

# Raspoznavanje emocija iz izraza lica pomoću radijalne neuronske mreže

---

Štiglec, Mislav

Master's thesis / Diplomski rad

2023

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:720498>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-05**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

RASPOZNAVANJE EMOCIJA IZ IZRAZA LICA POMOĆU  
RADIJALNE NEURONSKE MREŽE

Diplomski rad

Mislav Štiglec

Osijek, 2023.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit**

Osijek, 11.07.2023.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za diplomski ispit**

<b>Ime i prezime Pristupnika:</b>	Mislav Štiglec
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Računarstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	D-1252R, 07.10.2021.
<b>OIB studenta:</b>	80758295293
<b>Mentor:</b>	doc. dr. sc. Dražen Bajec
<b>Sumentor:</b>	dr. sc. Mario Dudjak
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	doc. dr. sc. Krešimir Romić
<b>Član Povjerenstva 1:</b>	dr. sc. Mario Dudjak
<b>Član Povjerenstva 2:</b>	doc. dr. sc. Hrvoje Leventić
<b>Naslov diplomskog rada:</b>	Raspoznavanje emocija iz izraza lica pomoću radijalne neuronske mreže
<b>Znanstvena grana diplomskog rada:</b>	<b>Umjetna inteligencija (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	U radu je potrebno opisati problem raspoznavanja emocija iz izraza lica kao problem nadzirane klasifikacije te istaknuti uobičajene postupke za predobradu i učenja iz takvih slika. Uz to, potrebno je opisati algoritam radijalne neuronske mreže kao i njegovu primjenu za rješavanje razmatranog problema. U praktičnom dijelu rada nužno je razviti programsko rješenje koje izdvaja određene značajke iz slika lica te na temelju njih gradi odgovarajući klasifikacijski model radijalne neuronske mreže za rješavanje problema raspoznavanja emocija. Također, potrebno je napraviti eksperimentalnu analizu na skupovima podataka iz literature. Sumentor: Mario Dudjak Tema rezervirana za: Mislav Štiglec
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Vrlo dobar (4)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	11.07.2023.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 14.07.2023.

**Ime i prezime studenta:**

Mislav Štiglec

**Studij:**

Diplomski sveučilišni studij Računarstvo

**Mat. br. studenta, godina upisa:**

D-1252R, 07.10.2021.

**Turnitin podudaranje [%]:**

8

Ovom izjavom izjavljujem da je rad pod nazivom: **Raspoznavanje emocija iz izraza lica pomoću radijalne neuronske mreže**

izrađen pod vodstvom mentora doc. dr. sc. Dražen Bajer

i sumentora dr. sc. Mario Dudjak

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

## Sadržaj

<b>1.</b>	<b>UVOD</b>	<b>1</b>
<b>2.</b>	<b>RASPOZNAVANJE EMOCIJA IZ IZRAZA LICA</b>	<b>3</b>
2.1.	Uvod u problem klasifikacije	4
2.2.	Izazovi pri rješavanju problema	7
2.3.	Pregled literature	8
<b>3.</b>	<b>OBRADA ULAZNIH PODATAKA</b>	<b>10</b>
3.1.	Opis ulaznog skupa podataka	10
3.2.	Izdvajanje značajki iz ulaznog skupa podataka	10
3.2.1.	Algoritam HOG	12
3.2.2.	Algoritam LBP	15
3.2.3.	Algoritam LDA	18
<b>4.</b>	<b>RADIJALNA NEURONSKA MREŽA</b>	<b>20</b>
4.1.	Struktura radijalne neuronske mreže	20
4.2.	Trening i izgradnja klasifikacijskog modela	23
<b>5.</b>	<b>OPIS PROGRAMSKOG RJEŠENJA</b>	<b>25</b>
5.1.	Tijek postupaka	25
5.2.	Algoritmi za izdvajanje značajki	25
5.3.	Postupci izgradnje RBFN	29
5.3.1.	Postupak nasumične izgradnje	29
5.3.2.	Izgradnja pomoću algoritma K-means	30
5.3.3.	Izgradnja pomoću algoritma optimizacije rojem čestica	31
<b>6.</b>	<b>EKSPERIMENTALNA ANALIZA</b>	<b>35</b>
6.1.	Postavke i metodologija eksperimentalne analize	35
6.2.	Rezultati i diskusija	36
6.3.	Usporedba RBFN s ostalim klasifikatorima	39
<b>7.</b>	<b>ZAKLJUČAK</b>	<b>43</b>
	Literatura	44
	Popis slika	47

Popis tablica . . . . .	48
Sažetak . . . . .	49
Abstract . . . . .	50
Životopis . . . . .	51

# 1. UVOD

Problem raspoznavanja emocija iz izraza lica (engl. *facial emotion recognition*, FER) došao je u središte istraživačke pozornosti s razvojem strojnog učenja. Strojno učenje bavi se proučavanjem i oblikovanjem algoritama koji grade modele problema iz različitih domena na temelju empirijskih podataka. Ti se modeli mogu koristiti za obavljanje raznih zadataka, kao što su predviđanje, donošenje odluka te raspoznavanje uzoraka. Problemi klasifikacije predstavljaju istaknut razred problema u strojnom učenju jer se primjena algoritama strojnog učenja često svodi na izvođenje klasifikacije dostupnih podataka. Klasifikacija može biti korisna kada se nekakvi ulazni podaci žele kategorizirati u unaprijed određeni broj skupina (grupa ili klasa). Međutim, izgradnja učinkovitih klasifikacijskih algoritama nije lagan zadatak jer uvelike ovisi o vrsti i sadržaju podataka pomoću kojih je opisan problem. S obzirom na velik broj različitih unutarnjih karakteristika podataka koje otežavaju izgradnju učinkovitih modela, ne čudi što je predloženo pregršt algoritama za klasifikaciju. No, svaki od ovih algoritama radi drugačije te u pravilu nije unaprijed poznato koji algoritam radi najbolje za dani problem. Ipak, u literaturi postoje brojna istraživanja u kojima su vrednovani i uspoređeni razni algoritmi za različite probleme.

Jedan od poznatih problema klasifikacije jest raspoznavanje emocije čovjeka iz njegova izraza lica, koji je detaljno obrađen u ovom radu. Automatska klasifikacija emocija iz izraza lica može imati brojne primjene kao što je povećanje sigurnosti na javnim mjestima (primjerice, aerodrom i trgovački centar) ili otkrivanje psihičkih oboljenja u psihijatrijskim ustanovama. Za izgradnju klasifikacijskog modela u ovome radu korišten je algoritam radijalne neuronske mreže (engl. *radial basis function network*, RBFN). Opisan je princip rada takvog algoritma, njegovi parametri i struktura. Brojna istraživanja [1, 2] upućuju na prikladnost algoritma radijalne neuronske mreže za rješavanje problema klasifikacije emocija iz izraza lica. Stoga je u ovom radu u tu svrhu testiran algoritam RBFN, gdje je primarni naglasak na pronalaženju prikladnog načina izgradnje modela ovog algoritma kada se on koristi za raspoznavanje emocija iz izraza lica.

Osim algoritma za klasifikaciju, za učinkovitu izgradnju klasifikacijskog modela potreban je i skup podataka koji opisuje zadani problem. Skupovi podataka koji opisuju problem raspoznavanja emocija iz izraza lica najčešće sadrže slike ili videozapise. S obzirom na to da slika i videozapis nisu tipovi podataka s kojima velik broj algoritama strojnog učenja može raditi (primjerice, RBFN) potrebno ih je obraditi prije korištenja. Postoje različiti postupci kojima se iz slika izdvajaju značajke koje nose bitnu informaciju za rješavanje zadanog problema. Takve se značajke mogu uobličiti u strukture prigodne za rad s algoritmima strojnog učenja. Neki od postupaka izdvajanja značajki iz slike detaljnije su objašnjeni u ovome radu. Također, objašnjena je važnost takvih postupaka za cjelokupni proces strojnog učenja.

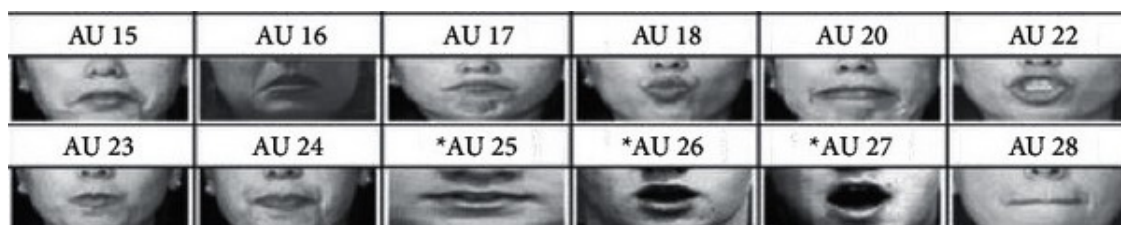
U drugome poglavlju opisan je problem raspoznavanja emocija iz izraza lica kao problem klasifikacije te je dan kratki pregled literature. Uz to su navedene i objašnjene mjere korištene za vrednovanje performansi klasifikacijskih modela izgrađenih prilikom eksperimenta.

U trećem poglavlju naglasak je na opisu ulaznog skupa podataka i njegovoj predobradi. Opisani su različiti postupci korišteni za izdvajanje značajki iz početnog skupa podataka. Četvrto poglavlje naglasak stavlja na radijalnu neuronsku mrežu. Dan je opis neuronskih mreža, njihova struktura i značaj u grani strojnog učenja, a detaljnije je pojašnjena struktura radijalne neuronske mreže. Peto poglavlje opisuje izrađeno programsko rješenje i prikazuje tijek postupaka provedenih kako bi se izgradio klasifikacijski model. Navedeni su dijelovi programskog koda kojim su implementirani postupci izdvajanja značajki. Uz to su opisana tri različita algoritma korištena za izgradnju klasifikacijskog modela. U šestome poglavlju opisan je provedeni eksperiment koji služi za usporedbu spomenutih algoritama. Osim usporedbe klasifikacijskih algoritama, usporedili su se i rezultati konačnog klasifikacijskog modela izgrađenog pomoću algoritma radijalne neuronske mreže s drugim modelima izgrađenih pomoću drugačijih klasifikacijskih algoritama.



## 2. RASPOZNAVANJE EMOCIJA IZ IZRAZA LICA

Raspoznavanje emocija iz izraza lica može se shvatiti kao problem koji nije moguće lako i brzo riješiti. Novija istraživanja definiraju čak 27 emocija koje čovjek može izraziti, a među njima se može izdvojiti šest temeljnih emocija: sreću, tugu, strah, ljutnju, iznenađenje te gađenje [3]. U ovome radu, fokus je na temeljnim emocijama gdje se u obzir uzima i kada osoba ne prikazuje nikakvu emociju (odnosno, osoba je neutralna). Ljudska emocija problematičan je pojam jer ju je gotovo nemoguće generalno opisati ili definirati. Svaka osoba može biti drukčije pogođena istom situacijom te zbog toga imati drugačiji izraz lica sukladno svojim osjećajima. Izraz lica nije jedini način iskazivanja emocija, ali prema Aleksicu [4], 55% izražaja emocija pokazuje se preko izraza lica. Primjer se može vidjeti na slici 2.1 gdje je prikazan samo jedan dio lica iste osobe prilikom osjećanja različitih emocija (pretpostavljeno je da se osoba nalazi u različitim životnim situacijama). Svaka od tih situacija potiče u osobi drugačije emocije, a time i drugačiji izraz lica. Za svaku sliku mogla bi se odrediti emocija osobe samo na temelju oblika jednog dijela njezina lica (primjer AU 15 na slici 2.1 prikazivao bi osjećaj tuge), pa se tako može zaključiti da lice nosi veliku količinu informacije kada je u pitanju raspoznavanje čovjekove emocije. Bez obzira na rečeno, postoje mnogi problemi prilikom određivanja čovjekove emocije. Uzmimo primjer AU 15 na slici 2.1 i recimo da svi ispitanici odrede emociju tuge za osobu s tim položajem usta. Mišljenje većine prihvatilo bi se kao važeće i zaključili bi da se ta osoba osjeća tužno, ali osjeća li se ta osoba zaista tužno može istinito reći samo ta osoba. Upravo zbog navedenih izazova FER je posljednjih desetak godina zadobio znatnu istraživačku pozornost.

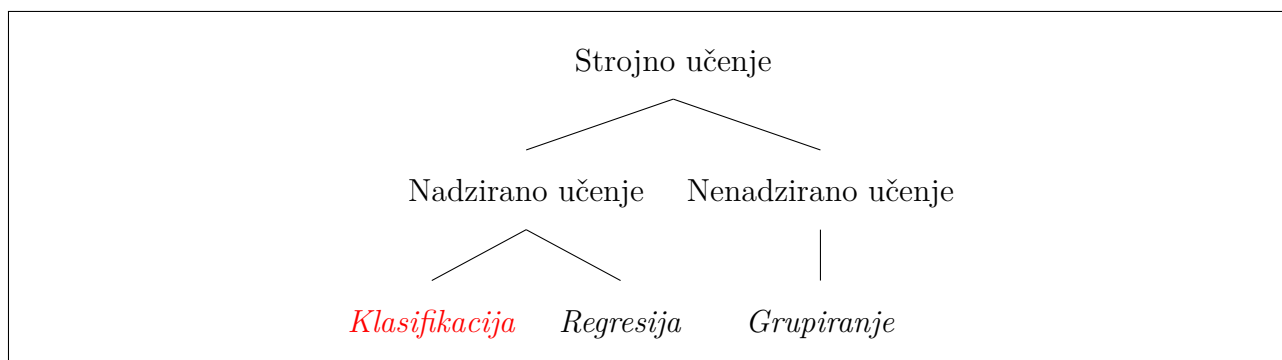


**Slika 2.1:** *Prikaz pozicije dijela lica ovisno o osjećaju različitih emocija [5]*

Raspoznavanje emocija iz izraza lica zanimljiv je problem koji se pojavljuje u mnogim područjima znanosti kao što su psihologija, medicina, pravo, biometrija, javna sigurnost i ekonomija [6]. Zanimljiv primjer primjene i korisnosti klasifikacijskog modela izgrađenog kako bi riješio problem FER je raspoznavanje emocija iz izraza lica putnika prilikom ukrcavanja na aerodromu. Naime, u brojnim državama postoji velik broj trgovanja drogom, lijekovima i općenito zabranjenim supstancama. Raspoznavanjem emocije mogu se utvrditi potencijalni kriminalci zbog specifičnog ponašanja i izgleda lica [7]. Drugi primjer može biti raspoznavanje emocija čovjeka prilikom kupovine. Kako se kupac kreće trgovinom, njegove se emocije mogu mijenjati u odnosu na proizvode oko njega (primjerice, kupac je sretniji na odjelu čokolade jer ga podsjeća na djetinjstvo). Tako trgovina može rasporediti proizvode tako da izgradi okolinu u kojoj se kupac osjeća najzadovoljnije (premissa je da zadovoljan i sretan kupac više troši).

## 2.1. Uvod u problem klasifikacije

S porastom količine podataka posljednjih godina razvile su se metode koje pomažu pri obradi, selekciji, grupiranju, filtriranju i ostalim radnjama nad tim podacima. Jedna od mogućih radnji nad podacima je klasifikacija tj. pridruživanje unaprijed određene oznake klase nekakvom primjerku unutar ulaznog skupa podataka. Broj klasa u koje se može klasificirati podatke određen je samim ulaznim podacima. Razredi problema strojnog učenja obično se dijele na nadzirano i nenadzirano strojno učenje, što se može vidjeti na slici 2.2.



**Slika 2.2:** Razredi problema strojnog učenja

Klasifikacija spada u razred nadziranih problema strojnog učenja jer su oznake ulaznih podataka unaprijed poznate. Ulazni skup podataka može se definirati kao matrica  $X_{N \times d}$  gdje svaki red matrice  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in X$  predstavlja jedan primjerak u skupu podataka. Svaki primjerak opisan je vektorom značajki, gdje broj značajki određuje dimenzionalnost problema. U nadziranom učenju svakome je primjerku pridružena oznaka klase  $l_i \in L$ ,  $i, \dots, m$ , što je ključna razlika između nadziranog i nenadziranog učenja (kod nenadziranog učenja oznake primjeraka nisu poznate). Nakon pridruživanja oznake primjerku, moguće je formirati matricu  $O_{N \times (d+1)}$  gdje svaki red predstavlja uređeni par jednog vektora značajki i njemu pridružene oznake (slika 2.3).

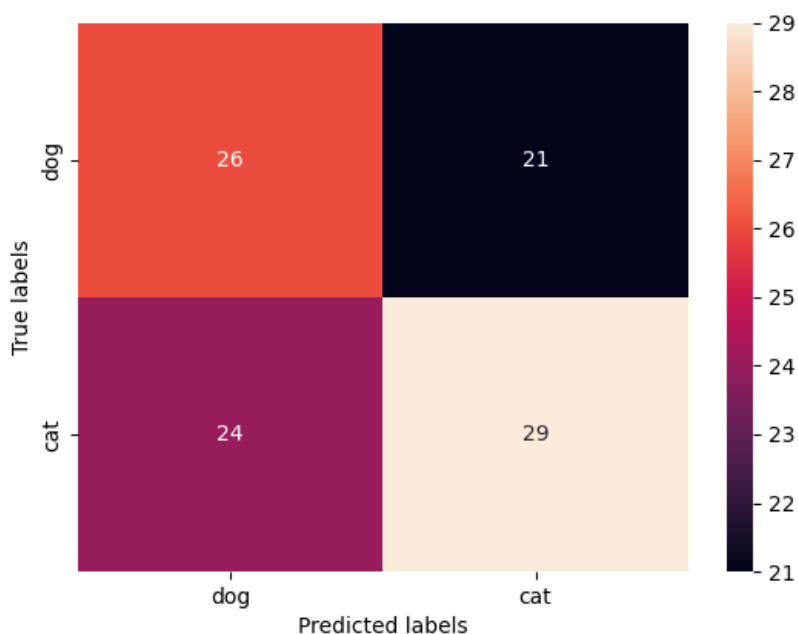
$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Nd} \end{pmatrix} \quad L = \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{pmatrix} \quad O = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & l_1 \\ x_{21} & x_{22} & \dots & x_{2d} & l_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Nd} & l_N \end{pmatrix}$$

**Slika 2.3:** Matrični zapis označenih primjeraka  $O$

Cilj klasifikacije jest izgradnja klasifikacijskog modela na temelju dostupnih podataka pomoću određenog klasifikacijskog algoritma, odnosno klasifikatora (engl. *classifier*) koji sa zadovoljavajućom razinom točnosti pridružuje oznaku klase svakom novom, do sada nepoznatom primjerku. Često se kaže da se klasifikator trenira nad podacima s ciljem izgradnje klasifikacijskog modela. Tako se skup podataka uobičajeno dijeli na podskupove za trening (engl.

*training data*) i test (engl. *test data*). Nakon podjele uzima se podskup podataka za trening te se nad njime pokušava izgraditi model koji je nakon toga testiran na podskupu podataka za testiranje. Prije nego što se pristupi postupku treniranja podataka nekada je potrebno napraviti predobradbu ulaznih podataka zbog potencijalnih problema u podacima koji otežavaju treniranje učinkovitih klasifikatora. Primjerice, može se dogoditi da neki podaci nedostaju te je takve primjerke potrebno izbaciti (moguće je i raznim metodama nadopuniti nedostajuće vrijednosti). Ako su podaci prisutni u različitim oblicima (primjerice, u pisanom i numeričkom obliku) tada je potrebno prebaciti podatke pisanog oblika u onaj numerički (brojevi su lakši za procesiranje pa im se zato daje prednost nad nizovima znakova). Ako postoje velike oscilacije u vrijednostima značajki tada se koristi postupak normalizacije koji sve vrijednosti unutar vektora značajki transformira u određeni interval vrijednosti. Normalizacija se provodi jer velike oscilacije u vrijednostima značajki mogu uzrokovati slabije performanse nekih klasifikatora. Neki od postojećih klasifikatora koji se mogu koristiti za izgradnju modela su nasumične šume (engl. *random forest*, RF), metoda potpornih vektora (engl. *support vector machine*, SVM), klasifikator K-najbližih susjeda (engl. *K-nearest neighbors*, K-NN) i radijalna neuronska mreža (RBFN). Nemoguće je odrediti koji od klasifikatora je najbolji jer svaki od njih može dati drugačije rezultate ovisno o zadanom skupu podataka.

Rezultati klasifikatora mogu se prikazati pomoću matrice zabune (engl. *confusion matrix*) čiji se primjer može vidjeti na slici 2.4. Iz matrice zabune jasno se vidi koliko je puta klasifikator točno predvidio oznaku klase ulaznih primjeraka. Mogu se definirati četiri vrijednosti: broj stvarno pozitivnih (engl. *true positive*,  $T_p$ ), broj stvarno negativnih (engl. *true negative*,  $T_n$ ), broj lažno pozitivnih (engl. *false positive*,  $F_p$ ) i broj lažno negativnih (engl. *false negative*,  $F_n$ ). Navedene vrijednosti koriste se za definiranje različitih mjera uspješnosti klasifikacije. Prva



Slika 2.4: Primjer matrice zabune

i najjednostavnija od mjera koja se može definirati zove se pogreška klasifikacije (engl. *misclassification rate*, MCR). Pogreška klasifikacije govori koliki postotak primjeraka iz ulaznog skupa podataka je izgrađeni klasifikator pogrešno klasificirao, odnosno koliki je omjer između netočno klasificiranih primjeraka i ukupnog broja primjeraka (2.1). U primjeru sa slike 2.4. MCR bi iznosio 45%. Pomoću pogreške klasifikacije može se lako dobiti i točnost klasifikacije (engl. *classification accuracy*, CAC) (2.2).

$$MCR = \frac{F_p + F_n}{T_p + T_n + F_p + F_n} \quad (2.1)$$

$$CAC = 1 - MCR \quad (2.2)$$

Pogreška i točnost klasifikacije jednostavne su mjere točnosti klasifikatora, ali u slučaju da su primjerci jedne klase puno zastupljeniji od primjeraka druge klase one mogu biti nepouzdana. Zbog toga su definirane druge mjere koje pouzdanije vrednuju točnost klasifikatora. Stopa stvarno pozitivnih predviđanja (engl. *true positive rate*, TPR) koja se još naziva mjera osjetljivosti (engl. *sensitivity*) ili mjera odziva (engl. *recall*, Rec) je mjera koja pokazuje postotak točno predviđenih pozitivnih primjeraka (2.3).

$$TPR = \frac{T_p}{T_p + F_n} \quad (2.3)$$

Stopa stvarno negativnih predviđanja (engl. *true negative rate*, TNR) ili mjera specifičnosti (engl. *specificity*) je mjera koja pokazuje postotak točno predviđenih negativnih primjeraka (2.4). Bitno je naglasiti da su stopa stvarno pozitivnih i negativnih predviđanja otporne na neuravnoteženost primjeraka.

$$TNR = \frac{T_n}{T_n + F_p} \quad (2.4)$$

Stopa lažno pozitivnih predviđanja (engl. *false positive rate*, FPR) je mjera koja pokazuje postotak netočno predviđenih primjeraka negativne klase (2.5).

$$FPR = \frac{F_p}{F_p + T_n} \quad (2.5)$$

Mjera preciznosti (engl. *precision*, Pre) pokazuje koliki je udio primjeraka pozitivne klase bio točno predviđen (2.6).

$$Pre = \frac{T_p}{T_p + F_p} \quad (2.6)$$

Do sada je već spomenuto šest različitih mjera kojima se može vrednovati točnost klasifikatora. Korištenje velikog broja mjera može biti nepregledno pri analiziranju podataka. Zbog toga se uobičajeno predlaže objedinjavanje podataka iz matrica zabune u jednu mjeru. Jedna takva mjera zove se F-mjera (engl. *F-score*) koja povezuje preciznost i mjeru odziva. Ako se želi poboljšati rezultat koji daje mjera preciznosti moramo ugroziti rezultat mjere odziva i obrnuto.

Zbog toga F-mjera povezuje mjere preciznosti i odziva te daje rezultat čijim poboljšanjem dolazi i do poboljšavanja obiju mjera. Generički zapis F-mjere može se definirati prema (2.7). gdje se za vrijednost  $\beta$  najčešće uzima 1 (zbog toga se mjera još može nazivati F1-mjera).

$$F1 = \frac{(1 + \beta^2) \cdot Pre \cdot Rec}{(\beta^2 \cdot Pre) + Rec} \quad F1 = \frac{2 \cdot Pre \cdot Rec}{Pre + Rec}, \quad \beta = 1 \quad (2.7)$$

Pomoću navedenih mjera mogu se vrednovati ili uspoređivati kvaliteta tj. točnost izgrađenih klasifikatora i na samom kraju izabrati onaj klasifikator koji daje najbolje rezultate.

## 2.2. Izazovi pri rješavanju problema

Problem raspoznavanja emocija iz izraza lica povezan je s raznim etičkim, pravnim, i tehnološkim izazovima na koje treba obratiti pozornost prilikom njegova rješavanja. Neki od tih izazova su [6]:

1. Zaštita podataka
2. Privatnost podataka i pojedinca
3. Točnost podataka i individualnost
4. Pogreška algoritama zbog medicinskih i etničkih razlika
5. Upotreba izgrađenog klasifikacijskog modela za raspoznavanje emocija iz izraza lica.

Zaštita podataka je problem koji se pojavljuje kada se koriste velike količine osobnih podataka pa tako i pri raspoznavanju emocija. Obrada osobnih podataka nužna je za funkcioniranje određenih aplikacija pa tako i za njihovo unapređenje [8]. Bez obzira na primjene i korisnost tih aplikacija osobni podaci moraju biti sigurni i tvrtke koje koriste te podatke moraju jamčiti za sigurnost tih istih podataka. Kada pričamo o raspoznavanju emocija, potrebni podaci su privatni i osjetljivi za svakog pojedinca (značajke lica). Slike ljudskih lica mogu se zloupotrijebiti na razne načine što uzrokuje problem sigurnosti podataka i povjerenja sudionika pri davanju svojih osobnih podataka. Kao primjer može se navesti skandal koji se dogodio 2010. godine kada je otkriveno da je Facebook dopustio analizu njihovih podataka jednome od instituta na sveučilištu Cambridge bez dopuštenja korisnika.

Privatnost je osnovno ljudsko pravo svakog pojedinca. Davanjem svojih privatnih podataka prepuštamo se određenom riziku. Onaj koji podatke želi analizirati ili obrađivati mora tražiti dopuštenje korisnika ili sudionika kako bi njegove podatke smio koristiti. U slučaju FER-a, načini na koji se podaci mogu skupljati su slika ili videozapis. Kako bi se problem FER mogao riješiti potreban je velik broj slika ili videozapisa ljudi različitih fizičkih, etničkih i dobrih skupina. Najlakši način prikupljanja podataka bilo bi postavljanje kamere na javna mjesta gdje bi se slike ljudskih lica spremale u bazu podataka. Tu dolazi do velikog problema, a to je pristanak pojedinca na obradu njegovih privatnih podataka. Zbog toga je pristup potrebnim

podacima za rješavanje problema FER problematičan u samom početku [9].

Točnost podataka problem je koji nastaje zbog individualnosti pojedinaca i njihovog različitog doživljaja životnih situacija te izražavanja emocija. Klasifikacijski model za raspoznavanje emocija mora biti u stanju raspoznati emociju čovjeka bez obzira na individualne razlike pojedinaca. Isto tako postoji mogućnost osjećanja više emocija istovremeno. Također, osoba može izraziti svoje emocije glasovno ili gestama ruku i nogu, što ne mora nužno izazvati specifičan izraz lica kojime bi mogli raspoznati tu emociju. Nadalje, izraz lica ne mora nužno biti promijenjen zbog osjećaja emocije. Lice se može promijeniti prilikom izricanja sarkazma ili glume. Točnost podataka može biti narušena fizičkim i psihološkim razlikama ljudi te zbog tehničkih uvjeta u kojima se podaci prikupljaju. Točnost slike varira ovisno o tome je li slika uzeta u tamnim uvjetima ili pod utjecajem sunčeve svjetlosti. Nadalje, na slici se mogu nalaziti drugi elementi koji mogu sakriti čovjekovo lice ili dio lica (primjerice, medicinska maska). Također, sam položaj kamere u trenutku uzimanja slike je važan faktor koji utječe na točnost podataka.

Algoritmi za FER mogu imati jako dobre rezultate na testnim podacima. Međutim, dolazi do problema ako je algoritam treniran na podacima koji su uzeti samo iz jedne etničke skupine ljudi. Istraživanja su pokazala kako su algoritmi za FER skloni osobama crne rase pridruživati negativne emocije (ljutnju, gnjev, žalost) [10]. Ovakvi problemi nisu samo prisutni pri raspoznavanju emocija, već i u drugim granama znanosti. Zbog toga je poželjno pri prikupljanju podataka u obzir uzeti što više ljudi različitih etničkih skupina i boje kože. Ako algoritmi ne rade ispravno za sve ljude bez obzira na boju kože može doći do diskriminacije prilikom primjene izgrađenog klasifikacijskog modela.

Izgrađeni klasifikacijski model s mogućnošću raspoznavanja emocije iz izraza lica može se koristiti u različite svrhe, a neke od njih su već nabrojane u prethodnom dijelu rada. Najveći problem upotrebe modela jest individualni pristanak osobe za upotrebu njegovih privatnih podataka. Kako bi klasifikacijski model raspoznao emociju, na neki način mu se mora omogućiti pristup slici lica osobe. S obzirom na to da osoba mora osobno prihvatiti procesiranje njezinih privatnih podataka, nije uvijek moguće napraviti predviđanje zbog nedostupnosti tih podataka. Tvrtke mogu u svojim prostorima tražiti suglasnost korisnika za uzimanje njihovih podataka tijekom logiranja u sustav ili tijekom korištenja određenih usluga te tvrtke. Drugi problem upotrebe nadovezuje se na problem generalizacije koji je ranije spomenut. Osoba može imati izraz lica koji nije uobičajen za određenu situaciju te određivanje njegovog emocionalnog stanja može biti pogrešno.

## 2.3. Pregled literature

Osim preko izraza lica, čovjek emocije može izraziti različitim dijelovima tijela, glasom ili pokretom [11]. Mnoga istraživanja napravljena su kako bi se pronašao najbolji izvor informacija pomoću kojeg bismo mogli odrediti čovjekovu emociju. Kao što je napomenuto u prijašnjim poglavljima, izgrađeni klasifikacijski model s mogućnošću raspoznavanja emocije iz izraza lica

može se primijeniti u mnogim granama znanosti [6]. Zbog toga se zadnjih desetljeća postupak izgradnje klasifikacijskih modela detaljno proučava te je do sada napisano mnoštvo znanstvenih radova i aplikacija koje u svom radu koriste algoritme čijim se treniranjem mogu izgraditi klasifikacijski modeli. Postupak izgradnje klasifikacijskog modela za raspoznavanje emocije iz izraza lica sastoji se od niza koraka gdje svaki može pridonijeti ili ugroziti točnost konačnog modela [5]. Jedan od skupova podataka široko korišten za raspoznavanje emocija jest FER2013 [12]. Većina skupova podataka sadrži prevelik broj značajki što može otežati rad algoritama za izgradnju. Prilikom rada sa slikama to je posebno izraženo te je potrebno pripremiti optimalni podskup značajki koji dovoljno dobro opisuje isti skup slika. Postoje razni načini za izdvajanje značajki, a konačni skupovi izdvojenih značajki uvelike se razlikuju ovisno o primijenjenom algoritmu [5, 13]. Postoje razni algoritmi i tehnike koji se mogu koristiti pri izgradnji klasifikacijskog modela za raspoznavanje emocija iz izraza lica, koji općenito proizlaze iz područja strojnog učenja [5]. Svaki od tih algoritama daje različite performanse ovisno o odabranom skupu podataka s kojim radi, pa nije moguće odrediti najbolji algoritam. U svome radu Dhvani, Siddiqui i Javaid [14] izlažu velik broj različitih kombinacija algoritama strojnog učenja s njihovim rezultatima. Najboljom se pokazala kombinacija algoritma LBP i algoritma K-NN s prosječnom sposobnošću predikcije od 97.16%. Jedan od algoritama kojim se rješavaju problemi FER naziva se radijalna neuronska mreža (engl. *radial basis function neural network*, RBFNN), a njena uspješnost za rješavanje problema FER ispitana je u ovome radu. Jedan primjer korištenja radijalne neuronske mreže za raspoznavanje emocija u svome radu opisuju Agarwal i Bhanot [1]. Za pronalaženje parametara radijalne neuronske mreže tj. broja čvorova skrivenog sloja i centara tih čvorova korišten je algoritam *firefly*. Treniranjem mreže pomoću algoritma *firefly* pokazalo se uspješnijim od treniranja nekim drugim algoritmima kao što su K-means i diskretna kosinusna transformacija (engl. *discrete cosine transform*, DCT). Trenutno postoji velik broj algoritama strojnog učenja koji se koriste pri izgradnji klasifikacijskih modela sposobnih za rješavanje problema FER. Neke primjene napravljene su isključivo za raspoznavanje emocije iz čovjekova lica [15], dok neke složenije aplikacije koriste izgrađene klasifikacijske modele samo kao dio svoje funkcionalnosti.

### 3. OBRADA ULAZNIH PODATAKA

Kako bi se uopće realizirao klasifikacijski model koji ima mogućnost raspoznavanja emocije iz izraza lica mora postojati prikladan skup podataka za treniranje klasifikacijskog algoritma. Skupovi podataka koji opisuju problem FER uobičajeno sadrže slike ili videozapise ljudi koji iskazuju različite emocije. S obzirom na to da algoritam radijalne neuronske mreže u pravilu ne može izravno rukovati sa slikama, potrebno je provesti izdvajanje značajki koje na odgovarajući način opisuju slike. Postoje različiti postupci izdvajanja značajki iz slika radi izrade skupova podataka, gdje svaki ima svoje prednosti i mane, a u ovome radu koriste se algoritmi koji su se u literaturi pokazali prigodnima za rad na ovakvom skupu podataka [14, 13].

#### 3.1. Opis ulaznog skupa podataka

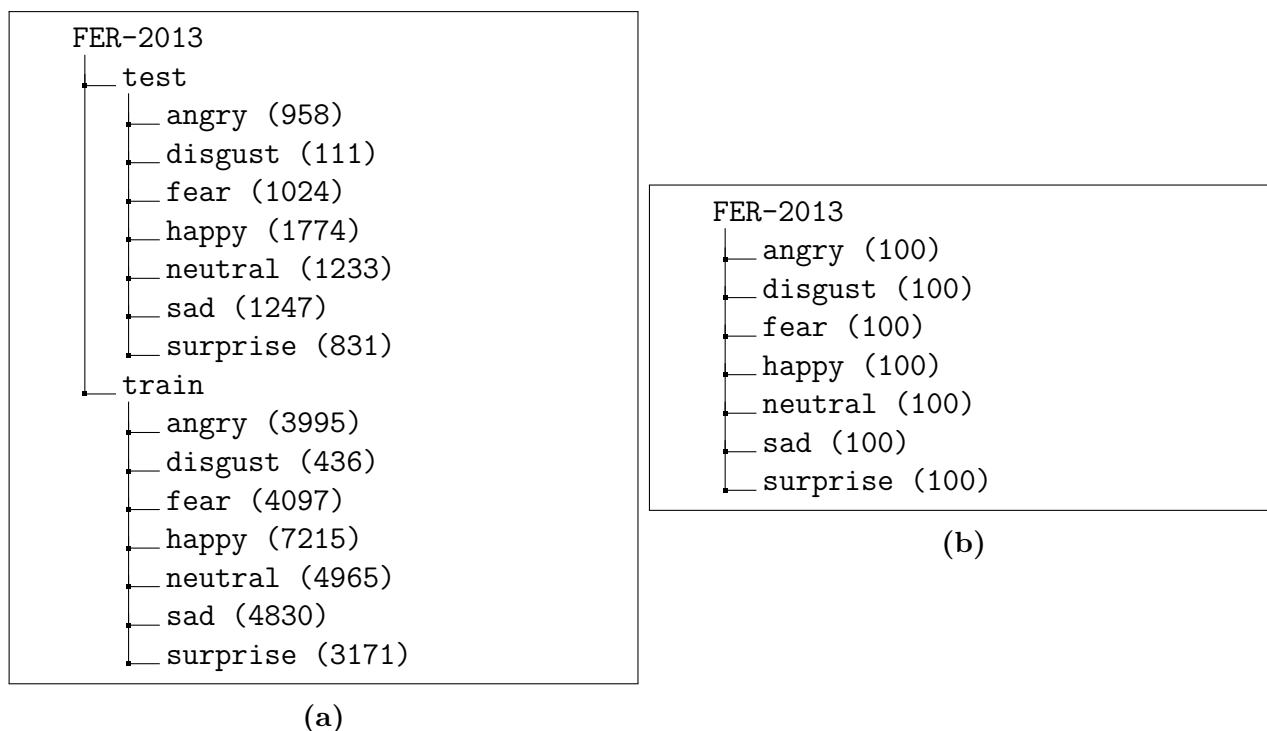
Skup podataka korišten u ovom radu zove se FER2013 i sastoji se od crno-bijelih slika ljudskih (izraza) lica. Sve slike su veličine  $48 \times 48$  piksela i sva lica su centrirana te postavljena tako da zaokupe većinu slike. Podaci su podijeljeni u dva direktorija - za treniranje (engl. *train*) i za test (engl. *test*). Unutar oba direktorija nalazi se još 7 direktorija gdje se nalaze slike izraza lica. Svaki od tih 7 poddirektorija predstavlja jednu od osnovnih ljudskih emocija uz neutralan izraz lica koji se smatra kao sedma emocija. Struktura skupa podataka koja se nalazi na *Kaggle-u* [16] prikazana je na slici 3.1a. U zagradama se nalaze brojevi koji označavaju broj slika unutar pojedinog direktorija. Može se vidjeti da direktorij predviđen za treniranje ima puno više slika nego direktorij za testiranje (raspodjela je u omjeru 88:12%). Također se može vidjeti da neki direktoriji sadrže puno manji broj slika od ostalih što može ugroziti točnost konačnog klasifikacijskog modela. Iz ove analize skupa podataka koji se nalazi na *Kaggle-u* jasno je da on mora biti predobrađen prije nego što se iz njega krenu izdvajati značajke. Iz početnog skupa podataka skinutog s *Kaggle-ove* stranice iz svakog je direktorija izdvojeno samo 100 slika, točnije, 70 slika iz svakog direktorija gdje je u pitanju trening i 30 slika iz svakog direktorija gdje je u pitanju test (ukupno 100 slika za prikaz svake emocije). Taj broj slika dovoljan je za potrebe ovog rada. Struktura novog direktorija korištenog za treniranje algoritma vidljiv je na slici 3.1b. Iz ovakve su strukture direktorija programski razdvojeni testni i trening podaci. Nakon ove procedure uzorkovanja slika smanjen je broj podataka koji ulaze u postupak izdvajanja značajki te je smanjena razlika između broja slika koje prikazuju određenu klasu (emociju).

#### 3.2. Izdvajanje značajki iz ulaznog skupa podataka

Kao što je rečeno u uvodnom dijelu ovoga poglavlja, izdvajanje značajki jest jedan dio cjelokupnog postupka obrade ulaznih podataka kako bi se oni doveli do stanja u kojemu ih algoritmi strojnog učenja mogu procesirati. Algoritmi strojnog učenja najlakše rade s numeričkim vrijednostima, pa se zato iz ulaznih skupova podataka izdvajaju one numeričke vrijednosti koje nose

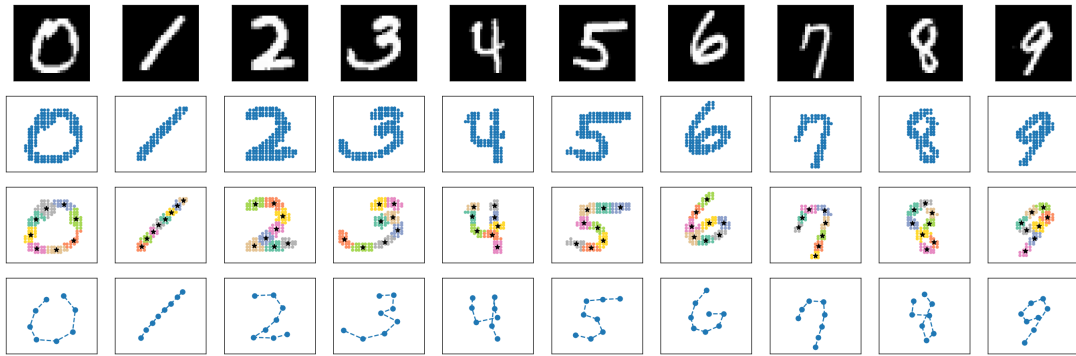


bitnu informaciju za rješavanje zadanog problema. Ako se kao primjer uzme slika čovjekova lica, mogu se odrediti oni dijelovi slike koji nose bitnu informaciju o tome kako se taj čovjek



**Slika 3.1:** Prikaz strukture FER-2013 direktorija prije(a) i poslije prilagodbe(b)

osjeća (koju emociju izražava). Ti dijelovi slike jesu dijelovi lica (oči, nos, usta, obrazi, obrve i slično). Kada su definirani dijelovi slike koji su potrebni kako bi se odredile emocije (one koje nose informaciju), postavlja se pitanje jesu li potrebni drugi dijelovi slike. Ako nisu, mogu se izbaciti iz daljnje obrade jer usporavaju rad algoritma, a potencijalno i pogoršavaju rezultat zbog nevažnih informacija koje sadrže. Zbog tih problema takvi se podaci ne uzimaju u obzir te se iz slike izdvajaju samo značajke koje nose bitnu informaciju. U ovom primjeru to su pikseli slike koji predstavljaju dijelove lica, dok je ostatak slike odbačen. U sljedećim poglavljima opisani su algoritmi korišteni za izdvajanje značajki iz ulaznog skupa podataka. Svaki algoritam može izdvajati značajke na različiti način tj. na temelju različitih kriterija. Neki algoritmi izdvajaju značajke na temelju geometrijskih obilježja dijelova tijela (često se koristi pri radu sa slikama lica jer su otporni na promjenu položaja lica). Postoje algoritmi koji izdvajaju značajke na kojima je prisutna promjena boje, svjetline ili određena predefinirana vrijednost (engl. *threshold*). Ovisno o korištenom algoritmu mijenja se broj značajki i njihov raspon vrijednosti što na samom kraju može utjecati na konačnu točnost klasifikacijskog modela. Još jedan slikoviti primjer izdvajanja značajki može se vidjeti na slici 3.2 gdje je prikazan postupak izdvajanja značajki iz slika brojeva. Bitno je uočiti da bez obzira na to što početni skup podataka (slike brojeva) ima puno veću količinu podataka nego konačni skup koji prikazuje brojeve sa samo 8 točaka on nosi gotovo istu količinu informacije (brojevi se jasno mogu raspoznati prije i nakon izdvajanja značajki).



Slika 3.2: Primjer izdvajanja značajki iz slika brojeva [17]

### 3.2.1. Algoritam HOG

Koncept raspoznavanja ljudi iz njihovih kretnji prvi je opisao Robert K. McConnell 1986. godine [18]. Jedna od prvih primjena ovog algoritma bila je unutar Mitsubishijevog projekta čiji je cilj bio raspoznavanje geste ljudske ruke [19]. Ovaj algoritam danas je široko korišten kao alat za raspoznavanje objekata na slici. Algoritam HOG (histogram orijentiranih gradijenata) sastoji se od nekoliko koraka koji moraju biti napravljeni određenim redoslijedom. U nastavku je svaki korak objašnjen te prikazan na primjeru jedne slike iz FER2013 skupa podataka korištenog u ovom radu. Promjena dimenzija slike prvi je korak koji je potrebno napraviti pri radu s algoritmom HOG. Naime, promjena veličine slike nije nužna, ali se predlaže zbog olakšavanja matematičkih operacija koje se obavljaju u kasnijim fazama ovog algoritma. Predlaže se promjena dimenzija slike tako da širina i visina budu u omjeru 1:2. U ovom radu (unutar FER2013 skupa podataka) sve slike imaju dimenzije  $48 \times 48$  piksela pa se prebacuju u povoljniji omjer. Odabrane su dimenzije 64 piksela za širinu i 128 piksela za visinu kako je predloženo i u originalnom radu [18]. Prikaz slike prije i nakon promjene veličine nalazi se na slici 3.3.



(a) Slika dimenzija  $48 \times 48$  piksela



(b) Slika dimenzija  $64 \times 128$  piksela

Slika 3.3: Slika prije (a) i poslije (b) promjene dimenzije

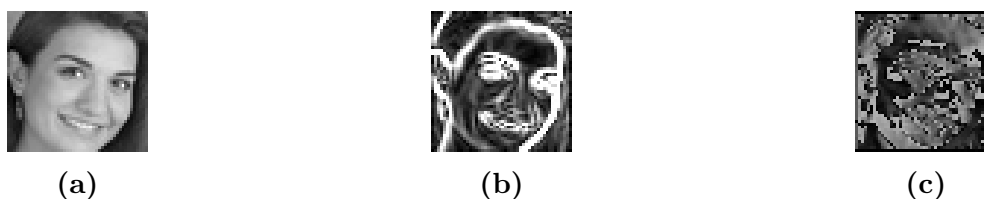
Izračunavanje magnitude i orijentacije gradijenta svakog piksela drugi je i najbitniji dio

algoritma HOG. Prije objašnjavanja izračuna približen je pojam gradijenta piksela. Gradijent piksela definiran je kao promjena intenziteta svjetline slike. Zbog toga gradijenti predstavljaju osnovne elemente mnogih algoritama korištenih u računalnoj obradi slike kao što su cannyev detektor rubova i sobel filter. Pomoću pronalazjenja gradijenata pronalaze se granice objekta tj. odvađa se traženi objekt od ostatka slike (sve unutar granica jest objekt). Gradijent predstavlja promjenu vrijednosti piksela na x i y-osi, a definiran je prema (3.1) [20]. Promjena vrijednosti na x-osi označena je sa  $G_x$  dok je promjena na y-osi označena sa  $G_y$ .

$$\nabla f(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} f(x+1, y) - f(x-1, y) \\ f(x, y+1) - f(x, y-1) \end{bmatrix} \quad (3.1)$$

Za svaki piksel unutar slike izračunat je njegov gradijent. Može se zaključiti da pikseli koji imaju veću razliku u intenzitetu susjednih piksela imaju veću vrijednost gradijenta. Računanjem gradijenata stvorene su dvije nove matrice, jedna gdje su spremljene vrijednosti gradijenata prema x-osi i druga gdje su spremljeni gradijenti prema y-osi. Nakon što su vrijednosti gradijenata poznate za svaki piksel izračunava se vrijednost magnitude i orijentacije za svaki gradijent. Magnituda predstavlja jačinu promjene dok orijentacija predstavlja smjer promjene intenziteta slike. Magnituda i orijentacija izračunate su prema (3.2) gdje  $|r|$  označava magnitudu, a  $\phi$  orijentaciju gradijenta. Na slici 3.4 mogu se vidjeti slike prikazane pomoću magnituda i orijentacija gradijenata piksela.

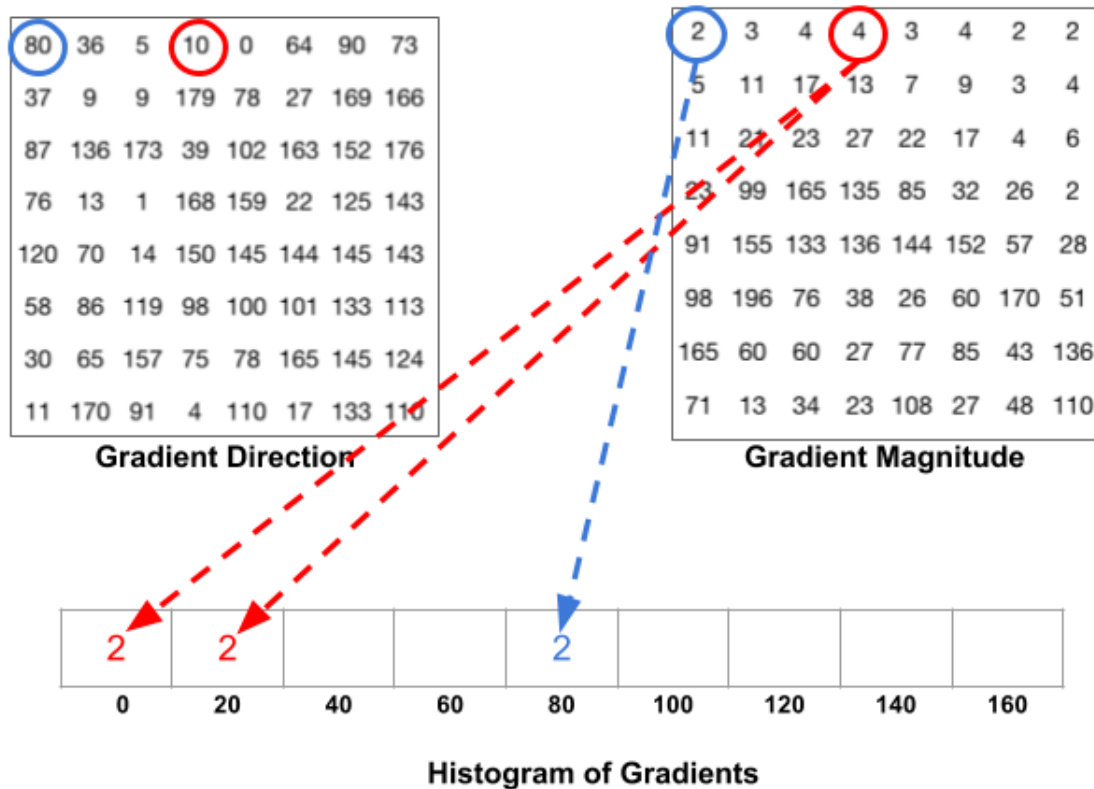
$$|r| = \sqrt{G_x^2 + G_y^2}, \quad \phi = |\tan^{-1}(G_x, G_y)| \quad (3.2)$$



**Slika 3.4:** *Prikaz gradijenata piksela slike(a), magnitude(b) i orijentacije(c)*

Sljedeći korak je podjela slike na manje dijelove, točnije na blokove veličine  $8 \times 8$ . Slika može biti podijeljena i na blokove od  $16 \times 16$  ili  $32 \times 32$ , ali je podjela na što manje blokove bolja zbog veće otpornosti na šumove u slici. Za svaki blok izrađuje se histogram pomoću vrijednosti magnitude i orijentacije gradijenta. Histogram je oblik statističkog prikaza podataka kojim se prikazuje frekvencija pojavljivanja podataka u određenom numeričkom intervalu. Algoritam HOG izrađuje histogram veličine  $9 \times 1$  za svaki prethodno napravljeni blok veličine  $8 \times 8$  [21]. Orijetacija tj. kut gradijenta piksela podijeljena je u rasponu od  $0 - 180^\circ$  stupnjeva. Raspon nije od  $0 - 360^\circ$  stupnjeva zbog toga što se iste negativne i pozitivne vrijednosti predstavljaju istom vrijednošću tj. uzeta je apsolutna vrijednost negativnih stupnjeva (engl. *unsigned*

*gradients*). Tako se interval od  $180^\circ$  stupnjeva dijeli na 9 jednakih dijelova te svaki interval histograma obuhvaća  $20^\circ$  stupnjeva. Histogram se popunjava tako da se na temelju vrijednosti orijentacije gradijenta piksela odabire interval histograma u koji se sprema magnituda gradijenta tog istog piksela. Ako je orijentacija gradijenta na granici između dva intervala, onda je vrijednost magnitude upisana u oba intervala. Opisani postupak može se vidjeti na slici 3.5. Normalizacija je zadnji korak algoritma HOG i potrebna je kako bi se smanjio utjecaj svjetlosti



Slika 3.5: Kreiranje histograma [22]

na vrijednosti histograma. Normalizacija se obavlja u blokovima od  $16 \times 16$  piksela. Blok od  $16 \times 16$  piksela u sebi sadrži 4 bloka od  $8 \times 8$  piksela. Ako se za svaki blok veličine  $8 \times 8$  izradi histogram veličine  $9 \times 1$  tada jedan blok veličine  $16 \times 16$  sadržava vektor veličine  $36 \times 1$ . Tako se normaliziraju vektori veličine  $36 \times 1$ , a ne vektori  $9 \times 1$  što pridonosi kvaliteti normalizacije (bolje je raditi normalizaciju na većim vektorima) [22]. Na kraju normalizacije potrebno je spojiti sve vektore u jedan vektor koji predstavlja značajke slike izdvojene algoritmom HOG. Koliko je konačni vektor dugačak ovisi o odabiru veličine (broju intervala) histograma te veličini bloka na koju je podijeljena slika pri normalizaciji. Početna slika dimenzije je  $64 \times 128$  piksela što znači da u nju stane  $7 \times 15$  blokova veličine  $16 \times 16$ . S obzirom na činjenicu da svaki od blokova sadrži vektor s 36 značajki može se izračunati duljina konačnog vektora, a to je  $7 \times 15 \times 36$  značajki. Na slici 3.6 prikazan je izgled slike na kojoj je primijenjen algoritam HOG u usporedbi s originalnom slikom. Na slici 3.6b može se vidjeti kako su bijelim pikselima označeni dijelovi slike u kojima dolazi do veće promjene intenziteta (primjerice, prijelaz između kose i lica).



(a)



(b)

**Slika 3.6:** Usporedba originalne slike(a) i izlazne slike nakon primjene algoritma HOG(b)

### 3.2.2. Algoritam LBP

Sljedeći algoritam korišten za izdvajanje značajki iz slike korišten u ovom radu je algoritam LBP (engl. *Local Binary Patterns*). Ideja izdvajanja slikovnih značajki kojom se koristi ovaj algoritam opisana je već početkom 90-ih godina prošlog stoljeća, međutim popularnost ovog algoritma povećava se objavom rada [23]. Algoritam ima popularnost zbog postizanja određenih karakteristika kao što su nepromjenjivost (engl. *rotation invariance*) rezultata prilikom zaokretanja slike i otpornosti rezultatnih vektora značajki na promjenu svjetlosti (engl. *luminance invariance*). Prvi korak pri implementaciji ovog algoritma jest pretvaranje slike u boji u crno-bijelu sliku. Skup podataka korišten u ovom radu sadrži crno-bijele slike pa je ovaj korak moguće preskočiti. Sljedeći korak algoritma LBP generira binarni uzorak za svaki piksel slike (u slučaju crno-bijele slike to je vrijednost u intervalu  $[0,255]$ ). Kako bi daljnji postupak mogao biti razumljiv potrebno je upoznati se s ulaznim (i promjenjivim) parametrima ovog algoritma koji utječu na oblik i veličinu izlaznih struktura. U svom radu Ojala [23] definira dva parametra, a to su:

1. Broj točaka (vrijednosti, susjeda) u okolini promatranog piksela čije se vrijednosti uzimaju u obzir prilikom kreiranja binarnog vektora (oznaka: P)
2. Udaljenost (radijus) od promatranog piksela (oznaka: R).

Binarni uzorak generiran je na temelju vrijednosti P točaka koje se nalaze na udaljenosti R od promatranog piksela. Koordinate susjednih točaka računaju se prema (3.3). Promatrani referentni piksel  $g_p$  ima koordinate (0,0). Na slici 3.7 prikazani su položaji susjednih točaka za

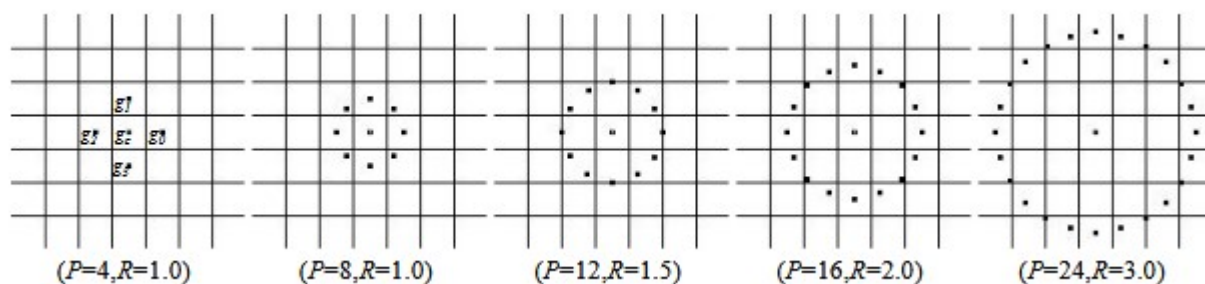
različite vrijednosti parametara P i R.

$$g_c(x, y) = g_c(-R \times \sin(\frac{2\pi p}{P}), R \times \cos(\frac{2\pi p}{P})), \quad p = [0, \dots, P - 1] \quad (3.3)$$

Binarni uzorak izrađuje se tako da se vrijednosti svih susjednih točaka oduzmu od centralnog piksela. Ako je razlika manja od nule (ako je vrijednost susjednog piksela veća) tada je rezultat 0, dok kada je vrijednost susjednog piksela manja tada zapisujemo 1. Tako se mogu odrediti binarne vrijednosti za svaku susjednu točku. Računanje spomenutih vrijednosti opisano je izrazom (3.4) gdje  $x$  predstavlja vrijednost piksela.

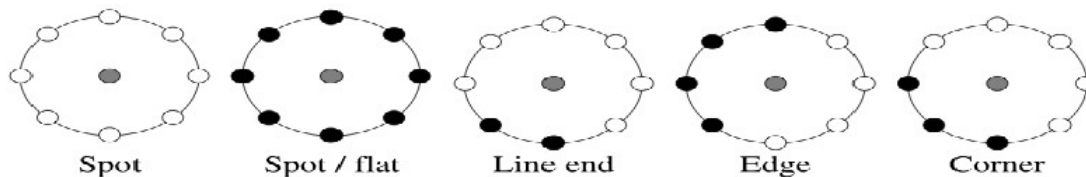
$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3.4)$$

Vrijednosti tada zapisujemo u vektor veličine P kružnim redoslijedom tj. odabiremo početnu točku (može biti bilo koja) i u smjeru kazaljke na satu (ili u obrnutom smjeru) zapisujemo sve ostale vrijednosti. Jedino pravilo koje se ovdje mora poštivati je to da su početna točka od koje se kreće zapisivati u vektor i smjer zapisivanja vrijednosti iste za sve piksele unutar slike. Ako se za primjer uzmu vrijednosti parametara  $(P, R) = (4, 1)$  imamo prvu situaciju sa slike 3.7. U toj situaciji jasno je koje su vrijednosti susjednih piksela, međutim ako se uzme bilo koju drugu kombinaciju parametara s iste slike vidi se kako susjedne točke na radijusu R ne padaju točno na svaki piksel (ne zna se točna vrijednost). Tada se vrijednosti tih točaka računaju interpolacijom. Ovakav pristup generiranja binarnih vektora otporan je na promjenu svjetlosti zbog toga što promjenom svjetlosti ne dolazi do promjene bitova. Zbog toga što su bitovi dobiveni kao rezultat razlike između početnih vrijednosti, a ne kao rezultat samih vrijednosti tada povećanjem ili smanjenjem vrijednosti piksela (povećanjem svjetline povećava se vrijednost crno-bijelih piksela i obrnuto) razlika među njima ostaje ista, pa tako i svi bitovi ostaju jednaki. Nakon što je pronađen binarni vektor za svaki piksel može se generirati histogram pomoću kojega se izrađuje konačni vektor značajki pojedine slike. Svaki binarni vektor predstavlja broječane vrijednosti, a tih brojećanih vrijednosti može biti  $2^P$ . Tako se može generirati histogram koji ima  $2^P$  stupaca i svaki stupac prikazuje koliko ima piksela čija okolina daje istu vrijednost. Na primjer ako 30 piksela ima isti binarni vektor koji kada se prebaci u dekadsku vrijednost iznosi 100, tada



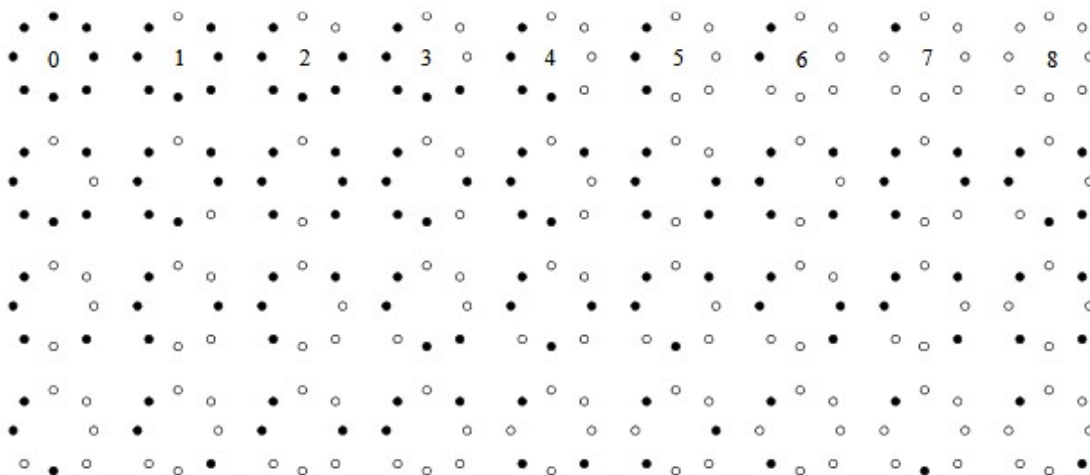
**Slika 3.7:** Kružno raspoređeni susjedi promatranog piksela za različite parametre  $(P, R)$  [23]

stupac 100 u histogramu ima vrijednost 30. Konačni vektor značajki predstavljaju vrijednosti svih stupaca histograma. Ovakav pristup generiranja histograma je ispravan, međutim može biti unaprijeđen ako se iskoristi pravilo uniformnosti. Svaki generirani binarni vektor piksela slike može biti promatran kao uzorak. Neki primjeri uzoraka prikazani su na slici 3.8.



**Slika 3.8:** *Primjeri značenja pojedinih uzoraka (binarnih vektora) [23]*

Prilikom rada na ovom algoritmu i proučavanja binarnih uzoraka došlo je do spoznaje da se neki uzorci pojavljuju u puno većoj mjeri od ostalih [23]. Takvi uzorci nazivaju se uniformnim uzorcima. Za svaki binarni uzorak može se odrediti je li on uniforman ili ne. Uzorak je uniforman ako je moguće pronaći najviše dvije tranzicije između vrijednosti njegovih bitova. Kao primjer uzeti ćemo uniformni vektor bitova  $01111110_2$  i neuniformni vektor bitova  $10010011_2$ . Prvi vektor sadrži tranzicije između 1. i 2. bita te između 7. i 8. bita što su ukupno dvije tranzicije. Drugi vektor sadrži četiri tranzicije što je više od dvije tranzicije i po pravilu on nije uniforman. Na slici 3.9 može se vidjeti 36 različitih uzoraka gdje prvi red prikazuje uniformne uzorke. Primjećujemo da svi uzorci u prvom redu nemaju više od dvije tranzicije između bitova za razliku od ostalih redova. Na temelju premise da je uniformnih uzoraka puno više prilikom izgradnje binarnih vektora može se zaključiti da dolazi do velikog odstupanja u vrijednostima stupaca histograma. Zbog toga se histogram radi tako da se za svaki uniformni uzorak napravi pojedinačni stupac te se svi ostali neuniformni stupci stavljaju u jedan zajednički stupac. Za binarni vektor veličine  $P$  postoji  $P+1$  uniformnih uzoraka (binarni vektor mora biti generiran kružnim dodavanjem podataka). Iz toga slijedi da histogram ili konačni vektor značajki sadrži  $P+2$  značajki. Taj vektor značajki sadrži vrijednosti u intervalu  $[0, \dots, P+2]$  gdje svaka vrijednost predstavlja jednu vrstu uzorka osim vrijednosti  $P+2$  koja predstavlja sve ostale (neuniformne) vrste uzoraka.

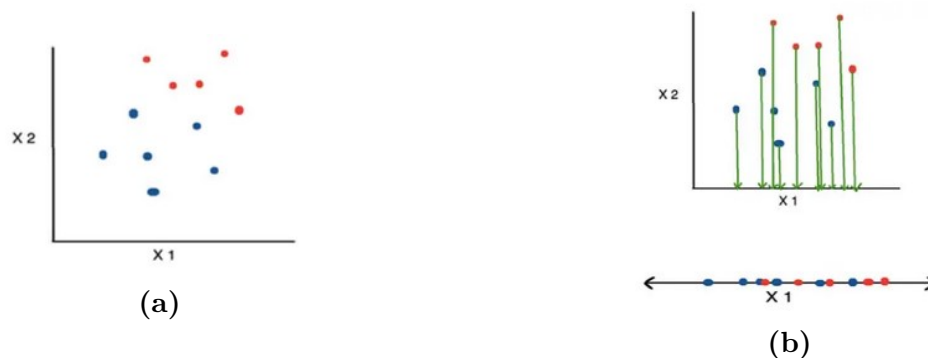


**Slika 3.9:** Prikaz svih mogućih uniformnih uzoraka (prvi red) i neuniformnih uzoraka (ostali redovi) za  $P=8$  [23]

### 3.2.3. Algoritam LDA

Ovaj algoritam prvi je opisao Ronald A. Fisher 1936. godine i nazvao ga linearna diskriminantna analiza (engl. *Linear Discriminant Analysis*) ili fisherova linearna diskriminantna analiza (engl. *Fisher's Discriminant Analysis*) [24]. Fisher je ovaj algoritam opisao na primjeru binarne klasifikacije. Algoritam koji je temeljen na LDA, a rješava problem višeklasne klasifikacije zove se multipla linearna diskriminantna analiza (engl. *Multiple Discriminant Analysis*) i njega je prvi puta opisao Fisherov student Radhakrishna Rao [25]. Algoritam LDA smanjuje dimenzionalnost ulaznih podataka i tako olakšava obradu podataka. Također je bitno naglasiti da je smanjenje dimenzionalnosti potrebno obaviti sa što manjim gubitkom informacije (sve značajke nose određenu količinu informacije te nije moguće samo ih ukloniti). U ovakvim situacijama koristi se algoritam LDA koji smanjuje dimenziju ali pri tome u obzir uzima međusobnu povezanost značajki te maksimalno smanjuje gubitak informacije. Ovaj algoritam koristi se za smanjenje dimenzije pri višeklasnoj klasifikaciji te zbog toga ima brojne primjene u različitim granama znanosti kao što su raspoznavanje emocija iz izraza lica, klasifikacija tretmana liječenja za pojedinog pacijenta, klasifikacija kupaca u trgovini i slično. Algoritam LDA smanjuje dimenziju tako da pokušava projicirati sve vrijednosti u pravac. Moguće je samo odabrati jednu značajku i projicirati njene vrijednosti na pravac ali tu dolazi do velikog gubitka informacije te konačni raspored vrijednosti na pravcu nije pravilno raspoređen. Takva situacija može se vidjeti na slici 3.10.





**Slika 3.10:** Odnos točaka na temelju dvije varijable (a) i odnos točaka zanemarivanjem jedne dimenzije (b) [26]

Na slici 3.10b može se vidjeti pravac koji predstavlja jednodimenzionalan prostor u koji su projicirane točke iz dvodimenzionalnog prostora. Projekcija se realizirala isključivanjem jedne značajke (varijable) te je došlo do gubitka informacije. S pravca na slici 3.10b jasno se može zaključiti da bi klasifikacija predstavljala problem jer ne postoji granica za koju bismo mogli reći da odvajaju vrijednosti dvije klase. Ovakvi problemi mogu se riješiti upotrebom naprednijih algoritama kao što je LDA. Kao što je već ranije rečeno algoritam LDA rješava problem gubitka informacije pri smanjenju dimenzionalnosti. Cilj algoritma jest opisati takav pravac koji uključuje informacije svih značajki prilikom projiciranja točaka. Izrada takvog pravca prati dva kriterija:

1. Udaljenost između srednjih vrijednosti klasa mora biti maksimalna
2. Međusobna udaljenost između točaka koje predstavljaju istu klasu mora biti minimalna.

Na slici 3.11 može se vidjeti generiranje pravca pomoću algoritma LDA. Iz pravca sa slike 3.11b jasno se može odrediti granica između dvije klase što pokazuje da je algoritam LDA očuvao informaciju prilikom smanjenja dimenzije. Zbog ovakvih mogućnosti algoritam LDA i slični algoritmi u širokoj su primjeni kod rada s predobradom podataka u strojnom učenju.



**Slika 3.11:** Novo generirani pravac pomoću algoritma LDA (a) i prikaz novog rasporeda u 1D ravnini (b) [26]

## 4. RADIJALNA NEURONSKA MREŽA

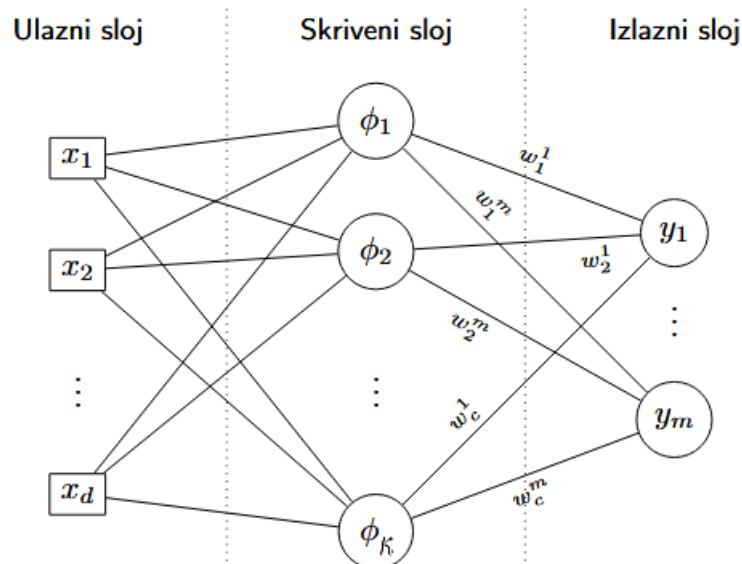
Već polovinom devedesetih godina prošlog stoljeća javlja se ideja simuliranja rada ljudskog mozga i njegovih neurona pomoću računala. Već 1948. godine McCulloch i Pitts objavljuju rad u kojem objašnjavaju na koji način ljudski mozak kroz niz povezanih neurona može rješavati kompleksne probleme [27]. Svaki biološki neuron prima i prosljeđuje signal drugom neuronu, a to radi isključivo ako je ispunjen određeni uvjet (engl. *threshold*). Na temelju činjenice da svaki neuron ima sposobnost odlučivanja, razvila se teorija umjetnih neuronskih mreža. Umjetna neuronska mreža dobila je naziv zbog simulacije bioloških neurona i njihovog rada. Nešto kasnije došlo je do prve implementacije perceptrona (naziv za funkciju ili algoritam koji oponaša ponašanje biološkog neurona u umjetnoj neuronskoj mreži) kojeg razvija Frank Rosenblatt 1958. godine [28]. Implementacija perceptrona bila je prva računalna implementacija koja je imala sposobnost učenja iz podataka. Perceptron se nalazi unutar jednog čvora neuronske mreže, ali on zapravo nije ništa više od matematičke funkcije koja kao izlaz daje binarnu vrijednost (u naprednijim mrežama izlaz može biti i vrijednost unutar određenog intervala, a ne nužno cjelobrojna binarna vrijednost). Prilagođavanjem parametara perceptrona mijenjaju se njegove izlazne vrijednosti pa s time i rad cjelokupne mreže (što je i konačni cilj). Definicija jednostavnog perceptrona dana je formulom (4.1) gdje  $n$  predstavlja broj ulaza u čvor,  $w_i \in \mathbb{R}$  predstavlja težinu veze koja ulazi u čvor,  $x_i \in \mathbb{R}$  predstavlja ulaznu vrijednost značajke, a  $b \in \mathbb{R}$  predstavlja prag (engl. *bias*) koji je potrebno zadovoljiti da bi došlo do aktivacije čvora.

$$f(x) = \begin{cases} 1, & \sum_{i=1}^n w_i x_i + b > 0 \\ 0, & \text{inače} \end{cases} \quad (4.1)$$

Radijalna neuronska mreža vrsta je umjetne neuronske mreže koju prvi put spominju Broomhead i Lowe prilikom rada na projektu uspostavljanja radarskih sustava u Ujedinjenom Kraljevstvu [29]. U njihovom radu radijalna neuronska mreža koristi se za interpolaciju varijabli u višedimenzionalnim prostorima, no ovakve vrste mreža koriste se i za probleme klasifikacije, kontrole sustava, predviđanje i aproksimaciju. Glavna razlika između radijalne neuronske mreže i ostalih neuronskih mreža jest da čvorovi kao aktivacijsku funkciju imaju radijalnu funkciju o kojoj se više govori u sljedećem poglavlju.

### 4.1. Struktura radijalne neuronske mreže

Radijalna neuronska mreža sastoji se od tri dijela, a to su ulazni sloj, jedan skriveni sloj i izlazni sloj [30]. Za razliku od ostalih neuronskih mreža, radijalne neuronske mreže mogu imati samo jedan skriveni sloj. Ulazni sloj radijalne neuronske mreže sastoji se od onoliko čvorova kolika je dimenzionalnost (broj značajki) svakog primjerka u skupu podataka. Izlazni sloj sadrži onoliko čvorova kolika je klasa ulaznog skupa. Za razliku od ulaznog i izlaznog sloja, broj čvorova u skrivenom sloju ne može se unaprijed odrediti već se njegova veličina traži



Slika 4.1: Struktura radijalne neuronske mreže [31]

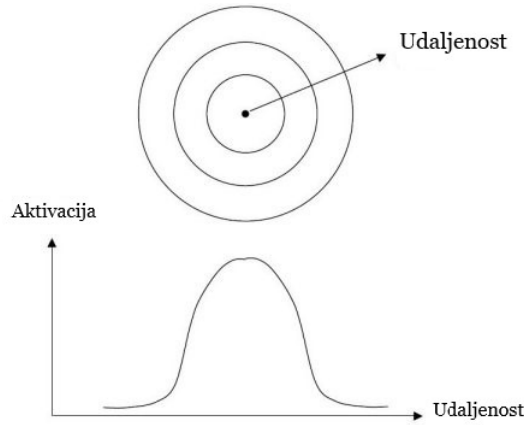
izgradnjom klasifikacijskog modela. Svi podaci se prenose od ulaznog sloja preko skrivenog do izlaznog sloja te takvu neuronsku mrežu nazivamo mreža s propagacijom prema unaprijed (engl. *feed forward neural network*). Kako bi struktura radijalne neuronske mreže bila potpuno definirana prvo se moraju pronaći čvorovi skrivenog sloja, a to zapravo znači da mora biti određen njihov broj te svi parametri radijalne funkcije koju sadrži svaki od tih čvorova. Osim čvorova potrebno je pronaći i težine veza između skrivenog i izlaznog sloja neuronske mreže. Konačna struktura jedne takve mreže prikazana je na slici 4.1. Za radijalnu funkciju najčešće se uzima Gaussova funkcija  $\phi(\mathbf{x})$  koja je definirana prema (4.2) [31].

$$\phi(\mathbf{x}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\cdot\sigma^2}} \quad (4.2)$$

Za definiciju Gaussove funkcije potrebno je odrediti dva parametra:

1. centar funkcije (*oznaka* :  $\mathbf{z} \in \mathbb{R}^N$ )
2. širina funkcije (*oznaka* :  $\sigma \in \mathbb{R}, \sigma > 0$ )

Osim parametara  $\mathbf{z}$  i  $\sigma$  u brojniku eksponenta Gaussove funkcije nalazi se norma označena s  $\|\cdot\|$ , gdje se najčešće za normu uzima Euklidska norma. Radijalna funkcija daje različite vrijednosti u ovisnosti o udaljenosti ulazne točke (ulaznog primjerka) od centra neurona. Za sve ulazne primjerke koji se nalaze na istoj udaljenosti od centra radijalne funkcije ona daje istu izlaznu vrijednost, a vrijednosti izlaza radijalne funkcije opadaju s udaljavanjem ulaznog primjerka od centra funkcije [31]. Grafički prikaz može se vidjeti na slici 4.2.



**Slika 4.2:** Izlaz aktivacijske funkcije u ovisnosti o udaljenosti ulazne točke od centra neurona [32]

Radijalna funkcija ima drugačije rezultate u ovisnosti o spomenutima parametrima, a zbog toga je i izlaz neuronske mreže drugačiji. Cilj treniranja ovakve neuronske mreže jest postaviti parametre  $\mathbf{z}$  i  $\sigma$  svih radijalnih funkcija čvorova skrivenog sloja tako da neuronska mreža postigne najbolje moguće rezultate pri klasifikaciji nepoznatih elemenata. Postoje različiti algoritmi kojima se pronalaze centri i širine radijalnih funkcija, a neki od njih koji su korišteni u ovom radu opisani su u idućim potpoglavljima. Osim centara i širina postoje parametri koji također mogu utjecati na izlaz radijalne neuronske mreže, a to su težine veza između skrivenog sloja i izlaznog sloja [32]. Težine veza su realne vrijednosti koje se postavljaju prilikom treniranja algoritma. Težine se mogu opisati kao vrijednosti koje govore koliko je važan izlaz svakog od čvorova skrivenog sloja mreže. Tako je moguće odrediti koji čvor više, a koji manje utječe na konačni izlaz mreže. Nakon definiranja svih parametara koji mogu utjecati na izlaz radijalne neuronske mreže može se napisati izraz (4.3), gdje parametar  $w_i^j \in \mathbb{R}, i = 1, \dots, \kappa, j = 1, \dots, m$  predstavlja težinu veze (engl. *weight*) između dva čvora mreže,  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}$  predstavlja ulazni vektor značajki, a  $\kappa$  predstavlja broj čvorova skrivenog sloja. Iz izraza se može vidjeti da je svaki pojedinačni izlaz neuronske mreže određen s linearnom kombinacijom izlaza radijalnih funkcija i težina veza između čvorova skrivenog i izlaznog sloja.

$$y_j = \sum_{i=1}^{\kappa} w_i^j \cdot \phi_i(\mathbf{x}) \quad (4.3)$$

Na izlazima neuronske mreže mogu se pojaviti vrijednosti u intervalu  $[0,1]$ . Nakon što svaki izlaz poprimi neku vrijednost iz tog intervala svi izlazi se obrađuju tako da izlaz s najvećom vrijednosti poprimi vrijednost 1, a ostali izlazi vrijednost 0. Samo jedan izlazni čvor može poprimiti vrijednost 1 što znači da je neuronska mreža ulaznom primjerku pridružila oznaku klase koja se nalazi na tom čvoru. Drugačiji izraz izlaza radijalne neuronske mreže može se definirati prema (4.4) gdje  $\mathbf{x}^i \in X, i = 1, \dots, N$  predstavlja ulazni vektor značajki,  $l_i \in L, i = 1, \dots, m$  predstavlja oznaku klase koja uistinu pripada ulaznom vektoru značajki, a  $l_o \in L, o = 1, \dots, m$  predstavlja oznaku klase koju je klasifikacijski algoritam pridružio ulaznom vektoru ( $m$

predstavlja broj klasa u koje je moguće klasificirati ulazni primjerak).

$$f((x^i, l_i), l_o) = \begin{cases} 1, & l_i = l_o \\ 0, & \text{inače} \end{cases} \quad (4.4)$$

## 4.2. Trening i izgradnja klasifikacijskog modela

Jedan od već spomenutih parametara radijalne neuronske mreže je broj čvorova u skrivenom sloju [33]. Zadatak izgradnje modela RBFN je pronaći optimalni broj čvorova skrivenog sloja kako bi klasifikacijski model dao dovoljno dobre rezultate pri klasifikaciji. Postupak pronalaženja optimalnog broja čvorova u skrivenom sloju nije jednostavan te je zbog toga predloženo mnogo različitih postupaka za tu namjenu. Kao što je objašnjeno u prethodnom poglavlju, u svaki čvor skrivenog sloja ugrađena je radijalna funkcija čiji izlaz ovisi o dvama parametrima (širine i centri). Pronalaženje širina i centara prvi je dio treninga neuronske mreže, dok je drugi dio pronalaženje vrijednosti težina veza između čvorova skrivenog i izlaznog sloja. Kao što je već rečeno u prethodnom poglavlju, izlaz radijalne neuronske mreže može se odrediti kao linearna kombinacija težina i izlaza radijalnih funkcija unutar čvorova skrivenog sloja. Zbog toga se (4.3) može zapisati u matricnom obliku gdje  $\mathbf{o} = (o_1, o_2, \dots, o_N) \in O$  predstavlja vektor željenih izlaza,  $\mathbf{x}^N = (x_1^N, x_2^N, \dots, x_d^N) \in X$  predstavlja vektor ulaznih primjeraka (gdje  $N$  predstavlja broj ulaznih primjeraka), a  $\Phi_{N \times \kappa}$  predstavlja matricu popunjenu izlazom radijalne funkcije definirane pomoću (4.2). Matricni zapis izlaza radijalne neuronske mreže može se definirati pomoću (4.5) [31].

$$\mathbf{o} = \Phi \cdot \mathbf{x}^N \quad (4.5)$$

Kako bi mreža imala što bolje performanse, cilj treniranja mreže je pronaći optimalne vrijednosti težina. Težine je moguće pronaći rješavanjem problema najmanjih kvadrata. Rješavanjem problema najmanjih kvadrata dobivamo takve vrijednosti težina za koje je zbroj kvadrata grešaka (engl. *sum of squared errors*, SSE) između očekivanih oznaka ulaznih primjeraka i predviđenih oznaka ulaznih primjeraka minimalan. SSE se može prikazati pomoću (4.6) gdje  $y_i$  predstavlja očekivani izlaz za ulazni primjerak, a  $\hat{y}_i$  predviđeni izlaz za isti ulazni primjerak [31].

$$SSE = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.6)$$

Osim SSE često se koristi i srednja kvadratna greška (engl. *mean squared error*, MSE) pomoću koje se iskazuje točnost konačnog klasifikacijskog modela. Ova mjera predstavlja prosječnu kvadratnu razliku (grešku) između predviđenih i ulaznih oznaka promatranih klasa. MSE se može definirati prema (4.7) gdje  $y_i$  i  $\hat{y}_i$  imaju isto značenje kao i kod SSE [31]. Stoga obje navedene mjere prikazuju ukupnu grešku predviđanja radijalne neuronske mreže i zbog toga su

često korištene pri iskazivanju kvalitete konačnog klasifikacijskog modela.

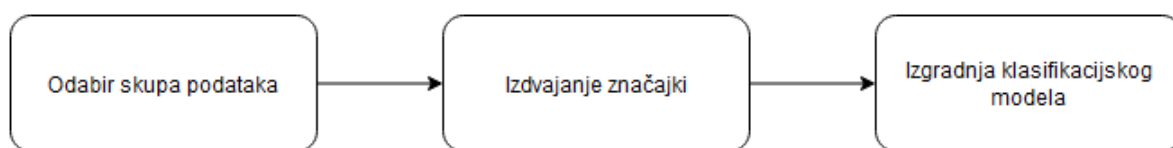
$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.7)$$

## 5. OPIS PROGRAMSKOG RJEŠENJA

Kako bi se izgradio klasifikacijski model potrebno je provesti određeni niz postupaka. Svi postupci opisani su u prethodnim poglavljima, a u nastavku je dan uvid u implementaciju svakog postupka koji objedinjeni čine programsko rješenje za izgradnju klasifikacijskog modela RBFN za raspoznavanje emocija iz izraza lica. Također, uz kratke opise svakog postupka bit će priloženi i dijelovi programskog koda.

### 5.1. Tijek postupaka

Prvi postupak jest odabir skupa podataka koji opisuje problem. Skup podataka koji se koristi u ovom radu zove se FER2013 i on sadrži crno-bijele slike lica ljudi koji osjećaju različite emocije. S obzirom na to da ulazni skup podataka sadrži isključivo slike, klasifikacijski algoritmi ne mogu ga kao takvog koristiti. Kako bi se ulazni podaci prilagodili algoritmima s kojima se radi, provodi se postupak izdvajanja značajki. U ovom radu postupak izdvajanja značajki proveo se pomoću tri različita algoritma. Nakon što je formiran skup podataka čije podatke klasifikacijski algoritmi mogu obraditi, provodi se postupak treniranja algoritma radijalne neuronske mreže kako bi se pronašli njegovi optimalni parametri tj. kako bi se izgradio klasifikacijski model. Na slici 5.1 može se vidjeti tijek provedenih postupaka.



**Slika 5.1:** *Tijek postupaka pri izgradnji klasifikacijskog modela radijalne neuronske mreže*

### 5.2. Algoritmi za izdvajanje značajki

U nastavku su priložene implementacije (programski kod) postupaka za izdvajanje značajki. Svi postupci implementirani su u Python programskom jeziku pomoću biblioteke *scikit-learn* [34] iz koje su korištene implementacije algoritama za izdvajanje značajki. Svi korišteni algoritmi opisani su u potpoglavlju 3.2. Prvi algoritam je histogram orijentiranih gradijenata, a implementacija funkcije koja kreira skup podataka pomoću ovog algoritma može se vidjeti na slici 5.2. Funkcija otvara direktorij sa slikama te pomoću `hog()` funkcije iz *scikit-learn* biblioteke iz svake slike izdvaja značajke primjenom HOG algoritma. Povratna vrijednost te funkcije jest matrica decimalnih vrijednosti u intervalu  $[0,1]$  gdje svaki redak predstavlja jedan primjerak ulaznog skupa podataka (vektor značajki). S obzirom na to da svaki ulazni primjerak u ulaznom skupu podataka mora imati pridruženu oznaku klase, na slici 5.4 nalazi se programski kod kojim se kreira takav skup podataka. Na slici 5.3 prikazan je kod kojim se kreira vektor oznaka koje se pridružuju ulaznim primjercima. Ako je potrebno, moguće je napraviti i

normalizaciju podataka (slika 5.5). Parametri koji su bitni za djelovanje algoritma HOG su broj intervala histograma (engl. *orientations*), veličina bloka korištenog za generiranje histograma (engl. *pixels-per-cell*) i veličina bloka korištenog pri normalizaciji (engl. *cells-per-block*). Izmjenom ovih parametara može se utjecati na performanse algoritma HOG.

```
1 def getHOGFeatures():
2     images_path = "D:/Diplomski/FER2013/Dataset/"
3     HOG_features = []
4     for dir in os.listdir(images_path):
5         if os.path.isdir(images_path + dir):
6             for item in os.listdir(images_path + dir):
7                 image = Image.open(images_path + dir + '/' + item)
8                 image_resized = image.resize((64,128), resample=Image.LANCZOS)
9                 fd, _ = hog(image_resized,
10                            orientations=9,
11                            pixels_per_cell=(8,8),
12                            cells_per_block=(2,2),
13                            visualize=True)
14                 HOG_features.append(fd)
15
16     return HOG_features
```

Slika 5.2: Izdvajanje značajki iz slike pomoću algoritma HOG

```
1 def createLabels():
2     labels_array = []
3     label_classes = range(0,7) # postoji 7 klasa, dakle 7 oznaka
4     for i in range(len(label_classes)):
5         for _ in range(100):
6             labels_array.append(label_classes[i])
7
8     return labels_array # vektor oznaka
```

Slika 5.3: Izrada vektora klasnih oznaka



```

1 if __name__ == '__main__':
2     HOG_features = getHOGFeatures()
3     labels = createLabels()
4     for i in range(len(HOG_features)): # pridruzi oznake ulaznim primjercima
5         HOG_features[i] = np.append(HOG_features[i], labels[i])
6     for i in range(700): # 700 primjeraka
7         for j in range(3361): # dimenzionalnost ulaznog skupa podataka
8             HOG_features[i][j] = np.array(HOG_features[i][j]).round(7)

```

**Slika 5.4:** *Izrada ulaznog skupa podataka*

```

1 HOG_features_dataframe = pd.DataFrame(HOG_features)
2 for column in HOG_features_dataframe.columns:
3     A = HOG_features_dataframe[column]
4     minA = HOG_features_dataframe[column].min()
5     maxA = HOG_features_dataframe[column].max()
6     HOG_features_dataframe[column] =
7         (A - minA) / (maxA - minA) # min_max normalizacija
8 HOG_features_normalized = HOG_features_dataframe.to_numpy()

```

**Slika 5.5:** *Normalizacija podataka*

Sljedeći korišteni algoritam je LBP. Treba napomenuti da su postupci izrade vektora klasnih oznaka, normalizacije te izrade konačnog skupa podataka isti za sve algoritme. Jedina razlika između svih implementacija je algoritam koji se koristi za izdvajanje značajki. Zbog toga su u nastavku dane samo funkcije izdvajanja značajki, a sve ostale funkcije mogu se vidjeti na slikama iznad. Pri izdvajanju značajki pomoću algoritma LBP korišteni su parametri  $(P,R) = (24,3)$  što se može vidjeti na slici 5.6.

```

1 def getLBPFeatures():
2     radius = 3
3     points_num = radius * 8
4     hist_n_bins = points_num + 2
5     images_path = "D:/Diplomski/FER2013/Dataset/"
6     data = []
7     for dir in os.listdir(images_path):
8         if os.path.isdir(images_path + dir):
9             for item in os.listdir(images_path + dir):
10                image = Image.open(images_path + dir + '/' + item)
11                lbp = feature.local_binary_pattern(image, points_num,
12                                                    radius, method="uniform")
13                print(lbp);
14                hist, _ = np.histogram(lbp.ravel(),
15                                      bins=np.arange(0, hist_n_bins),
16                                      range=(0, hist_n_bins))
17                hist = hist.astype("float")
18                data.append(hist)
19     return data

```

**Slika 5.6:** Izdvajanje značajki pomoću algoritma LBP

```

1 def getLDAFeatures(feature_matrix, target_array):
2     lda = LinearDiscriminantAnalysis(n_components=6)
3     LDA_features = lda.fit(feature_matrix, target_array)
4                     .transform(feature_matrix)
5     return LDA_features

```

**Slika 5.7:** Izdvajanje značajki pomoću algoritma LDA

Na kraju treba spomenuti i sve ostale pomoćne biblioteke koje su korištene pri implementaciji, a one se mogu vidjeti na slici 5.8.

```

1 import os
2 import numpy as np
3 from PIL import Image
4 import pandas as pd
5 from skimage.feature import hog, local_binary_pattern
6 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

```

Slika 5.8: Prikaz korištenih biblioteka

Ovi algoritmi izabrani su zbog čestog navođenja u literaturi te se njihovi rezultati uspoređuju u kasnijim poglavljima. Nakon usporedbe može se zaključiti koji je od njih najprikladniji za treniranje klasifikacijskog algoritma radijalne neuronske mreže i koliko je izbor algoritma za izdvajanje značajki utjecajan na konačnu kvalitetu izgrađenog klasifikacijskog modela.

### 5.3. Postupci izgradnje RBFN

U prethodnim poglavljima objašnjen je zadatak izgradnje radijalne neuronske mreže, a ovaj zadatak moguće je riješiti pomoću raznih postupaka koji su dani u literaturi. Na temelju rezultata dobivenih provedbom ovog postupka može se reći je li određeni klasifikacijski algoritam povoljan za rješavanje problema. U ovom radu vrednuju se tri automatska postupka za izgradnju klasifikacijskog modela radijalne neuronske mreže gdje je cilj pronaći onaj najprikladniji.

#### 5.3.1. Postupak nasumične izgradnje

Prvi postupak za izgradnju modela korišten u ovome radu je postupak nasumične izgradnje koji za broj čvorova skrivenog sloja mreže bira nasumičnu vrijednost unutar određenog intervala. Interval iz kojeg se bira nasumična vrijednost je  $[2, \sqrt{N}]$ , gdje je  $N$  broj primjeraka u ulaznom skupu podataka. Nakon što se generira nasumična vrijednost  $\kappa \in [2, \sqrt{N}]$ , iz ulaznog skupa podataka nasumično se kopira  $\kappa$  primjeraka koji predstavljaju centre čvorova skrivenog sloja mreže. S obzirom na to da su centri predstavljeni kao vektori (primjerci iz ulaznog skupa podataka jesu vektori), nakon odabiranja dobiva se matrica  $M_{\kappa \times d}$ , gdje je  $d$  dimenzionalnost primjeraka. Sljedeći korak je definiranje širina čvorova mreže. Širine svih čvorova u skrivenom sloju računaju se prema (5.1), gdje je  $d_{max}$  najveća moguća udaljenost između bilo koja dva centra mreže [35].

$$\sigma = \frac{d_{max}}{\sqrt{2\kappa}} \quad (5.1)$$

Na kraju potrebno je još pronaći težine veza između skrivenog i izlaznog sloja, a njih dobivamo rješavanjem problema najmanjih kvadrata. Pseudokod postupka nasumične izgradnje prikazan je algoritmom 1.

---

**Algoritam 1:** Postupak nasumične izgradnje

---

```
1 Postavi broj primjeraka  $N$ ;  
2 Postavi dimenzionalnost primjeraka  $d$  ;  
3 Postavi broj centara  $\kappa$  na nasumičnu vrijednost iz intervala  $[2, \sqrt{N}]$  ;  
4 Postavi ulazni skup podataka  $X[N, d]$ ;  
5 Nasumično odaberi centre  $C[1 \dots \kappa]$  iz  $X$ ;  
6 za  $i := 1, \dots, \kappa$  čini  
7   |    $r :=$  nasumična vrijednost iz intervala  $[0, N - 1]$ ;  
8   |   za  $j := 1, \dots, d$  čini  
9   |   |    $C[i] := X[r][j]$ ;  
10  |   kraj  
11 kraj  
12 definiraj širine  $W[1 \dots \kappa]$  pomoću algoritma za pronalaženje širina (algoritam 2).
```

---

**Algoritam 2:** Algoritam pronalaženja širina klasifikacijskog modela

---

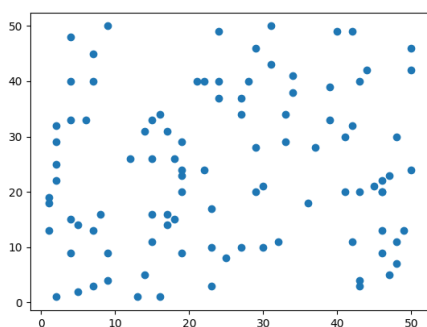
```
1  $W[\kappa]$  ; // Definicija širina  
2  $d_{max} := 0$ ;  
3 za  $i:=1, \dots, \kappa$  čini  
4   |   za  $j:=1, \dots, \kappa$  čini  
5   |   |   ako  $i \neq j$  onda  
6   |   |   |    $dist =$  izračunaj udaljenost između  $C[i]$  i  $C[j]$ ;  
7   |   |   |   ako  $dist > d_{max}$  onda  
8   |   |   |   |    $d_{max} := dist$   
9   |   |   |   kraj  
10  |   |   kraj  
11  |   kraj  
12 kraj  
13 za  $i:=1, \dots, \kappa$  čini  
14  |    $W[i] := \frac{d_{max}}{\sqrt{2 * \kappa}}$ ;  
15 kraj  
16 pronađi  $W$ ;
```

---

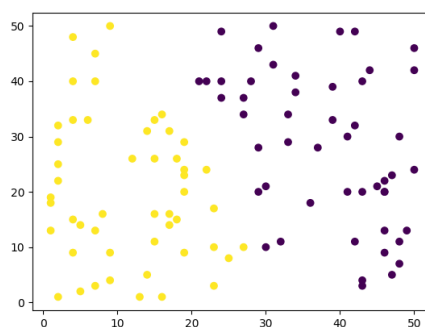
### 5.3.2. Izgradnja pomoću algoritma K-means

Algoritam K-means je algoritam za grupiranje podataka koji je zbog svoje učinkovitosti i jednostavnosti široko korišten u području strojnog učenja i rudarenju podataka. Ovaj algoritam grupira početni skup podataka u K grupa, gdje je broj K potrebno unaprijed odrediti, što nije lagan zadatak. Naime, s obzirom na broj K mijenjaju se i performanse algoritma, a s time i kvaliteta grupiranja podataka. Kako bi mogli zaključiti koji broj grupa je prikladan za određeni skup podataka moramo višestruko izvesti K-means algoritam s različitim brojem grupa. Na slici 5.9 može se vidjeti rad algoritma K-means za različit odabir broja grupa K. Ovaj algoritam iskorišten je za definiranje centara čvorova skrivenog sloja, pri čemu se broj grupa (centara) nasumično određuje. Nakon grupiranja podataka određuju se centri tih grupa te se oni sma-

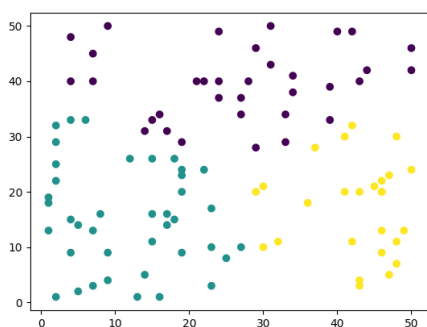
traju centrima radijalnih funkcija u čvorovima skrivenog sloja mreže. Širine čvorova i težine veza određuju se na isti način kao i kod postupka nasumične izgradnje. Pseudokod izgradnje modela pomoću algoritma K-means prikazan je algoritmom 3.



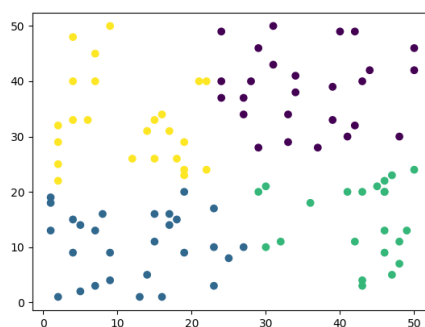
(a) Izgled podataka prije grupiranja



(b) Izgled podataka nakon grupiranja ( $K=2$ )



(c) Izgled podataka nakon grupiranja ( $K=3$ )



(d) Izgled podataka nakon grupiranja ( $K=4$ )

**Slika 5.9:** Prikaz grupiranja podataka K-means algoritmom za različit broj grupa  $K$

---

**Algoritam 3:** Izgradnja pomoću algoritma K-means

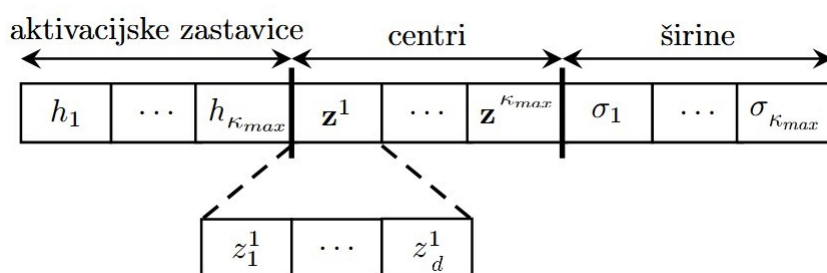
---

- 1 Postavi broj primjeraka  $N$ ;
  - 2 Postavi dimenzionalnost primjeraka  $d$ ;
  - 3 Postavi broj centara  $\kappa$  na nasumičnu vrijednost iz intervala  $[2, \sqrt{N}]$ ;
  - 4 Postavi ulazni skup podataka  $X[N, d]$ ;
  - 5 Nasumično odaberi centre  $C[1 \dots \kappa]$  iz  $X$ ;
  - 6 Pronađi  $\kappa$  centara pomoću algoritma K-means, gdje je  $K = \kappa$ ;
  - 7 Pronađi širine  $W[1 \dots \kappa]$  pomoću algoritma za pronalaženje širina (algoritam 2).
- 

### 5.3.3. Izgradnja pomoću algoritma optimizacije rojem čestica

Algoritam optimizacije rojem čestica (engl. *particle swarm optimization*, PSO) je algoritam koji su 1995. osmislili James Kennedy and Russell Eberhart [36]. Ovaj algoritam pokušava simulirati ponašanje određenih populacija životinja, kao što su rojevi pčela, jato ptica ili čopor vukova. Naime, jasno je da se populacija sastoji od više jedinki i da populacija od svake jedinke ima određenu korist. Pri letu na daleke udaljenosti ptice se formiraju u jato zbog toga što

svaka individualna jedinka može dati svoj doprinos cijelom jatu te tako olakšati let. Najjača ptica predvodi jato i svojim krilima olakšava let drugim pticama. Kada se ptica na čelu jata umori, ona se mijenja s drugom pticom iz jata koja može podnijeti let na čelu jata. Na ovaj način jato u svakom trenutku leta pokušava postići najbolji mogući raspored ptica kako bi let bio što lakši. Takvo ponašanje pokušava se simulirati algoritmom PSO gdje svaka čestica (engl. *particle*) unutar roja čestica (engl. *swarm*) doprinosi pri pronalaženju kakvog optimalnog rješenja problema. Na početku izvođenja algoritma zadaje se određeni broj početnih pozicija čestica. U slučaju izgradnje klasifikacijskog modela može se reći da su to početne pozicije centara i početne vrijednosti širina (ove vrijednosti mogu biti nasumično odabrane). Postupak pronalaženja novih centara i širina ponavlja se određeni broj iteracija. Rješenja u algoritmu PSO kodiraju se kao vektor duljine  $\kappa_{max} \cdot d + \kappa_{max} \cdot 2$ ,  $\kappa_{max} = \sqrt{N}$ , gdje  $\kappa_{max}$  predstavlja maksimalan broj čvorova skrivenog sloja, dok  $d$  predstavlja dimenzionalnost vektora značajki. Vektor se može podijeliti u tri dijela. Prvi dio predstavljaju aktivacijske zastavice  $h_1, \dots, h_{\kappa_{max}}$  gdje  $h_j \in [0, 1], j = 1, \dots, \kappa_{max}$ , drugi dio centre  $\mathbf{z}^1, \dots, \mathbf{z}^{\kappa_{max}}$ , a treći dio širine  $\sigma_1, \dots, \sigma_{\kappa_{max}}$  [37]. Pomoću aktivacijskih zastavica moguće je uključiti ili isključiti određeni čvor i širinu iz rješenja tj. za svaki  $i \in [1, \kappa_{max}]$  ako vrijedi  $h_i > 0.5$ , tada centar  $\mathbf{z}^i$  i širina  $\sigma_i$  neće biti uključene u rješenje. S obzirom na to da je težine modela moguće pronaći rješavanjem problema najmanjih kvadrata (za što su potrebni samo centri i širine) ne mora ih se nužno kodirati u rješenja algoritma PSO. Prikaz opisanog vektora možemo vidjeti na slici 5.10. Nakon što su postavljeni prvi centri i širine, moguće je formirati model RBFN te izračunati vrijednosti funkcije dobrote (engl. *fitness function*). Funkcija dobrote govori o kvaliteti rješenja, a u ovom radu funkcija dobrote predstavljena je pomoću mjere MSE i člana kazne, kako je prikazano formulom (5.2) gdje je dimenzionalnost vektora značajki označena sa  $d$ , duljina vektora izlaznih čvorova sa  $m$ , duljina vektora širina jednaka je broju čvorova skrivenog sloja uključenom u svakom pojedinom rješenju  $\kappa_{pos}$ , a  $\lambda \in [0, +\infty]$  težina koja skalira  $\kappa_{pos}$ .



**Slika 5.10:** Predstavljanje rješenja klasifikacijskog modela izgrađenog pomoću PSO algoritma bez prikaza težina [37]

$$\frac{MSE}{d \cdot m} + \lambda \cdot \kappa_{pos} \quad (5.2)$$

Član kazne je brojčana vrijednost koja se pridodaje mjeri MSE kako ne bi došlo do prevelike složenosti izgrađene mreže. Kad bi funkcija dobrote bila predstavljena isključivo mjerom MSE, najkvalitetnije bi bile mreže s najvećim brojem čvorova u skrivenom sloju. Međutim, ne mora

se dogoditi da takve mreže imaju visoku sposobnost generalizacije nad neviđenim podacima. Zbog toga se uvodi dodatni član kazne koji sprječava da funkcija dobrote vraća isključivo mreže s velikim brojem čvorova. Član kazne najčešće je proporcionalan broju čvorova skrivenog sloja mreže  $\kappa$  te se može izraziti kao  $\kappa \cdot \lambda$ , gdje  $\lambda$  iznosi  $\frac{0.01}{2 \cdot m}$  [37]. Prilikom svake iteracije čestice se pomiču u nekom smjeru na promatranom prostoru i izračunava se funkcija dobrote s obzirom na nove položaje čestica (nove centre i širine). Čestice se pomiču s obzirom na izračunate vrijednosti za vlastitu poziciju te s obzirom na najbolju poziciju pronađenu za bilo koju od čestica. Svaka čestica pamti svoju najbolju pronađenu poziciju ( $pos_{best}$ ), a najbolja pozicija u roju ( $glo_{best}$ ) poznata je svim česticama nakon svake iteracije. Na ovaj način čestice međusobno mogu "komunicirati" i zajedno sudjelovati u pronalaženju najboljeg mogućeg rješenja. Promjena brzine (engl. *velocity*) svake čestice može se izraziti prema (5.3), gdje konstanta  $w$  predstavlja inerciju čestice, konstante  $c_1$  i  $c_2$  predstavljaju kognitivnu i socijalnu težinu (još se mogu nazvati i akceleracijskim koeficijentima), a  $r_1$  i  $r_2$  predstavljaju nasumične brojeve u intervalu  $[0,1]$  [38]. Odabirom različitih vrijednosti konstanti možemo utjecati na brzinu kretanja čestica pri traženju rješenja. Osim promjene brzine, čestice prilikom svake iteracije mijenjaju i smjer (engl. *position*), što se može izraziti prema (5.4) [36]. Težine modela računaju se uobičajenim postupkom, odnosno metodom najmanjih kvadrata.

$$vel[i][j] = w \cdot vel[i][j] + c1 \cdot r1 \cdot (pos_{best}[i][j] - pos[i][j]) + c2 \cdot r2 \cdot (glo_{best}[j] - pos[i][j]) \quad (5.3)$$

$$pos[i][j] = pos[i][j] + vel[i][j] \quad (5.4)$$

U ovome radu, nakon promjene pozicija svih čestica, za svaku česticu izgrađuje se klasifikacijski model na temelju njihovih novih pozicija. Već je spomenuto kako svaka čestica pamti svoju najbolju poziciju, pa tako i ovdje svaka čestica pamti onu poziciju u kojoj je ostvarila najveću kvalitetu. Pseudokod izgradnje pomoću algoritma PSO prikazan je algoritmom 4.

---

**Algoritam 4:** Izgradnja pomoću algoritma PSO

---

```
1 Postavi broj primjeraka  $N$ ;  
2 Postavi dimenzionalnost primjeraka  $d$  ;  
3 Postavi ulazni skup  $X[N][d]$ ;  
4 Postavi inerciju  $w$  i težine  $c_1$  i  $c_2$ ;  
5 Postavi broj čestica  $N_s$ ;  
6 Postavi broj iteracija  $iter$ ;  
7 Postavi veličinu vektora kodiranih rješenja  $v_{size} = \kappa_{max} \cdot (d + 2)$ ;  
8 Postavi brzine  $Vel[N_s][v_{size}]$  i pozicije  $Pos[N_s][v_{size}]$ ;  
9 Inicijaliziraj  $P_{best}[N_s][v_{size}]$  i  $G_{best}[v_{size}]$  ;  
10 dok  $iter > 0$  čini  
11   za  $i:=1, \dots, N_s$  čini  
12     izračunaj  $fit(Pos[i])$  ; // funkcija dobrote  
13     ako  $fit(Pos[i])$  bolji od prethodno najboljeg  $fit(Pos[i])$  za  $Pos[i]$  onda  
14        $P_{best}[i] := Pos[i]$ ;  
15     kraj  
16   kraj  
17   izaberi najbolji  $P_{best}[i]$  i spremi kao  $G_{best}$ ;  
18   za  $i:=1, \dots, N_s$  čini  
19     za  $i:=j, \dots, v_{size}$  čini  
20        $Vel[i][j] := w \cdot Vel[i][j] + c_1 \cdot r_1 \cdot (P_{best}[i][j] - Pos[i][j]) + c_2 \cdot r_2 \cdot (G_{best}[j] - Pos[i][j])$ ;  
21        $Pos[i][j] := Pos[i][j] + Vel[i][j]$ ;  
22     kraj  
23   kraj  
24    $iter = iter - 1$ ;  
25 kraj
```

---



## 6. EKSPERIMENTALNA ANALIZA

U prošlim poglavljima opisani su postupci izgradnje klasifikacijskog modela s njihovim obilježjima. Kako bi se moglo utvrditi koji od navedenih postupaka daje najbolje rezultate potrebno je provesti eksperimentalnu analizu. Rezultati se promatraju prvenstveno u smislu kvalitete te veličine izgrađenih mreža, izraženih pomoću mjera opisanih u poglavlju 2.1. Usporedbom i analizom takvih rezultata donosi se zaključak koji je od korištenih postupaka najuspješniji za rješavanje ovog problema, što je u suštini i cilj ovoga rada. Osim postupka izgradnje, na kvalitetu modela može utjecati i ulazni skup podataka koji sadrži značajke izdvojene algoritmima opisanim u poglavlju 3.2. S obzirom na to da su korištena tri algoritma za izdvajanje značajki i tri različita algoritma za treniranje, postoji devet mogućih kombinacija za koje se može napraviti eksperiment. U tablici ispod mogu se vidjeti karakteristike svakog ulaznog skupa podataka korištenog pri eksperimentu. Iz tablice 6.1 može vidjeti da je skup podataka dobiven pomoću algoritma HOG puno veći od preostala dva skupa podataka. Svi skupovi podataka imaju isti broj primjeraka i raspon vrijednosti.

Algoritam	Broj primjeraka	Dimenzionalnost	Raspon vrijednosti	Broj klasa
HOG	700	3361	[0.0 - 1.0]	7
LBP	700	26	[0.0 - 1.0]	7
LDA	700	7	[0.0 - 1.0]	7

**Tablica 6.1:** Karakteristike ulaznih skupova podataka

### 6.1. Postavke i metodologija eksperimentalne analize

Prvi korak eksperimenta jest podjela ulaznog skupa podataka na podskupove za trening i test u omjeru 75:25%. Ova podjela provedena je postupkom izdvajanja (engl. *holdout*) gdje se nasumično odabiru primjerci za trening, odnosno test. Kako bi se dobio bolji uvid u performanse određenog postupka za izgradnju klasifikacijskog modela, podjela ulaznog skupa napravljena je 10 puta. Drugim riječima, svaki postupak proveden je 10 puta na istom skupu podataka s različitom podjelom podataka za trening i test. Sljedeći korak je postavljanje parametara. Promjenom vrijednosti parametara mijenja se ponašanje algoritma, što zauzvrat može utjecati na performanse izgrađenog modela. Parametri zajednički svim algoritmima su najmanji i najveći mogući broj čvorova skrivenog sloja mreže, a oni se označavaju sa  $\kappa_{min}$  i  $\kappa_{max}$ . Algoritam K-means i PSO složeniji su od algoritma nasumične izgradnje te sadrže više dodatnih parametara. Parametri korišteni za algoritam K-means su:

1. *Iter* - ovaj parametar koristi se pri treniranju pomoću algoritma K-means, a predstavlja broj iteracija koje provede algoritam

2.  $\epsilon$  - određuje minimalnu razliku između pozicija prijašnjih centara i novo odabranih centara pri radu K-means algoritma za koju algoritam nastavlja s radom. Ako je razlika između svih prijašnjih pozicija centara i novo odabranih pozicija centara manja od  $\epsilon$  algoritam se zaustavlja.

Parametri korišteni za algoritam PSO su:

1.  $NFEs$  - broj vrednovanja
2.  $N_s$  - broj čestica u populaciji
3.  $\omega$  - inercija čestica
4.  $c_1, c_2$  - akceleracijski koeficijenti
5.  $\lambda$  - član kazne.

U tablici 6.2 prikazane su vrijednosti svih parametara za svaki od korištenih algoritama u provedenom eksperimentu. Vrijednosti parametara za algoritam K-means preuzete su iz [31], dok su za algoritam PSO preuzete iz [39].

	Nasumična izgradnja	K-means	PSO
			$NFEs = 2 \cdot 10^5$
		$Iter = 500$	$N_s = 30$
	$\kappa_{min} = 2$	$\epsilon = 0.001$	$\omega = 0.724$
Parametri	$\kappa_{max} = \sqrt{N}$	$\kappa_{min} = 2$	$c_1 = c_2 = 1.468$
		$\kappa_{max} = \sqrt{N}$	$\kappa_{min} = 2$
			$\kappa_{max} = \sqrt{N}$
			$\lambda = \frac{0.01}{2 \cdot m}$

**Tablica 6.2:** *Vrijednosti parametara postavljenih pri eksperimentu*

Nakon što su postavljeni svi parametri, trenira se klasifikacijski algoritam. Nad svakim skupom podataka postupak pronalaženja vrijednosti centara i širina neuronske mreže provodi se  $2 \cdot 10^5$  puta. Ono se u suštini svodi na vrednovanje klasifikacijskog modela izgrađenog pomoću pronađenih vrijednosti pomoću funkcije dobrote algoritma korištenog za pretragu.

## 6.2. Rezultati i diskusija

Performanse uspoređenih postupaka izgradnje u smislu veličine te kvalitete izgrađenih mreža na podskupu za trening prikazane su u tablici 6.3., a na podskupu za test u tablici 6.4. Za svaki postupak izgradnje i korišteni skup podataka prikazane su performanse najboljih pronađenih

mreža za svako pokretanje eksperimenta (ukupno 10). Za svako pokretanje, najbolji rezultati su podebljani. Uz vrijednosti mjera navedeni su i brojevi čvorova skrivenog sloja izgrađenih modela.

	Nasumična izgradnja		K-means		PSO	
	$\kappa$	MSE	$\kappa$	MSE	$\kappa$	MSE
HOG	2	0.855	14	0.836	6	<b>0.499</b>
	2	0.853	16	0.832	6	<b>0.479</b>
	2	0.851	12	0.830	4	<b>0.572</b>
	2	0.851	15	0.819	7	<b>0.427</b>
	2	0.851	11	0.815	8	<b>0.484</b>
	2	0.849	18	0.811	5	<b>0.495</b>
	2	0.848	17	0.810	7	<b>0.489</b>
	3	0.847	21	0.804	8	<b>0.453</b>
	4	0.846	16	0.801	7	<b>0.491</b>
	3	0.845	21	0.799	5	<b>0.537</b>
LBP	23	0.817	18	0.829	6	<b>0.792</b>
	24	0.816	24	0.818	7	<b>0.765</b>
	23	0.810	24	0.817	6	<b>0.779</b>
	24	0.810	24	0.816	7	<b>0.757</b>
	24	0.808	23	0.814	6	<b>0.789</b>
	24	0.808	24	0.813	5	<b>0.786</b>
	24	0.807	24	0.811	5	<b>0.781</b>
	24	0.807	24	0.809	6	<b>0.773</b>
	24	0.806	24	0.808	6	<b>0.775</b>
	24	0.802	24	0.805	7	<b>0.755</b>
LDA	21	0.194	10	0.126	9	<b>0.111</b>
	18	0.193	10	0.125	10	<b>0.116</b>
	23	0.190	10	0.125	10	<b>0.109</b>
	14	0.170	10	0.124	8	<b>0.123</b>
	12	0.156	10	<b>0.124</b>	11	0.135
	15	0.148	13	0.123	9	<b>0.119</b>
	16	0.148	14	0.123	10	<b>0.118</b>
	14	0.146	10	0.123	10	<b>0.112</b>
	15	0.139	10	<b>0.122</b>	8	0.130
	16	0.138	12	0.122	11	<b>0.110</b>

**Tablica 6.3:** Performanse uspoređenih postupaka izgradnje na podskupovima za treniranje

Iz rezultata u tablici 6.3., vidljivo je da postupak izgradnje temeljen na algoritmu PSO pronalazi najkvalitetnije mreže za svako pokretanje eksperimenta. Valja podsjetiti da sva tri postupka izgradnje imaju jednak broj vrednovanja funkcije dobrote, odnosno tijekom izgradnje i treniranja vrednuju jednak broj mreža. Ipak, evidentno je kako su ta vrednovanja bolje utrošena u algoritmu PSO jer uspijeva pronaći mreže s manjom greškom.

	Nasumična izgradnja			K-means			PSO		
	$\kappa$	MSE	F1	$\kappa$	MSE	F1	$\kappa$	MSE	F1
HOG	2	<b>0.849</b>	0.086	14	0.852	0.247	6	1.004	<b>0.285</b>
	2	<b>0.848</b>	0.102	16	0.873	0.269	6	0.969	<b>0.300</b>
	2	<b>0.853</b>	0.079	12	0.858	<b>0.234</b>	4	1.023	0.220
	2	<b>0.852</b>	0.084	15	0.871	0.205	7	1.071	<b>0.293</b>
	2	0.846	0.093	11	<b>0.828</b>	<b>0.329</b>	8	1.019	0.257
	2	<b>0.855</b>	0.064	18	0.860	0.271	5	0.977	<b>0.283</b>
	2	0.850	0.084	17	<b>0.839</b>	0.267	7	0.931	<b>0.293</b>
	3	<b>0.850</b>	0.093	21	0.856	0.249	8	1.053	<b>0.269</b>
	4	<b>0.856</b>	0.114	16	0.838	<b>0.279</b>	7	0.955	0.238
3	<b>0.853</b>	0.079	21	0.855	<b>0.325</b>	5	0.907	0.273	
LBP	23	0.902	0.159	18	<b>0.870</b>	<b>0.173</b>	6	31.325	0.142
	24	0.896	0.125	24	<b>0.886</b>	<b>0.197</b>	7	0.952	0.147
	23	0.910	0.126	24	<b>0.879</b>	0.165	6	0.900	<b>0.169</b>
	24	0.917	0.147	24	<b>0.879</b>	<b>0.161</b>	7	2.027	0.148
	24	0.914	0.15	23	<b>0.903</b>	<b>0.179</b>	6	0.936	0.145
	24	<b>0.895</b>	0.167	24	0.899	0.165	5	1.309	<b>0.239</b>
	24	0.899	0.109	24	<b>0.884</b>	<b>0.178</b>	5	10.991	0.126
	24	0.908	0.151	24	<b>0.890</b>	<b>0.183</b>	6	0.933	0.147
	24	0.907	0.138	24	<b>0.897</b>	0.182	6	0.992	<b>0.188</b>
24	<b>0.902</b>	0.142	24	0.908	0.130	7	697.970	<b>0.191</b>	
LDA	21	0.215	0.916	10	0.164	0.905	13	<b>0.148</b>	<b>0.934</b>
	18	0.230	<b>0.918</b>	10	<b>0.163</b>	0.899	10	0.168	0.892
	23	0.219	<b>0.922</b>	10	0.162	0.899	11	<b>0.162</b>	0.909
	14	0.185	0.916	10	0.172	0.895	9	<b>0.138</b>	<b>0.951</b>
	12	0.162	0.928	10	0.173	0.897	8	<b>0.102</b>	<b>0.961</b>
	15	0.167	0.912	13	0.174	0.895	9	<b>0.143</b>	<b>0.917</b>
	16	<b>0.170</b>	0.909	14	0.176	0.897	10	0.177	<b>0.917</b>
	14	0.181	0.898	10	0.169	<b>0.912</b>	10	<b>0.159</b>	0.904
	15	0.178	0.904	10	<b>0.168</b>	<b>0.917</b>	8	0.193	0.911
16	0.176	0.898	12	0.170	0.910	11	<b>0.157</b>	<b>0.939</b>	

Tablica 6.4: Performanse uspoređenih postupaka izgradnje na podskupovima za testiranje

Nadalje, postupak izgradnje zasnovan na algoritmu PSO također pronalazi i mreže s najboljom sposobnosti generalizacije, na što upućuju rezultati u tablici 6.4. Iz predočenih rezultata može se vidjeti da je najbolji rezultat (najveća vrijednost F1-mjere) postignut PSO algoritmom za skup podataka u kojemu su značajke izdvojene algoritmom LDA. Najbolji rezultati postignuti su kada broj čvorova skrivenog sloja  $\kappa$  nije bio veći od 16, što znači da mreža ne mora imati veliki broj čvorova u skrivenom sloju da bi imala povoljne performanse. Također se može vidjeti da je utjecaj korištenog postupka izgradnje puno manji nego utjecaj algoritma za izdvajanje značajki. Za skupove podataka gdje su značajke izdvojene algoritmima HOG i LBP rezultati su loši za svaki korišteni postupak izgradnje, dok kod LDA svi postupci imaju visoku razinu kvalitete. Još se može napomenuti odnos između MSE i F1-mjere gdje je iz rezultata jasno vidljivo da su one obrnuto proporcionalne. U svim slučajevima kada MSE postiže visoke rezultate, F1-mjera poprima vrlo niske vrijednosti što je i očekivano s obzirom na njihove definicije. Na temelju rezultata prikazanih u tablici 6.4 izrađena je tablica 6.5 gdje se mogu vidjeti vrijednosti koje pokazuju koliko je puta određena kombinacija algoritama izgradila najbolji model pri pokretanju eksperimenta (isto možemo postići prebrojavanjem podebljanih vrijednosti u tablici 6.4 za svaku kombinaciju algoritama). Iz tablice je vidljivo kako je algoritam nasumične izgradnje dao najbolji model samo 2 puta, dok su najbolji modeli uglavnom izgrađeni pomoću algoritama K-means i PSO. Također se mogu vidjeti prosječne vrijednosti F1-mjere za sve kombinacije algoritama, a najbolji rezultat postignut je kombinacijom algoritama PSO i LDA.

Broj najboljih rezultata i prosječna vrijednost F1-mjere						
	Nasumična izgradnja	$\bar{F}1$	K-means	$\bar{F}1$	PSO	$\bar{F}1$
HOG	0	0.088	4	0.268	6	0.271
LBP	0	0.142	6	0.171	4	0.165
LDA	2	0.912	2	0.902	6	0.924

**Tablica 6.5:** Sažete performanse uspoređenih postupaka izgradnje na podskupovima za testiranje

### 6.3. Usporedba RBFN s ostalim klasifikatorima

Problem raspoznavanja emocija iz izraza lica jedan je primjer klasifikacijskog problema. Iz rezultata prikazanih u prošlom poglavlju može se zaključiti da je moguće napraviti klasifikacijski model radijalne neuronske mreže koji uspješno rješava spomenuti problem. Bez obzira na rezultate klasifikatora radijalne neuronske mreže, još uvijek nije moguće reći je li takav klasifikator najbolji za rješavanje ovakvog problema. Kako bi se mogao dati odgovor na to pitanje, napravljen je eksperiment s više različitih klasifikatora. Svi klasifikatori trenirani su na ulaznom skupu podataka koji sadrži značajke izdvojene algoritmom LDA (na ovom skupu podataka model radijalne neuronske mreže imao je najvišu sposobnost klasifikacije). Ispitani klasifikatori jesu -

SVM, stabla odlučivanja (engl. *Decision Trees*, DT), K-NN i naivni bayesov klasifikator (engl. *Gaussian Naive Bayes*, GNB). Razlika između navedenih klasifikatora i klasifikatora radijalne neuronske mreže je u načinu izgradnje modela. Naime, opisani postupci izgradnje modela radijalne neuronske mreže jesu automatski, dok je za ostale klasifikatore potrebno definirati skup hiperparametara koji se trebaju testirati. Odabir vrijednosti hiperparametara nije jednostavan zadatak te njihove vrijednosti znatno utječu na performanse klasifikatora. Mogućnost automatskog traženja hiperparametara daje klasifikatoru radijalne neuronske mreže prednost nad ostalim klasifikatorima jer se štede vrijeme i resursi potrebni za provedbu postupka određivanja hiperparametara.

Postupak podešavanja hiperparametara (engl. *tuning*) proveden je za sve navedene klasifikatore. Cilj ovog postupka jest pronaći kombinaciju hiperparametara koja na podskupu za trening daje najbolje performanse. Postoje razni postupci kojima se mogu pronaći optimalni hiperparametri klasifikatora, a jedan od njih je mrežna pretraga (engl. *grid search*). Mrežna pretraga pronalazi optimalne hiperparametre klasifikatora iscrpnim pretraživanjem vrijednosti hiperparametara klasifikatora. Za svaku kombinaciju hiperparametara klasifikator se trenira na istom ulaznom skupu podataka te se vrednuje izgrađeni klasifikacijski model. Kombinacija hiperparametara za koju je izgrađen najuspješniji klasifikacijski model uzima se kao optimalna te se kasnije klasifikator inicijalizira s takvim hiperparametrima. U tablici 6.6 prikazane su testirane postavke hiperparametara klasifikatora. Kao primjer može se uzeti SVM klasifikator kojemu su pri ovome eksperimentu podešavana tri hiperparametra (C, Gamma i Kernel). Iz tablice je vidljivo da parametar C može poprimiti pet različitih vrijednosti, parametar Gamma sedam različitih vrijednosti i Kernel parametar četiri različite vrijednosti. S obzirom na to da je potrebno ispitati sve moguće kombinacije hiperparametara, klasifikator je treniran  $5 \cdot 7 \cdot 4$  puta.

Algoritam	Parametar	Vrijednosti
SVM	Regularizacijski parametar	[0.1, 1, 10, 100, 1000]
	Utjecaj primjerka na hiperravninu	[1, 0.1, 0.01, 0.001, 0.0001, 'scale', 'auto']
	Jezgra	['linear', 'poly', 'rbf', 'sigmoid']
DT	Mjera nečistoće	['gini', 'entropy']
	Maksimalna dubina stabla	[1,...,12], 'None'
	Maksimalan broj lisnih čvorova	[2,...,100], 'None'
	Minimalna raspodjela primjeraka	[2,...,11]
K-NN	Broj susjeda	[1,...,30]
	Težina	['uniform', 'distance']
	Veličina lisnog čvora	[1,...,60]
	Metoda računanja udaljenosti primjeraka od centra grupe	['minkowski', 'euclidean', 'manhattan']
	Snaga	[1,2]
GNB	Zaglađivanje varijance	Ravnomjerno raspoređene vrijednosti na logaritamskoj skali u intervalu [0,001, 1]

**Tablica 6.6:** Postavke hiperparametara korištene pri podešavanju klasifikatora

Tablicom 6.7 prikazani su rezultati svih klasifikatora uključujući i radijalnu neuronsku mrežu. Svaki redak tablice prikazuje rezultate (vrijednosti F1-mjere) svih klasifikatora pri jednom pokretanju eksperimenta s drugačijom podjelom ulaznog skupa podataka na trening i test podskupove. Iz tablice se može vidjeti da je pomoću algoritma GNB najviše puta izgrađen najbolji klasifikacijski model, dok pomoću algoritama SVM i DT niti jednom nije izgrađen najbolji klasifikacijski model. Zadnji redak predstavlja prosječnu vrijednost F1-mjere svakog klasifikatora. Može se uočiti da algoritam radijalne neuronske mreže ima najbolju prosječnu vrijednost F1-mjere od svih testiranih klasifikatora. To još jednom potvrđuje kako je radijalna neuronska mreža dobar izbor klasifikatora za rješavanje problema klasifikacije emocija iz izraza lica.

	Vrijednosti F1-mjere				
	RBFN	SVM	DT	KNN	GNB
1.	0.934	0.932	0.891	0.937	<b>0.943</b>
2.	0.892	0.892	0.869	0.892	<b>0.903</b>
3.	0.909	0.932	0.846	0.937	<b>0.937</b>
4.	<b>0.951</b>	0.908	0.852	0.885	0.914
5.	<b>0.961</b>	0.903	0.830	0.897	0.903
6.	0.917	0.919	0.876	0.914	<b>0.926</b>
7.	0.917	0.914	0.868	<b>0.919</b>	0.908
8.	0.904	0.933	0.825	<b>0.939</b>	0.938
9.	0.911	0.926	0.908	<b>0.938</b>	0.931
10.	<b>0.939</b>	0.914	0.868	0.909	0.903
$\bar{F1}$	<b>0.924</b>	0.917	0.863	0.917	0.921

**Tablica 6.7:** Performanse uspoređenih klasifikatora na podskupovima za testiranje



## 7. ZAKLJUČAK

Raspoznavanje emocija iz izraza lica nije jednostavan problem te postoji mnogo postupaka, metoda i algoritama za njegovo rješavanje. Ovaj problem može se predstaviti kao klasifikacijski problem u području strojnog učenja. Iako je u literaturi predloženo pregršt klasifikatora za njegovo rješavanje, pregledom literature teško je uočiti najprikladniji među njima. Ipak, radijalna neuronska mreža nameće se u brojnim istraživanjima kao jedan od pogodnih klasifikatora za ovaj problem. Međutim, na ponašanje ovog klasifikatora uvelike utječu složenost klasifikacijskog modela i postavke njegovih parametara, čije određivanje predstavlja zahtjevan i dugotrajan zadatak. U literaturi su predloženi brojni postupci i algoritmi pomoću kojih se pronalaze spomenuti parametri, a neki od njih korišteni su u ovome radu. Najboljim od razmatranih postupaka pokazao se postupak izgradnje zasnovan na optimizacijskom algoritmu PSO. Pokazano je da ovaj algoritam tijekom izgradnje i treniranja mreže uspijeva pronaći mreže najbolje kvalitete, što se kasnije odražava i na njihovu sposobnost generalizacije. Osim samih klasifikatora za pronalaženje parametara radijalne neuronske mreže, na kvalitetu klasifikacijskog modela utječe i skup podataka na kojemu se mreža trenira. Ulazni skup podataka mora sadržavati bitne informacije za rješavanje nekog problema. Ako su ulazni podaci predstavljeni pomoću slike ili videozapisa potrebno ih je obraditi prije korištenja jer klasifikator ne može direktno učiti iz slika i videozapisa. Tada se pristupa izdvajanju značajki gdje se iz ulaznog skupa podataka izdvajaju one značajke koje nose bitnu informaciju za rješavanje problema. Od takvih se izdvojenih značajki izrađuju strukture poput vektora ili matrica koje mreža može koristiti kao ulaz. Skup podataka gdje su značajke izdvojene algoritmom LDA pokazao se najboljim za ovakav problem, dok su performanse klasifikatora treniranih pomoću značajki izdvojenih algoritmima HOG i LBP bile puno lošije. Također, eksperimentalnom analizom je utvrđeno da na performanse radijalne neuronske mreže veći utjecaj ima način izvlačenja značajki od načina izgradnje klasifikacijskog modela, što potvrđuje važnost ulaznih podataka pri izradi klasifikacijskog modela.

S obzirom na to da se klasifikator radijalne neuronske mreže pokazao povoljnim za problem raspoznavanja emocije iz izraza lica opisan FER2103 skupom podataka i moguće je da se takav algoritam može koristiti i za druge probleme opisane pomoću drugih skupova podataka. Zbog toga ispitivanje kvalitete radijalne neuronske mreže na drugim problemima tj. skupovima podataka može biti smjernica za budući rad. Osim klasifikatora, ovim radom pokazalo se da ulazni skup podataka uvelike utječe na kvalitetu klasifikacijskog modela. Pri eksperimentu su se koristila tri različita skupa podataka gdje su značajke bile izdvojene pomoću tri različita algoritma. Međutim, moguće je da neki drugi algoritmi za izdvajanje značajki iz literature u kombinaciji s RBFN daju još bolje rezultate kao što su gaborov filter i metoda analize glavnih komponenti (engl. *principal component analysis*, PCA) [14, 1]. Daljnje proučavanje algoritama za izdvajanje značajki zbog toga može biti dio budućeg istraživanja.

## Literatura

- [1] V. Agarwal and S. Bhanot, “Radial basis function neural network-based face recognition using firefly algorithm.” *SpringerLink*, 2017.
- [2] F. Hernández-Luquin and H. J. Escalante, “Multi-branch deep radial basis function networks for facial emotion recognition.” *SpringerLink*, 2021.
- [3] A. S. Cowen and D. Keltner, “Self-report captures 27 distinct categories of emotion bridged by continuous gradients,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 38, pp. E7900–E7909, 2017.
- [4] P. Aleksic and A. Katsaggelos, “Automatic facial expression recognition using facial animation parameters and multistream hmms.” *IEEE Trans. Information Forensics and Security*, 2006.
- [5] A. Anil, A. Mejdal, V. Priti, K. Purushothama, A. Dimitrios, S. Prathibha, and H. Awal, “Methods for facial expression recognition with applications in challenging situations.” *National Library of Medicine*, Bethesda, Maryland, USA, 2022.
- [6] K. Vemou, A. Horvath, and T. Zerdick, “Facial emotion recognition.” *TechnoDispatch*, 2021.
- [7] “iborderctrl technologies.” Intelligent Portable Control System, <https://www.iborderctrl.eu/>. Zadnje pristupljeno: 5.7.2023.
- [8] J. K. Khan and D. Upadhyay, “Security issues in face recognition.” Amity School of Engineering and Technology (ASET), Amity University Noida, India, 2014.
- [9] P. Jiang, “Analysis on privacy issues with facial recognition technology.” School of Marxism, Shanghai Jiao Tong University, Minhang, Shanghai, China, 2014.
- [10] L. Rhue, “Racial influence on automated perceptions of emotions.” *Social Science Research Network*, University of Maryland, 2018.
- [11] K. Minhee, C. Youngwug, and S.-Y. K., “Effects of diagnostic regions on facial emotion recognition: The moving window technique.” *National Library of Medicine*, Department of Psychology, Seoul, South Korea, 2022.
- [12] Y. Khairuddin and Z. Chen, “Facial emotion recognition, state of the art performance on fer2013.” *Department of Electrical and Computer Engineering*, Boston University, Boston, MA, USA, 2021.
- [13] M. M. Htay, “Feature extraction and classification methods of facial expression.” *Faculty of Computer Science*, University of Computer Studies, Mandalay, Myanmar, 2021.

- [14] D. Mehta, M. Siddiqui, and Y. Javaid, "Recognition of emotion intensities using machine learning algorithms." *National Library of Medicine*, Basel, Switzerland, 2019.
- [15] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, "Challenges in representation learning: A report on three machine learning contests," 2013.
- [16] "Kaggle." <https://www.kaggle.com>. Zadnje pristupljeno: 1.7.2023.
- [17] A. Paudel, "Pen stroke sequence feature extraction from mnist digits." <https://abpaudel.com/blog/mnist-sequence-feature-extraction/>. Zadnje pristupljeno: 18.6.2023.
- [18] R. K. McConnell, "Method of and apparatus for pattern recognition." *Waylnad Research Inc.*, Wayland, MA, United States, 1986.
- [19] M. R. William T. Freeman, "Orientation histograms for hand gesture recognition," Tech. Rep. TR94-03, MERL - Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, Dec. 1994.
- [20] W. Lilian, "Object detection for dummies part 1: Gradient vector, hog, and ss." <https://lilianweng.github.io/posts/2017-10-29-object-recognition-part-1/>. Zadnje pristupljeno: 20.5.2023.
- [21] S. Aishwarya, "Feature engineering for images: A valuable introduction to the hog feature descriptor." <https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor>. Zadnje pristupljeno: 20.5.2023.
- [22] Satya Mallick, "Histogram of Oriented Gradients explained using OpenCV." <https://learnopencv.com/histogram-of-oriented-gradients/>. Zadnje pristupljeno: 1.7.2023.
- [23] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray scale and rotation invariant texture classification with local binary patterns." University of Oulu, Finland, 2002.
- [24] A. R. Fisher, "The use of multiple measurements in taxonomic problems," 1936.
- [25] C. Rao Radhakrishna, "The use of multiple measurements in taxonomic problems." *The Indian Journal of Statistics*, 1946.
- [26] C. Maklin, "Linear discriminant analysis in python." <https://towardsdatascience.com/linear-discriminant-analysis-in-python-76b8b17817c2>. Zadnje pristupljeno: 20.5.2023.
- [27] S. M. Warren and W. Pitts, "A logical calculus of the ideas immanent in nervous activity." University of Illinois, Department of Psychiatry at the Illinois Neuropsychiatric Institute and the University of Chicago, 1943.

- [28] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." Cornell Aeronautical Laboratory, 1958.
- [29] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks." Royal Signals and Radar Establishment, United Kingdom, 1988.
- [30] L. Ramadhan, "Radial basis function neural network simplified." <https://towardsdatascience.com/radial-basis-function-neural-network-simplified-6f26e3d5e04d>. Zadnje pristupljeno: 7.7.2023.
- [31] M. Dudjak, "Učenje iz neuravnoteženih podataka unaprijeđenim postupcima za odabir značajki, preuzorkovanje i izgradnju radijalnih neuronski mreža." Sveučilište Josipa Jurja Strossmayera, FERIT, Osijek, 2022.
- [32] Dtreg, "Rbf neural networks." <https://www.dtreg.com/solution/rbf-neural-networks>. Zadnje pristupljeno: 5.7.2023.
- [33] M. Kovacevic, N. Ivanisevic, T. Dasic, and L. Markovic, "Application of artificial neural networks for hydrological modelling in karst," *Gradjevinar*, vol. 70, 01 2018.
- [34] "Skimage documentation." <https://scikit-image.org/>. Zadnje pristupljeno: 1.7.2023.
- [35] N. Benoudjit<sup>1</sup>, C. Archambeau<sup>1</sup>, A. Lendasse, J. Lee, and M. Verleysen<sup>1</sup>, "Width optimization of the gaussian kernels in radial basis function networks." Université Catholique de Louvain - Microelectronics Laboratoy, European Symposium on Artificial Neural Networks, Bruges, Belgium, 1988.
- [36] J. Kennedy and R. Eberhart, "Particle swarm optimization." PurdueSchool of Engineering and Technology, Indianapolis, 1995.
- [37] Bajer, D. (2017). Unaprjedenja algoritma diferencijalne evolucije podešavanjem parametara i izborom počrtne populacije (Disertacija). Osijek: Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek. Preuzeto s <https://urn.nsk.hr/urn:nbn:hr:200:645925>.
- [38] H. P. Singh and A. Kaur, "Statistical analysis of velocity update rules in particle swarm optimization." Guru Nanak Dev University, Amritsar, Computer Science Department, India, 2015.
- [39] K. R. Harrison, B. M. Ombuki-Berman, and A. P. Engelbrecht, "A parameter-free particle swarm optimization algorithm using performance classifiers," *Information Sciences*, vol. 503, pp. 381–400, 2019.

## Popis slika

2.1. Prikaz pozicije dijela lica ovisno o osjećaju različitih emocija [5] . . . . .	3
2.2. Razredi problema strojnog učenja . . . . .	4
2.3. Matrični zapis označenih primjeraka $O$ . . . . .	4
2.4. Primjer matrice zabune . . . . .	5
3.1. Prikaz strukture FER-2013 direktorija prije(a) i poslije prilagodbe(b) . . . . .	11
3.2. Primjer izdvajanja značajki iz slika brojeva [17] . . . . .	12
3.3. Slika prije (a) i poslije (b) promjene dimenzije . . . . .	12
3.4. Prikaz gradijenata piksela slike(a), magnitude(b) i orijentacije(c) . . . . .	13
3.5. Kreiranje histograma [22] . . . . .	14
3.6. Usporedba originalne slike(a) i izlazne slike nakon primjene algoritma HOG(b) .	15
3.7. Kružno raspoređeni susjedi promatranog piksela za različite parametre (P,R) [23]	16
3.8. Primjeri značenja pojedinih uzoraka (binarnih vektora) [23] . . . . .	17
3.9. Prikaz svih mogućih uniformnih uzoraka (prvi red) i neuniformnih uzoraka (os- tali redovi) za $P=8$ [23] . . . . .	18
3.10. Odnos točaka na temelju dvije varijable (a) i odnos točaka zanemarivanjem jedne dimenzije (b) [26] . . . . .	19
3.11. Novo generirani pravac pomoću algoritma LDA (a) i prikaz novog rasporeda u 1D ravnini (b) [26] . . . . .	19
4.1. Struktura radijalne neuronske mreže [31] . . . . .	21
4.2. Izlaz aktivacijske funkcije u ovisnosti o udaljenosti ulazne točke od centra ne- urona [32] . . . . .	22
5.1. Tijek postupaka pri izgradnji klasifikacijskog modela radijalne neuronske mreže	25
5.2. Izdvajanje značajki iz slike pomoću algoritma HOG . . . . .	26
5.3. Izrada vektora klasnih oznaka . . . . .	26
5.4. Izrada ulaznog skupa podataka . . . . .	27
5.5. Normalizacija podataka . . . . .	27
5.6. Izdvajanje značajki pomoću algoritma LBP . . . . .	28
5.7. Izdvajanje značajki pomoću algoritma LDA . . . . .	28
5.8. Prikaz korištenih biblioteka . . . . .	29
5.9. Prikaz grupiranja podataka K-means algoritmom za različit broj grupa $K$ . . . .	31
5.10. Predstavljanje rješenja klasifikacijskog modela izgrađenog pomoću PSO algo- ritma bez prikaza težina [37] . . . . .	32

## Popis tablica

6.1. Karakteristike ulaznih skupova podataka . . . . .	35
6.2. Vrijednosti parametara postavljenih pri eksperimentu . . . . .	36
6.3. Performanse uspoređenih postupaka izgradnje na podskupovima za treniranje .	37
6.4. Performanse uspoređenih postupaka izgradnje na podskupovima za testiranje . .	38
6.5. Sažete performanse uspoređenih postupaka izgradnje na podskupovima za testi- ranje . . . . .	39
6.6. Postavke hiperparametara korištene pri podešavanju klasifikatora . . . . .	41
6.7. Performanse uspoređenih klasifikatora na podskupovima za testiranje . . . . .	42

## Sažetak

Problem klasifikacije prisutan je u mnogim granama znanosti te je zbog toga u zadnjih nekoliko desetljeća osmišljeno i predloženo mnoštvo postupaka kojima se ovakav problem može riješiti. Raspoznavanje emocija iz izraza lica jedan je od mogućih problema koji se može riješiti izradom klasifikacijskih modela. Takvi modeli imaju sposobnost pridruživanja predefiniranih oznaka ulaznim primjercima. U ovom radu kao klasifikator korištena je radijalna neuronska mreža za koju se pokazalo da može uspješno riješiti ovakav problem. Kako bi se mogao izgraditi klasifikacijski model radijalne neuronske mreže potrebno je definirati skup podataka koji dobro opisuje navedeni problem. U ovom radu korišten je FER2013 skup podataka. Eksperimentalnom analizom nad ovim skupom podataka utvrđeno je kako je moguće izgraditi klasifikacijski model radijalne neuronske mreže povoljne izvedbe pri raspoznavanju emocija iz izraza lica. Problem ovog skupa podataka je to što su podaci predstavljeni slikama, a radijalna neuronska mreža ne može raditi sa strukturom podataka kao što je slika ili videozapis. Zbog toga je potrebno provesti postupak izdvajanja značajki gdje se bitne informacije izdvajaju iz slika te se tako izrađuju strukture s kojima mreža može raditi. U ovom radu opisana su tri algoritma za izdvajanje značajki, a to su HOG, LBP I LDA. Eksperimentom je utvrđeno da najbolje rezultate daju modeli mreža trenirani nad skupom podataka gdje su značajke izdvojene LDA algoritmom. Na kvalitetu modela utječe i izbor postupka izgradnje strukture mreže. Eksperimentom je pokazano da algoritam PSO daje bolje rezultate nego algoritam nasumične izgradnje i algoritam K-means. Također je eksperimentalno utvrđeno da radijalna neuronska mreža postiže nešto bolje performanse od nekolicine drugih klasifikatora iz literature, što ju čini prikladnim klasifikatorom za ovaj problem.

**Ključne riječi:** izdvajanje značajki, klasifikacija, raspoznavanje emocija iz izraza lica, radijalna neuronska mreža, strojno učenje

## Abstract

We can find classification problem in many different types of scientific areas and because of that in the last couple of decades there is a lot of proposed procedures which should be able to solve this problem. Facial emotion recognition is one of the problems that can be solved by designing classification model. Classification models have ability to join predefined class labels to each sample contained inside input dataset. In this paper, radial basis function network is used as classifier and results show that this type of classifier is able to successfully solve classification problem. For classification model to be made, it is necessary to define dataset which contains relevant information for solving a given problem. To build classifier with ability to solve facial emotion recognition problem FER2013 dataset was used and experiment results shown that it is possible to build reliable classification model with this dataset. Problem with this dataset is data representation. Data is represented by images and classification algorithms can not work with that type of input. Because of that, feature extraction procedure is carried out on input dataset where essential information is extracted from images. Extracted data then can be used by classification algorithms as input. In this paper three algorithms for feature extraction were described - HOG, LDA and LDA. Experiment results show that models trained on dataset whose features were extracted with LDA algorithm gave best results. Model quality is also influenced by classification algorithm that is used for training. Experiment results show that PSO algorithm gives better results then random algorithm and K-means algorithm. Beside radial basis function network there are other types of classifiers that can be used for solving classification problems e.g. random forest algorithm and support vector machines. Also, it is experimentally determined that radial neural network achieves better performance in comparison to several other classifiers from literature, which makes it a suitable classifier for this problem.

**Keywords:** feature extraction, classification, facial emotion recognition, radial basis function network, machine learning



## Životopis

Mislav Štiglec rođen je 17.5.1999. godine u Osijeku. U Osijeku završava osnovnu školu i 2014. godine upisuje Prirodoslovno-matematičku gimnaziju Osijek. Nakon srednje škole upisuje preddiplomski i diplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Tijekom studija višestruki je dobitnik STEM državne stipendije te stipendije Osječko-baranjske županije na temelju uspješnosti tijekom studija. Od 2021. godine do danas kao student na poziciji softver developera odrađuje praksu u Ericsson Nikola Tesla grupi u Osijeku na razvoju 5G tehnologija.