# Klasterizacija slika primjenom algoritma roja čestica

Mirza Sinanović
*Elektrotehnički fakultet*
*Univerzitet u Sarajevu*
Sarajevo, Bosna i Hercegovina
msinanovic1@etf.unsa.ba

Senid Hodžić
*Elektrotehnički fakultet*
*Univerzitet u Sarajevu*
Sarajevo, Bosna i Hercegovina
shodzic4@etf.unsa.ba

***Sažetak*—Klasterizacija slika je važan, ali izazovan zadatak u mašinskom učenju i računarskoj viziji. U ovom radu je razvijena klasterizacija slika primjenom algoritma optimizacije rojem čestica (engl. *PSO - Particle Swarm Optimization*). Algoritam pronalazi centre svakog od klastera, gdje je svaki klaster sačinjen od grupe sličnih piksela. Implementirane su dvije verzije algoritma koje se razlikuju u načinu računanja parametara algoritma. Obzirom da PSO algoritam zahtijeva pogodno definisanu fitnes funkciju, u radu su detaljno opisane i upoređene tri fitnes funkcije pogodne za probleme klasteringa slike. Kombinacije algoritama i fitnes funkcija su upoređene na slici magnetne rezonance glave.**

***Abstract*—Image clustering is an important but also challenging task in machine learning and computer vision. In this paper, image clustering was developed using the particle swarm optimization algorithm. The algorithm finds the center of each of the clusters, where each cluster is made up of groups of similar pixels. Two versions of the algorithm have been implemented that differ in the way the parametric algorithm is calculated. Considering the PSO algorithm requires a convenient definition of the fitness function, the paper describes in detail and compares three fitness functions suitable for image clustering problems. Combinations of algorithms and fitness functions were compared in the head magnetic resonance image.**

## I. Uvod

U našem svakodnevnom životu segmentacija nam pomaže prilikom donošenja različitih odluka. Segmentacija slike odnosi se na proces razdvajanja digitalne slike na više nepreklapajućih regiona. Pikseli grupisani u jednoj regiji su slični prema nekim kriterijima sličnosti kao što su boja, intenzitet ili tekstura. Algoritmi korišteni za segmentaciju imaju dvije osobine: dijele piksele u različite klastere na osnovu razlike u intenzitetu i grupišu slične piksele na osnovu kriterija sličnosti [1]. Segmentacije slika potrebna je pri rješavanju različitih problema, uključujući kvantizaciju boja, otkrivanje promjena zemljišnog prekrivača (engl. *land cover monitoring*), te u dubinskoj analizi podataka (engl. *data mining*). U području satelitskog istraživanja, cilj algoritma segmentacije slika je da automatski kategoriše sve piksele na slici u klase specifičnih zemljišnih pokrivača, pri čemu se ne narušava topologija [2].

Obzirom da i metode klasteringa na isti način dijele podatke, jasno je da se algoritmi za klastering mogu koristiti za segmentaciju slike tako što se izvrši klastering piksela [3]. Iako često korištene, standardne klastering metode poput K-Means-a imaju nekoliko nedostataka. Najveći problem predstavljaju ovisnost o izboru početnog rješenja te preuranjena konvergencija. U cilju prevazilaženja ovih nedostataka, predloženo je nekoliko optimizacijskih metoda, pri čemu se korištenje kolektivne inteligencije (engl. *Swarm Intelligence*) pokazalo kao dobar izbor. U ovom radu je odabran algoritam roja čestica (engl. *PSO - Particle Swarm Optimization*) kao najpopularnija metoda kolektivne inteligencije [4].
Mnogo radova je napisano na temu segmentacije slike korištenjem PSO, pri čemu se ti radovi razlikuju u načinu primjene PSO algoritma na problem segmentacije. U radovima [2], [5], [6] svaka od čestica predstavlja niz centara klastera koji se kroz iteracije poboljšavaju u odnosu na fitnes funkciju. S druge strane, u radu [7] čestica predstavlja matricu koja sadrži funkcije pripadnosti $u_{ij}$ , odnosno funkciju pripadnosti $i$-tog piksela $j$-tom klasteru. Implementacija PSO se, osim po predstavljanju čestica i fitnes funkcija, mogu razlikovati i po parametrima algoritma. Rad [2] predlaže korištenje fiksnih parametara $w$,$c_1$ i $c_2$, dok se u radu [5] predlaže formula prema kojoj se oni računaju u zavisnosti od broja trenutne iteracije.
U ovom radu je korišten pristup predstavljanja čestice kao niza centara klastera, u kombinaciji sa različitim fitnes funkcijama. Algoritam je implementiran u dvije varijante, pri čemu je razlika između varijanti u tome da li su parametri algoritma mijenjaju u toku izvršavanja algoritma ili ne. Korištenjem dvije mjere kvaliteta klasteringa izvršeno je poređenje navedena dva algoritma i tri fitnes funkcije.

## II. Korišteni algoritam

Kolektivna inteligencija (SI) je ponašanje prirodnih i umjetnih decentraliziranih, samoorganiziranih sistema. [8]. Algoritam optimizacije rojem čestica, jedna od metoda kolektivne inteligencije modelira ponašanje koje se susreću među živim bićima koja žive u rojevima, a predložen je od strane Kennedy-a, Eberharta i Shi-a 1995. godine.

Algoritam rojem čestica ne posjeduje mehanizam selekcije, za razliku od evolucionih algoritama. S tim u vezi, trenutno

loša potencijalna rješenja imaju priliku da kroz nadolazeće iteracije postanu korisna i upotrebljiva rješenja, pa se zbog toga ne uklanjaju iz populacije. Iako u standardnoj implementaciji svaka čestica u svakoj iteraciji bira određeni broj čestica kojima proslijeđuje informaciju, u implementaciji opisanoj u ovom radu svaka čestica prenosi svakoj drugoj čestici bitne informacije, odnosno svaka čestica predstavlja informanta za sve druge čestice. Informacija koja se dijeli među česticama odnosi se na poziciju najbolje čestice u roju pronađenu do trenutne iteracije. Pored informacije o najboljoj čestici u čitavom roju, svaka čestica pamti i najbolju poziciju do tada pronađenu od nje same [9].

Navedeni mehanizam se modificira sa svakom iteracijom, na osnovu raspoloživih informacija o česticama:

- $v_i$ : trenutna vrijednost vlastite brzine
- $y_i$ : memorirana najbolja vlastita pozicija
- $\hat{y}$ : pozicija najbolje čestice

Vlastita najbolja pozicija čestice $i$ do trenutne iteracije $t$ je ona u kojoj je čestica ostvarila najbolju vrijednost fitnesa. Ako $f$ označava funkciju fitnesa, tada se najbolja pozicija određuje kao: [2]

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad (1)$$

Kao što je ranije navedeno, nabolja najbolja čestica se određuje iz čitavog roja. Ovaj način istraživanja okoline se često naziva i gbest model. Ako je pozicija najbolje čestice označena sa $\hat{y}$, onda vrijedi:

$$\begin{aligned} \hat{y}(t) &\in \{y_0, y_1, ..., y_s\} \\ &= min\{f(y_0(t)), f(y_1(t)), ..., f(y_s(t))\} \end{aligned} \quad (2)$$

gdje je s ukupan broj čestica u roju. Za svaku iteraciju *gbest* PSO algoritma, $v_i$ i $x_i$ se određuju kao:

$$\begin{aligned} v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1(t) \cdot (y_i(t) - x_i) \\ + c_2 \cdot r_2(t) \cdot (\hat{y}(t) - x_i(t)) \end{aligned} \quad (3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

gdje je $w$ koeficijent inercione težine, $c_1$ i $c_2$ su akceleracione konstante a $r_1(t)$ i $r_2(t)$ slučajni brojevi iz intervala $[0, 1]$. Koeficijent inercione težine je pozitivan realan broj manji od 1, koji se u jednoj verziji algoritma opisanog u radu određuje kao :

$$w = w_{max} - iter\frac{w_{max} - w_{min}}{iter_{max}} \quad (5)$$

gdje se inerciona težina mijenja sa pozicijom čestice, tako da će algoritam teško zapasti u lokalni ekstrem. Vrijednost $iter$ predstavlja trenutnu iteraciju, a $iter_{max}$ predstavlja maksimalan broj iteracija. Parametri $c_1$ i $c_2$ u početku imaju konstantnu vrijednos, koja se mijenja nakon određenog broja iteracija tako što postepeno povećava parametar $c_1$, a parametar $c_2$ postepeno smanjuje $c_2$. Na ovaj način se u kasnijim iteracijama postepeno pridaje veća važnost poziciji najbolje čestice čitavog roja u odnosu na najbolju poziciju trenutne čestice. Oslanjajući se na rad [5], parametri $c_1$ i $c_2$ se računaju kao:

$$\begin{aligned} c_{1novo} &= e^{-\frac{iter}{iter_{max}}} \cdot c_1 \\ c_{2novo} &= c_2 + 1 - e^{-\frac{iter}{iter_{max}}} \end{aligned} \quad (6)$$

gdje $iter$ predstavlja vrijednost trenutne iteracije, a $iter_{max}$ predstavlja maksimalan broj iteracija. Ukoliko je odnos $\frac{iter}{iter_{max}} > 0.4$, tada se konstante $c_1$ i $c_2$ kao što je prethodno objašnjeno. U suprotnom one ostaju iste. Inicijalne vrijednosti konstanti se odabiru proizvoljno. Način računanja koeficienata opisan jednačinama i se koristi u jednoj verziji algoritma, dok su u drugoj verziji svi koeficienti konstantni tokom čitavog izvršavanja algoritma.

PSO algoritam ciklično izvršava navedene jednačine ažuriranja sve dok se ne ispuni uslov zaustavljanja, što je obično broj iteracija ili sve dok ažurirana brzina nije bliska nuli. Kvalitet čestica mjeri se pomoću fitnes funkcije koja utiče na optimalnost odgovarajućeg rješenja.

### A. Korištene fitnes funkcije

U kontekstu klasifikacije slika jedna čestica predstavlja $N_c$ centar klastera. To znači da je svaka čestica $x_i$ kreirana kao:

$$x_i = (m_{i1}, m_{i2}, ..., m_{ij}, ..., m_{iN_c}) \quad (7)$$

gdje se $m_{ij}$ odnosi na $j$-ti klaster centroid vektora $i$-te čestice.

U sklopu rada su implementirane 3 fitnes funkcije pogodne za rješavanje problema klasteringa. Prva funkcija je predložena u [2] i njen cijl je minimizirati maksimalnu prosječnu udaljenost unutar klastera, te maksimizirati udaljenosti između klastera. Funkcija, odnosno kvalitet svake čestice je opisan jednačinom:

$$f(x_i, Z) = w_1\bar{d}_{max}(Z, x_i) + w_2(z_{max} - d_{min}(x_i)) \quad (8)$$

gdje je $z_{max}$ maksimalna vrijednost piksela na slici (npr. $z_{max} = 2^s - 1$ za s-bitnu sliku), Z je matrica koja predstavlja dodjelu piksela klasterima čestice i. Svaki element $z_{ijp}$ indicira da li piksel $z_p$ pripada klasteru $C_{ij}$, čestice $i$. Konstante $w_1$ i $w_2$ su korisnički definisane konstante. Određivanje maksimalne prosječne Euklidove distance od čestica do vlastitih pridruženih klasa prikazano je izrazom:

$$\bar{d}_{max}(Z, x_i) = \max_{j=1,...,N_c} \{ \sum_{\forall z_p \in C_{ij}} d(z_p, m_{ij})/|C_{ij}| \} \quad (9)$$

dok se minimalna Euklidova distanca između bilo kojeg para klastera određuje izrazom:

$$d_{min}(x_i) = \min_{\forall j1,j2,j1 \neq j2} d(m_{ij1}, m_{ij2}) \quad (10)$$

Prethodno objašnjena fitnes funkcija ima za cilj istovremeno minimizati udaljenost između piksela i njihovih centara klastera, kao i maksimizirati udaljenost između parova klastera, što je postignuto određivanjem $\bar{d}_{max}(Z, x_i)$ i $d_{min}(x_i)$

Druga fitnes funkcija predstavlja modifikaciju fitnes funkcije predložene u [7], a definirana je kao količnik sume kvadratnih grešaka (engl. *SSE - Sum Squared Error*) i ukupne udaljenosti centara klastera, što je prikazano u nastavku:

$$f(x_i, Z) = \frac{\sum_{j=1}^{N_c} \sum_{\forall z_p \in C_{ij}} ||m_{ij} - z_p||^2}{\sum_{\forall j_1, j_2, j_1 \neq j_2} ||m_{ij1} - m_{ij2}||^2} \tag{11}$$

Treća fitnes funkcija je predložena u [6] i koristi isključivo sumu kvadratnih grešaka za evaluaciju rješenja. Funkcija je opisana sa:

$$f(x_i, Z) = \sum_{j=1}^{N_c} \sum_{\forall z_p \in C_{ij}}^{N_c} ||m_{ij} - z_p||^2 \tag{12}$$

Da bismo fitnes funkcije osigurali od praznih klastera, njihovu sumu kvadratnih grešaka postavljamo na znatno veću vrijednost. Navedene fitnes funkcije su primjenjene i uspoređene, o čemu će biti više riječi u narednom poglavlju.

## III. Simulacijski rezultati

Kao što je navedeno u poglavlju 2, u sklopu rada su implementirane dvije verzije algoritma. Svi algoritmi su implementirani u jeziku *Python* [10]. U okviru funkcije $PSO1$ implementirana je verzija algoritma koja ima fiksne parametre za svaku iteraciju, dok funkcija $PSO2$ za parametre koristi formule 4 i 5. Obje funkcije primaju iste parametre. Parametri algoritma su:

- Fitnes funkcija
- Slika nad kojom se izvršava algoritam
- Broj čestica
- Broj iteracija
- Broj klastera
- Niz parametara (w,$c_1$,$c_2$)

Izgled poziva bilo koje od navedenih funkcija PSO algoritma prikazan je u nastavku:

```
rezultat = PSO(f, slika, brojCestica,
      brojIteracija, brojKlastera, P)
```
(13)

Fitnes funkcija $f$ treba biti definirana tako da kao parametre prima česticu (niz centara klastera) i sliku čiji se pikseli dijele u klastere te vraća numeričku vrijednost. Algoritam je namijenjen za rad sa sivonijansiranim slikama dubine 8 bita. Broj čestica, broj iteracija i broj klastera su pozitivni cijeli brojevi. Posljednji parametar je niz od 3 cijela broja koji predstavljaju parametre algoritma. Funkcija $PSO$ vraća najbolje do tada pronađeno rješenje u odnosu na fitnes funkciju. Prvi korak PSO algoritma predstavlja inicijalizacija čestica i njima odgovarajućih brzina. Čestice su inicijalizirane tako da su centri klastera nasumično odabrani brojevi iz intervala $[dubinaMin, dubinaMax]$, gdje $dubinaMin$ i $dubinaMax$ predstavljaju najmanju i najveću dubinu na slici respektivno. Za brzinu svake čestice je za svaku od koordinata uzet nasumično odabran broj iz intervala$[0, 5]$. Pored čestica i brzina, inicijalizirana je i pozicija najbolje čestice čitavog roja, kao i najbolja pozicija za svaku česticu. Nakon inicijalizacije navedenih vrijednosti se u svakoj iteraciji za svaku česticu vrši računanje nove pozicije i brzine na osnovu jednačina 3 i 4. Nakon toga, izvršeno je ažuriranje globalnog rješenja, u slučaju pronalaska rješenja boljeg od trenutnog globalnog rješenja u odnosu na fitnes funkciju. Također, u svakoj iteraciji izvršena je provjera da li nova pozicija izlazi van opsega slike, te se u slučaju prekoračenja opsega preskaču navedena ažuriranja. Algoritam se zaustavlja nakon što je izvršen broj iteracija proslijeđen kao parametar funkcije. Razlika funkcije $PSO2$ u odnosu na $PSO1$ je što se prije ažuriranja pozicije čestica izračunavaju potrebni parametri za svaku od iteracija. U nastavku je prikazan pseudokod osnovnog PSO algoritma.

**Algorithm 1** Algoritam rojem čestica (PSO algorithm)

```
x = []
v = []
for svaku i-tu česticu do
    x[i] = niz od brojKlastera random vrijednost iz [dubinaMin, dubinaMax]
    v[i] = niz od brojKlastera random vrijednost od 0 do 5
end for
pb = x
gb = najbolja čestica iz roja
while iter < brojIteracija do
    for svaku i-tu česticu do
        v_novo =Računanje nove brzine prema relaciji 4
        x_novo = Računanje nove pozicije prema relaciji 3
        if x_novo van opsega then
            continue
        end if
        f_x = f(x_novo, slika)
        if f_x < f(pb[i], slika) then
            pb[i] = x_novo
        end if
        if f_x < f(gb, slika) then
            gb = x_novo
        end if
        x[i] = x_novo
        v[i] = v_novo
        iter+=1
    end for
end while
return gb
```

Obzirom da se pripadnost piksela određenom klasteru računa isključivo na osnovu intenziteta piksela, svi pikseli istog intenziteta pripadaju istom klasteru. U cilju bržeg izračunavanja fitnes funkcija, umjesto dodjele svakog pojedinačnog piksela određenom klasteru vrši se dodjela konkretnih intenziteta svakom klasteru. Ukupnu udaljenost u svakom klasteru je moguće izračunati na osnovu njemu

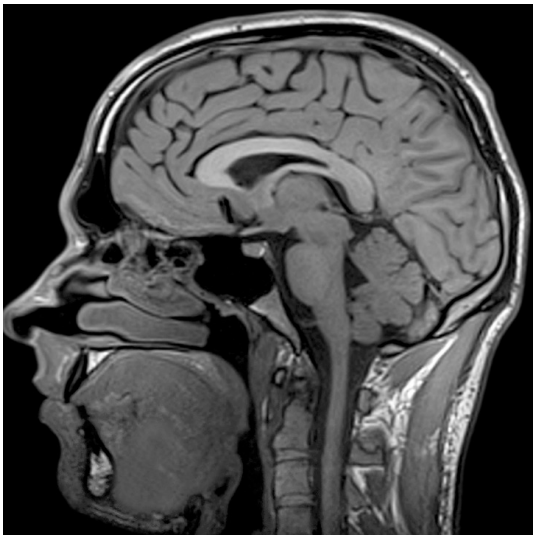
pridruženih intenziteta i histograma slike.

Funkcija `prosjecneUdaljenostiPoKlasterima` vrši dodjelu intenziteta svakom od klastera te na osnovu toga računaju prosječne udaljenosti za svaki od klastera na način prikazan u desnoj strani jednačine 9.

Funkcija `minimalnaUdaljenostIzmedjuKlastera` računa najmanju udaljenost između klastera, primjenom jednačine 10, dok se u funkciji `maksimalnaUdaljenost` bira najveća udaljenost iz niza prosječnih udaljenosti po klasterima, što je objašnjeno u jednačini 9.

`funkcijaDistance1`, `funkcijaDistance2` i `funkcijaDistance3` vrše proračun distanci opisanih jednačinama 8, 11 i 12, respektivno.

Algoritimi su testirani na slikama magnetne rezonance glave i magnetnoj rezonanci mozga prikazanim na 1 i respektivno. [11] .



Slika 1. Magnetna rezonanca glave

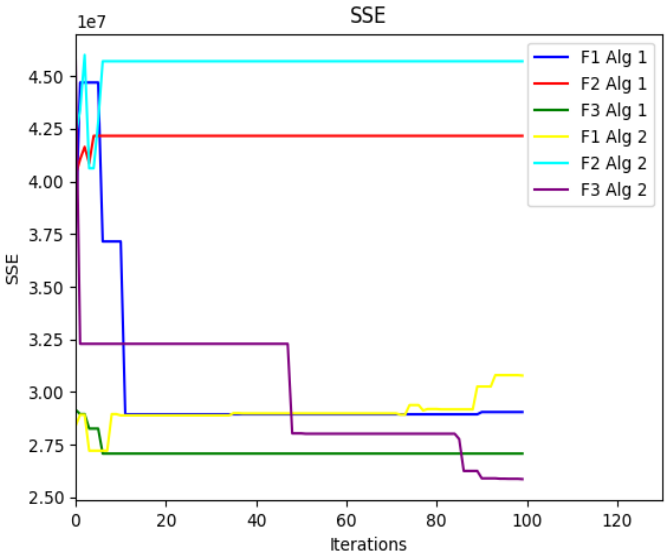
Prilikom testiranja algoritama korištene su dvije različite mjere kvaliteta klasteringa. Za prvu mjeru kvaliteta klasteringa odabrana je suma kvadratnih grešaka kao jedna od najpoznatijih mjera [12]. Druga mjera je predložena u [2] i opisana je jednacinom 14.

$$J_e = \frac{\sum_{j=1}^{N_c}[\sum_{\forall z_p \in C_j} ||m_j - z_p||]/|C_j|}{N_c} \qquad (14)$$

Oba algoritma su testirana u kombinaciji sa sve tri predložene fitnes funkcije. Prilikom svakog od poziva su korišteni sljedeći parametri :

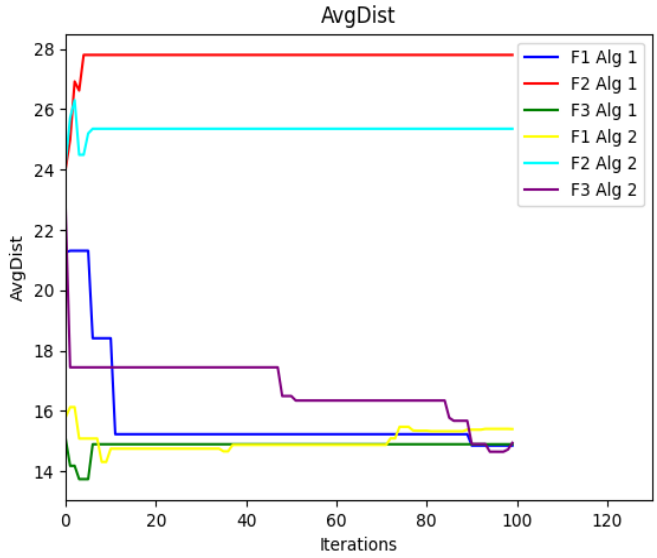
- brojČestica =30
- brojIteracije = 100
- brojKlastera = 3
- w = 0.9
- $c_1 = 2$
- $c_2 = 2$

Za parametre prve fitnes funkcije $w_1$ i $w_2$ su korištene vrijednosti 5 i 0.1 respektivno. Fitnes funkcije su na slikama 2. , 3. i 4. označene kao $F1$ , $F2$ i $F3$, a algoritmi sa $Alg1$ i $Alg2$. Prilikom testiranja su za svaku od 100 iteracija praćene mjere kvaliteta klasteringa čestica sa najboljom pozicijim. Na slici 2 je za svih 6 kombinacija prikazana suma kvadratnih grešaka.

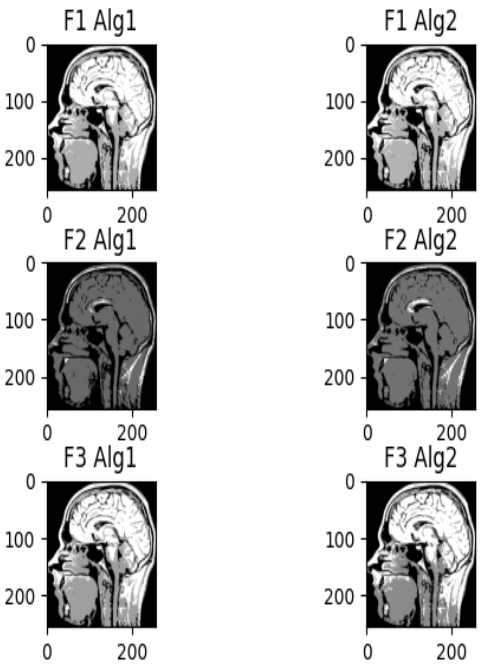



Slika 2. Grafik sume kvadratnih grešaka (SSE)

Na osnovu slike 2. možemo zaključiti da treća fitnes funkcija ($F3$) daje najbolje rezultate pri posmatranju sume kvadratnih grešaka. Ovo ponašanje je očekivano obzirom da je cilj druge funkcije isključivo minimizacija ove greške. Također, možemo primjetiti da prva fitnes funkcija ($F1$) daje bolje rezultate u odnosu na drugu fitnes funkciju ($F2$). Druga mjera klasteringa za svaku od iteracija prikazana je na slici 3. Kao i za sumu kvadratnih grešaka, druga fitnes funkcija daje lošije rezultate u odnosu na ostale fitnes funkcije. Na osnovu korištenih mjera klasteringa se može zaključiti da rezultat više ovisi o izboru fitnes funkcije nego od korištene verzije algoritma, te da su prva i treća fitnes funkcija bolji izbor u odnosu na drugu funkciju. Rezultati klasteringa za svih 6 kombinacija fitnes funkcija i algoritama prikazani su na slici 4., tako što su intenziteti svih piksela iz jednog klastera zamijenjeni sa intenzitetom centra klastera. Na slici 4. možemo vidjeti da je rezultat klasteringa prve i teće funkcije obje verzije algoritma sličan, te da se znatno razlikuje od rezultata dobijenih korištenjem druge fitnes funkcije. Iako se pri segmentaciji slike 1 prva fitnes funkcija pokazala kao bolja, korištenje druge fitnes funkcije za sliku 5 je dalo bolje rezultate u odnosu na prvu fitnes funkciju, što je prikazano na slici . Bitno je naglasiti da se rezultati segmentacije slike korištenjem prve fitnes funkcije u sklopu obje verzije algorita

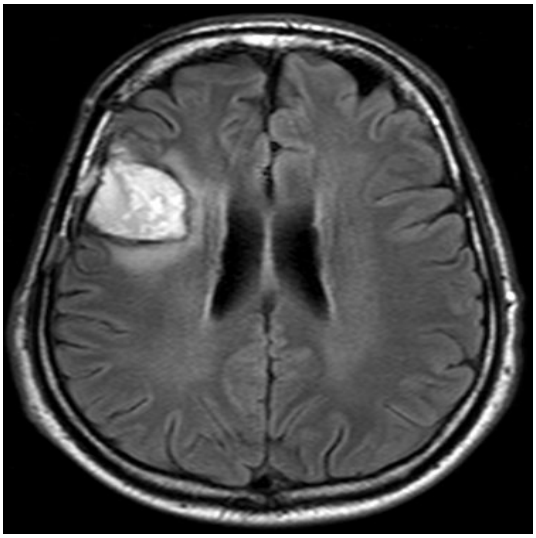



Slika 3. Grafik prosječne udaljednosti

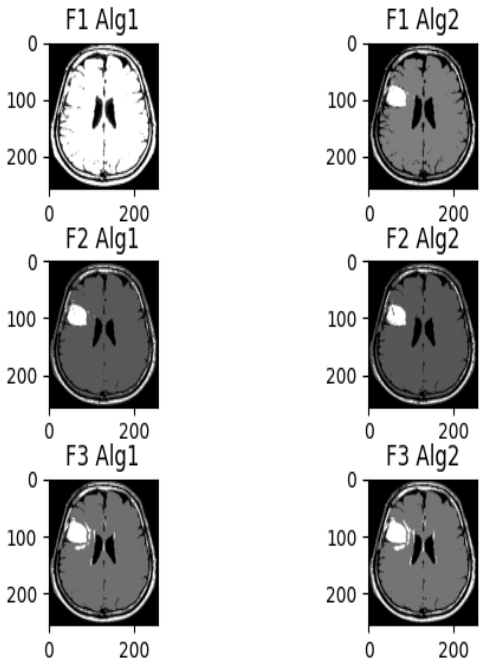



Slika 4. Prikaz kombinacije fitnes funkcija i algoritama na magnetnoj rezonanci glave

značajno razlikuju pri izboru različitih početnih tačaka, dok to nije slučaj sa drugom i trećom fitnes funkcijom.



Slika 5. MRI mozga




Slika 6. Prikaz kombinacije fitnes funkcija i algoritama na MRA mozga

## IV. Zaključak

U sklopu rada je izvršena klasterizacija slika primjenom algoritma roja čestica, pri čemu su implementirane tri različite fitnes funkcije. S obzirom na raznolikost u rezultatima, zaključujemo da ne postoji mehanizam u odabiru idealne fitnes funkcije, jer izbor pogodne fitnes funkcije ovisi o vrsti problema. Prednost PSO algoritam u odnosu na standardne klastering metode je to što zahvaljujući velikom broju čestica ima manje izražen problem izbora početnih tačaka i preuranjenje

konvergencije, iako su ti problemi prisutni.

Buduća poboljšanja PSO algoritma za segmentaciju slika mogu uključivati dodatna istraživanja u vezi izbora odgovarajuće fitnes funkcije i parametara algoritma za domenski specifičnu primjenu segmentacije slike. Osim izbora fitnes funkcije, problem vrijedan posmatranja je i izbor broja klastera, gdje bi se broj klastera određivao dinamički.

## LITERATURA


[1] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing." (2002).

[2] Omran, Mahamed G., Andries P. Engelbrecht, and Ayed Salman. "Image classification using particle swarm optimization." In Recent advances in simulated evolution and learning, pp. 2004.

[3] Kim, Iuliia, Anastasiia Matveeva, Ilya Viksnin, and Igor Kotenko. "Image Clustering Method based on Particle Swarm Optimization." In 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, 2018.

[4] Ghorpade-Aher, Jayshree, and Vishakha A. Metre. "Clustering multidimensional data with PSO based algorithm." arXiv preprint arXiv:1402.6428 (2014).

[5] Peng, Xia, Yao Lin, and Li-Hua Zhang. "An improved pso-fcm algorithm for image segmentation." In IOP conference series: earth and environmental science, vol. 267, no. 4, p. 042081. IOP Publishing, 2019.

[6] Pambudi, Elindra Ambar, Pulung Nurtantio Andono, and Ricardus Anggi Pramunendar. "Image segmentation analysis based on K-means PSO by using three distance measures." ICTACT J. Image Video Process. 9, no. 1 (2018): 1821-1826.

[7] Semchedine, Moussa, and Abdelouahab Moussaoui. "An efficient particle swarm optimization for MRI fuzzy segmentation." (2018).

[8] Beni, G., Wang, J. (1993). "Swarm Intelligence in Cellular Robotic Systems". Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989). Berlin, Heidelberg: Springer. pp. 703–712

[9] Samim Konjicija, Predavanja na predmetu Optimizacija resursa, Elektrotehnički fakultet, Univerzitet u Sarajevu, ak. 2020/2021. godina

[10] Van Rossum, G. Drake, F.L., 2009. Python 3 Reference Manual, Scotts Valley, CA: CreateSpace

[11] McNee, C. (2019, June 14). Head MRI. AffordableMRI.Com. *https://affordablemri.com/head-mri/* Accessed 15.02.2021.

[12] Dženana Đonko, Predavanja na predmetu Mašinsko učenje, Elektrotehnički fakultet, Univerzitet u Sarajevu, ak. 2020/2021

[13] Nath, Siddhartha Sankar, Girish Mishra, Jajnyaseni Kar, Sayan Chakraborty, and Nilanjan Dey. "A survey of image classification methods and techniques." In 2014 International conference on control, instrumentation, communication and computational technologies (ICCICCT), pp. 554-557. IEEE, 2014.

[14] López-Franco, Carlos, Luis Villavicencio, Nancy Arana-Daniel, and Alma Y. Alanis. "Image classification using PSO-SVM and an RGB-D sensor." Mathematical Problems in Engineering 2014 (2014).

[15] Chakrabarty, Navoneel. "Brain MRI Images for Brain Tumor Detection." Kaggle, April 14, 2019. *https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection.* [Accessed 13.02.2021]