu i P. Meer. „Mean shift analysis and applications". U: *Proceedings of the seventh IEEE international conference on computer vision*. IEEE, 1999.

[18] R. E. Bonner. „On some clustering techniques". U: *IBM journal of research and development*. IBM, 1964.

[19] D. Martin, C. Fowlkes, D. Tal i J. Malik. „A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics". U: *Proc. 8th Int'l Conf. Computer Vision*. IEEE, 2001.

Biografija autora

Jelena Stojiljković rođena je 17.11.1996. u Čačku gde je završila Osnovnu školu „Sveti Sava", a potom Gimnaziju u Čačku, prirodno-matematički smer. Matematički fakultet, modul statistika, aktuarska i finansijska matematika, je upisala 2015. godine i diplomirala 2020. godine. Paralelno sa master studijama, započinje i profesionalnu karijeru u oblasti veb programiranja.