



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA TELEKOMUNIKACIJE

Klasifikacija objekata na slici korištenjem superviziranog učenja

- DRUGI CIKLUS STUDIJA -

Autor:
Halać Delila

Mentori:

Doc.dr Emir Turajlić,
dipl.ing.el.

Doc.dr Emir Sokić,
dipl.ing.el.

Sarajevo,
septembar, 2017.

**Doc. dr Emir Sokić, dipl.ing.el.
Doc. dr Emir Turajlić, dipl.ing.el.
Elektrotehnički fakultet u Sarajevu
Odsjek za telekomunikacije**

Klasifikacija objekata na slici korištenjem superviziranog učenja

U industrijskom okruženju se nerijetko pojavljuje potreba za vizuelnom inspekcijom objekata koje nije moguće eksplisitno opisati analitičkim modelima. Kao primjer, korištenje tradicionalnih metoda klasifikacije u oblasti kontrole kvaliteta hrane (voća, mesa, orašastih plodova, keksa i dr.) nije pogodno zbog izuzetno velike varijacije vizuelnih značajki (boje, oblika, dimenzija, teksture) objekata koji pripadaju istoj klasi. Kod rješavanja ovog tipa problema se pribjegava korištenju heurističkih metoda i metoda mašinskog učenja. U ovom radu je potrebno obraditi prednosti i nedostatke korištenja heurističkih metoda i superviziranog učenja za klasifikaciju objekata na digitalnim slikama.

Postavka zadatka:

U okviru rada potrebno je:

- napraviti pregled literature vezan za ekstrakciju osnovnih vizuelnih značajki objekata (oblik, boja, tekstura),
- napraviti pregled literature vezan za korištenje metoda superviziranog učenja u klasifikaciji vizuelnih značajki,
- implementirati barem tri različite metode za klasifikaciju, sa akcentom na metode koje zahtijevaju manji testni skup ili daju rezultat u (skoro) realnom vremenu,
- izvršiti komparativnu analizu implementiranih algoritama korištenih u radu uz pomoć sintetiziranog skupa digitalnih slika.

Koncept i metode rješavanja:

Rad se treba sastojati iz sljedećih cjelina:

- Pregled literature i opis najznačajnijih metoda ekstrakcije vizuelnih značajki, te opis najznačajnijih metoda superviziranog učenja,
- Implementacija sistema za segmentaciju objekata i ekstrakciju vizuelnih značajki, kao i algoritama za odabir relevantnih značajki, te algoritama za klasifikaciju u MATLAB i/ili C++/OpenCV okruženju,
- Sinteza skupa digitalnih slika za treniranje/testiranje/validaciju.

- Eksperimentalna analiza implementiranih algoritama (brzina izvršavanja, tačnost klasifikacije).

Polazna literatura:

- [1] Szeliski, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.
- [2] Sonka, Milan, Vaclav Hlavac, and Roger Boyle. Image processing, analysis, and machine vision. Cengage Learning, 2014.
- [3] Russ, John C. The image processing handbook. CRC press, 2016.
- [4] Costa, Luciano da Fontoura Da, and Roberto Marcondes Cesar Jr. Shape analysis and classification: theory and practice. CRC Press, Inc., 2000.
- [5] Russ, John C. Image analysis of food microstructure. CRC press, 2004.
- [6] Du, Cheng-Jin, and Da-Wen Sun. "Learning techniques used in computer vision for food quality evaluation: a review." Journal of food engineering 72.1 (2006): 39-55.
- [7] Costa, Corrado, et al. "Shape analysis of agricultural products: a review of recent research advances and potential application to computer vision." Food and Bioprocess Technology 4.5 (2011): 673-692.
- [8] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar Foundations of Machine Learning MIT Press, 2012.

Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu
Odsjek za telekomunikacije

Izjava o autentičnosti radova

Završni rad
II ciklusa studija

Ime i prezime: Delila Halać

Naziv rada: Klasifikacija objekata na slici korištenjem superviziranog učenja

Vrsta rada: Završni rad II ciklusa studija

Broj stranica: 92

Potvrđujem:

- da sam pročitao/la dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjestan/na univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio/la prisustvo citiranog ili parafraziranog materijala i da sam se referirao/la na sve izvore;
- da sam dosljedno naveo/la korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio/la svaku pomoć koju sam dobio/la pored pomoći mentora/ice i akademskih tutora/ica.

Datum

Delila Halać

Sadržaj

Popis slika	vii
Popis tabela	viii
1 Uvod	3
2 Pregled stanja u oblasti istraživanja	5
2.1 Preprocesiranje slike	5
2.2 Segmentacija slike	6
2.3 Izdvajanje i selekcija značajki	7
2.4 Klasifikacija	8
3 Preprocesiranje slike	11
3.1 Segmentacija slike	12
3.1.1 Segmentacija pragom	12
3.1.2 Segmentacija klasterizacijom	14
3.1.3 Segmentacija na bazi diskontinuiteta	17
3.1.4 Segmentacija na bazi regije	17
4 Baza slika	19
4.1 Klasa I	19
4.2 Klasa II	20
4.3 Klasa III	20
4.4 Klasa IV	20
4.5 Klasa V	21
5 Ekstrakcija i selekcija značajki	24
5.1 Značajke oblika	24
5.1.1 Ekscentricitet	25
5.1.2 Dužina velike ose	26
5.1.3 Površina i obim	26
5.1.4 Cirkularnost	26
5.1.5 Poluprečnik	27
5.1.6 Kružnost	28
5.1.7 Omjer dužine i širine badema	28
5.2 Značajke boje	28
5.2.1 Modeli boja	28
5.3 Selekcija značajki	31
5.4 Ekstrakcija značajki	32

5.4.1	<i>Principal Component Analysis - PCA</i>	32
5.4.2	Analiza značajki badema pomoću metode glavnih komponenata	35
6	Metode klasifikacije	43
6.1	Pojam klasifikacije	43
6.2	Mašine vektorske podrške - SVM	45
6.3	Neuronske mreže	47
6.3.1	Historijski razvoj neuronskih mreža	47
6.3.2	Arhitektura neuronskih mreža	48
6.3.3	Aktivacijske funkcije	50
6.3.4	Učenje i treniranje neuronske mreže	51
7	Primjena metoda klasifikacije na postojeću bazu	53
7.1	Primjena mašina vektorske podrške	53
7.1.1	”Jedan naspram svih” pristup	53
7.1.2	”Jedan naspram jedan” pristup	56
7.1.3	Hibridni pristup	58
7.1.4	Poređenje rezultata SVM metoda	62
7.2	Primjena neuronskih mreža	64
7.2.1	Neuronske mreže sa N klase	64
7.2.2	Neuronske mreže sa M klase i P stanja	65
7.2.3	Hibridne neuronske mreže	66
7.2.4	Poređenje metoda neuronskih mreža	67
8	Zaključak	69
Prilozi		70
A	Prilog	71
A.1	Razvoj korisničkog interfejsa za SVM i NN hibridni pristup	71
A.2	Programski kodovi	72
Literatura		80

Popis slika

3.1	Klasifikacija segmentacije	12
3.2	Originalna slika	13
3.3	Slika nakon segmentacije	13
3.4	Tipovi klasterizacije	14
3.5	Procedura izvršavanja <i>K-means</i> algoritma	15
3.6	Ulazni podaci sa slike	16
3.7	Segmentacija K-means algoritmom	16
3.8	Originalni badem	16
3.9	Segmentirana slika	17
3.10	Objekt od interesa	17
4.1	Sirovi čitavi bademi	20
4.2	Sirovi slomljen badem	20
4.3	Sirovi oštećeni badem	20
4.4	Blanširani čitavi badem	20
4.5	Blanširani slomljeni badem	20
4.6	Pečeni čitavi badem	21
4.7	Pečeni slomljen badem	21
4.8	Pečeni oštećen badem	21
4.9	Pečeni-blanširani čitavi badem	21
4.10	Pečeni-blanširani slomljeni badem	21
4.11	Nepoznata klasa - lješnjak	22
4.12	Nepoznata klasa- lješnjak	22
5.1	Osnovne oznake elipse	26
5.2	Kružna cirkularnost [1]	27
5.3	RGB i CMY model boja [2]	30
5.4	Glavne komponente su ortogonalne. Ukoliko je varijansa PC_2 mala, onda se zanemaruje i dimenzionalnost se redukuje sa dva na jedan [3]	33
5.5	Usporedba i analiza klase 1 i 2	36
5.6	Usporedba i analiza klase 1 i 3	37
5.7	Usporedba i analiza klase 1 i 4	38
5.8	Usporedba i analiza klase 4 i 6	39
5.9	Usporedba i analiza klase 4 i 5	40
5.10	Usporedba i analiza klase 6 i 9	41
5.11	Usporedba i analiza klase 1 i 11	42
6.1	Pojednostavljena blok struktura modela za izgradnju klasifikatora	44
6.2	Klasifikacija korištenjem SVM klasifikatora	45

6.3	Primjer jednog neurona	48
6.4	Struktura jednog neurona	49
6.5	Struktura višeslojne neuronske mreže	50
6.6	Primjeri aktivacijskih funkcija	51
7.1	Stablo hibridnih klasifikatora za SVM i neuronske mreže	58
7.2	Primjer razdvajanja klase pečenih i nepečenih badema pomoću SVM metode	61
A.1	Izgled korisničkog interfejsa za SVM i NN hibridni metod	71
A.2	Izgled korisničkog interfejsa za SVM i NN hibridni metod	72
A.3	Prikaz tačno klasificirane odabrane slike u C++	72
A.4	Prikaz netačno klasificirane odabrane slike u C++	72

Popis tabela

2.1	Tehnike ekstrakcije značajki [4]	8
4.1	Broj slika po klasama u ovisnosti od stanja badema	22
4.2	Broj slika po klasama	23
5.1	Izdvojene značajke oblika i boje za bademe	25
7.1	Nasumično odabrani uzorci slika u omjeru 50 : 50	54
7.2	Nasumično odabrani uzorci slika u omjeru 80 : 20	54
7.3	Nasumično odabrani uzorci slika u omjeru 50 : 50	56
7.4	Nasumično odabrani uzorci slika u omjeru 80 : 20	56
7.5	Nasumično odabrani uzorci slika u omjeru 50 : 50	59
7.6	Nasumično odabrani uzorci slika u omjeru 80 : 20	59
7.7	Kernel funkcije i značajke korištene za hibridnu SVM baziranu klasifikaciju	61
7.8	Rezultati klasifikacije SVM metoda za omjer 50(%):50(%) za isti skup testnih i trening slika	62
7.9	Rezultati klasifikatora sa trening i test slikama u omjeru 80(%):20(%) . . .	62
7.10	Matrica konfuzije klasa za SVM hibridni pristup	62
7.11	Matrica konfuzije stanja za SVM hibridni pristup	63
7.12	Nasumično odabrani uzorci slika u omjeru 50 : 50 za neuronsku mrežu sa N klasa	64
7.13	Nasumično odabrani uzorci slika u omjeru 80 : 20 za neuronsku mrežu sa N klasa	65
7.14	Nasumično odabrani uzorci slika u omjeru 50 : 50 za neuronsku mrežu sa M klasa i P stanja	65
7.15	Nasumično odabrani uzorci slika u omjeru 80 : 20 za neuronsku mrežu sa M klasa i P stanja	65
7.16	Parametri neuronskih mreža	66
7.17	Nasumično odabrani uzorci slika u omjeru 50 : 50 za hibridnu neuronsku mrežu	67
7.18	Nasumično odabrani uzorci slika u omjeru 80 : 20 za hibridnu neuronsku mrežu	67
7.19	Rezultati klasifikatora sa trening i test skupom u omjeru 50(%):50(%) . . .	67
7.20	Rezultati klasifikatora sa trening i test skupom u omjeru 80(%):20(%) . . .	67
7.21	Matrica konfuzije klasa za NN hibridni pristup	68
7.22	Matrica konfuzije stanja za NN hibridni pristup	68

Zahvala

Zahvaljujem svom mentoru doc. dr. Emiru Sokiću na strpljenju, pomoći i korisnim diskusijama tokom izrade magistarskog rada. Hvala na mnogobrojnim savjetima i stručnoj pomoći koje su mi uveliko olakšali uspješno pisanje i izradu ovog rada. Također se zahvaljujem i mentoru doc. dr. Emiru Turajliću za pomoć i podršku tokom pisanja rada. Čitavoj komisiji hvala na trudu i uloženom vremenu za čitanje, te na korisnim savjetima i sugestijama koje su ovaj rad učinili još boljim.

Zahvaljujem i svim prijateljima, koji su mi uljepšali i olakšali ovih pet godina na bilo koji način, koji su cjelokupno razdoblje mog studiranja učinili nezaboravnim i najljepšim razdobljem mog života.

Posebnu zahvalnost iskazujem cijeloj svojoj porodici i tetki na podršci i razumijevanju koju su mi pružali tijekom trajanja studija.

I na kraju, najveću zaslugu za ono što sam postigla dugujem svojim roditeljima ocu i majci koji su uвijek bili tu uz mene i bez kojih sve ovo što sam postigla ne bi bilo moguće.

Sažetak

U okviru rada dat je postupak izgradnje sistema za klasifikaciju objekata na digitalnim slikama (bademi) baziran na metodama mašinskog učenja. Glavni dio rada obuhvata proces izdvajanja objekata na slici, koji podrazumijeva segmentaciju i preprocesiranje slike, potom ekstrakciju značajki oblika i boje izdvojenog objekta i na kraju klasifikaciju objekata koristeći supervizirano učenje.

Također je izvršena analiza značajki pomoću metode glavnih komponenata u svrhu poboljšanja tačnosti algoritama. Razvijen je novi hibridni metod zasnovan na mašinama vektorske podrške i neuronskim mrežama. Izvršeno je poređenje rezultata razvijenih algoritama po pitanju tačnosti i brzine, gdje se implementirani hibridni metod zasnovan na mašinama vektorske podrške pokazao kao najbolji po pitanju tačnosti i vremenskih performansi, što ga čini povoljnijim za korištenje u realnom vremenu. Na kraju radi bolje preglednosti i uvida u rezultate klasifikacije izgrađen je korisnički interfejs u programskom paketu MATLAB.

Abstract

The paper presents the procedure of constructing a system for object classification on digital images (almonds) based on the methods of machine learning. The main part of this paper consists of the extraction of objects in images, which includes segmentation and preprocessing techniques, extraction of features (shape and color) and consequently classification of objects using supervised learning.

In order to improve classification results feature analysis has been done with the method of principal component analysis. A novel algorithm for classification is developed using support vector machine and neural networks. Comparison of results regarding accuracy and time performance has been conducted. The proposed SVM based algorithm exhibits best result both in accuracy and time performance, which made it suitable for real-time usage. In order to visualize the classification results, a user interface has been developed.

Poglavlje 1

Uvod

Evaluacija kvalitete je jedan od ključnih faktora koji ima utjecaj na konačnu cijenu hrane u prehrambenoj industriji. Danas tehnike procesiranja slike postaju sve zastupljenije u proceduri ocjene kvalitete hrane. Također, povećani zahtjevi od strane klijenata za kvalitetniju hranu su doveli do bržeg razvoja automatske inspekcije hrane u prehrambenoj industriji.

Značajke poput oblika, boje, teksture i slično nose semantičko značenje i sadrže mnoštvo informacija o samom objektu i mogu se smatrati glavnim sredstvom komunikacije koja se odvija uz pomoć vizuelnih informacija [5]. Kvaliteta hrane se najčešće povezuje sa njenim oblikom i predstavlja jedan od bitnih parametara u prehrambenoj industriji, te je potrebno da se procjena kvalitete odvija brzo i tačno. Obzirom da u industrijskom okruženju lahko može doći do oštećenja hrane, to se najčešće očituje u promjeni njihovog oblika, stoga je bitno izdvojiti karakteristike od interesa i prepoznati vizualne nedostatke hrane kako bi se ona odstranila iz dalje upotrebe [6]. Također i druge značajke poput boje i teksture igraju važnu ulogu pri klasificiranju hrane. Promjenom boje i teksture na hrani se mogu izvući bitne informacije za dalju obradu. Stoga je bitno da sve značajke budu adekvatno izdvojene kako bi se proces prepoznavanja odvijao brzo i tačno.

U posljednje vrijeme automatska inspekcija sistema, većinom bazirana na kamerama i kompjuterima, koristi se za vizuelnu inspekciju kvalitete hrane. Pristup poznat pod nazivom kompjuterska vizija (engl. *computer vision*) se pokazao kao vrlo uspješan za objektivna mjerjenja različitih proizvoda hrane. Ovaj sistem uključuje procesiranje, analiziranje slike i ocjenjivanje vizuelnih karakteristika hrane.

Upravo zbog ovih dobrih karakteristika tradicionalno ocjenjivanje hrane od strane ljudi, koje je sporo i neobjektivno, se polako zamjenjuje automatiziranim mašinama koje rade taj posao. Na ovakav način se postiže ekonomičnost, konzistentnost i objektivnost.

Kod rješavanja ovakvih vrsta problema najčešće se koriste heurističke metode i metode mašinskog učenja. Mašinsko učenje je grana vještačke inteligencije i podoblast računarstva, koja se bavi izučavanjem i razvojem algoritama, sposobnih da uče na osnovu unesenih podataka i predviđaju podatke na osnovu procesa treniranja. Kako uspjeh algoritma učenja zavisi od korištenog seta podataka, mašinsko učenje je usko povezano sa analizom podataka i statističkom teorijom [7]. Dva velika naučnika u ovoj oblasti Arthur Samuel i Tom Mitchell, definirali su mašinsko učenje kao:

Oblast nauke koja daje računaru mogućnost da uči, a da prethodno ne bude eksplicitno programiran.

Posljednjih godina ova oblast se počela naglo razvijati, a problem klasifikacije nepoznatih instanci unaprijed predefinisane klase je jedan od najčešćih problema mašinskog učenja.

Za uspješnu dodjelu oblika određenoj kategoriji potrebno je izvršiti nekoliko generalnih koraka [5]:

- preprocesiranje slike (akvizicija slike, odstranjivanje šuma, primjena operacija poput normalizacije, skaliranja i slično),
- segmentacija slike - pronalazak regiona od interesa,
- izdvajanje i selekcija značajki (engl. *feature extraction and selection*),
- klasifikacija.

U okviru ovog rada izvršena je analiza badema na osnovu oblika i boje te njihova klasifikacija pomoću mašina vektorske podrške (engl. *Support Vector Machine - SVM*) i neuronskih mreža (engl. *Neural Networks - NN*). Za bazu slika su odabранe različite vrste badema (sirovi, blanširani, pečeni, blanširani - pečeni), koje mogu biti čitave, slomljene, ili oštećene. Na osnovu klasifikacijskih algoritama potrebno je uspješno prepoznati kojoj kategoriji pripada badem i shodno tome donijeti zaključak da li se radi o ispravnom ili neispravnom bademu.

U drugom poglavlju je dat pregled stanja u oblasti istraživanja (engl. *State of the art*), koji daje uvid u rade druge autora koji su radili na istim ili sličnim problemima. Poglavlje daje kratak uvid u dosad razvijene tehnike preprocesiranja slike i algoritama mašinskog učenja.

Treće poglavlje je posvećeno metodama i tehnikama preprocesiranja slike koje predstavljaju ključni korak za dalju analizu slike. Dat je poseban osvrt na metode segmentacije, te njihovim prednostima i nedostacima. Posebna pažnja je posvećena segmentaciji klasificacijom koja je korištena pri izdvajanju badema sa slike.

Četvrti dio opisuje kreiranu i korištenu bazu slika i njenu organizaciju. Prikazani su reprezentativni uzorci pojedinih klasa badema. Formirana baza je organizirana u dvije kategorije u skladu sa odgovarajućim tabelama.

U petom dijelu, predmet istraživanja su metode ekstrakcije i redukcije značajki oblika i boje. Analizirane su značajke boje i oblika. U svrhu redukcije dimenzionalnosti izvršena je analiza značajki pomoću PCA metode. Na osnovu dobijenih rezultata zaključeno je koje značajke je pogodno koristiti za klasifikaciju različitih vrsta badema.

U šestom poglavlju se razmatra problem klasifikacije koji predstavlja razvrstavanje nepoznate instance u jednu od unaprijed ponuđenih kategorija. U ovom dijelu poseban osvrt je na klasifikacijskim algoritmima, koji su korišteni u eksperimentalnim istraživanjima.

Sedmi dio daje razmatranje postavki eksperimentalnog istraživanja, prikaz rezultata istraživanja za različite metode prethodnog učenja, za svaki klasifikacijski algoritam posebno. Razmatraju se tačnost i preciznost kojima se mjeri uspješnost korištenog modela.

U osmom poglavlju je dat zaključak i smjernice za buduća istraživanja. Na kraju je dat prilog, u kome su smješteni analize, kodovi i korisnički interfejs.

Poglavlje 2

Pregled stanja u oblasti istraživanja

Industrijska procjena kvalitete hrane je posljednjih godina dosta napredovala i razvijene su nove metode koje olakšavaju vizuelnu procjenu kvalitete hrane. U skladu s tim izvršen je pregled literature u ovoj oblasti istraživanja, sa osvrtom na metode koje najviše doprinose razvoju automatizirane inspekcije hrane.

2.1 Preprocesiranje slike

Prilikom akvizicije digitalne slike, mogu se pojaviti mnogi problemi kao što su: velika količina šuma, granice objekta nisu jasno izražene, osvjetljenje nije pogodno, kontrast nije dovoljno oštar i sl. Stoga je potrebno izvršiti preprocesiranje slike, što predstavlja elementarni korak za dalju obradu, jer potiskuje informacije koje nisu bitne za nastavak analize [2]. Svrha preprocesiranja slike je da naglasi podatke na slici koje su bitne za dalje procesiranje i da stvori pogodniju sliku od originalne za datu aplikaciju. Naravno, u ovisnosti koje metode ekstrakcije značajki su primjenjene zavisi koji će se način preprocesiranja slike koristiti. Npr. lokalni binarni deskriptor (engl. *Local binary descriptor*) koristeći crno-bijele podatke će zahtijevati drugačije preprocesiranje nego SIFT (engl. *Scale Invariant feature transform*) koji koristi podatke u boji [8].

Akvizicija slike predstavlja prvi korak u obradi slike, jer bez njenog postojanja dalja obrada nije moguća. Tip uređaja koji se koristi zavisi od namjene, a može biti skener, kamera i slično [5, 9]. U radu [10] akvizicija slike je izvršena pomoću skenera kako bi se dobila originalna slika riže za dalju obradu, a potom je urađeno izglađivanje slike pomoću median filtera kako bi se dobile jasno izražene granice, te otklonio šum. Ovim je slika spremna za dalju obradu. Otklanjanje šuma se vrši različitim tehnikama filtriranja [5]. Vrste filtera koji se mogu primjenjivati su detaljno opisani u [11].

Također, osvjetljenje je bitno za dalju analizu, jer procjena kvaliteta hrane na slici može biti uveliko olakšana, ili otežana u ovisnosti od uslova osvjetljenosti. Načini osvjetljenja, prednosti i nedostaci su obradene u [12]. Naravno, različite primjene zahtijevaju različite uslove osvjetljenja a prema [13] osvjetljenje se može grupisati u: prednju osvjetljenost (engl. *front lightning*), zadnju osvjetljenost (engl. *back lightning*) i strukturalnu osvjetljenost (engl. *structured lightning*). Ova podjela je napravljena u skladu sa pozicijom osvjetljenja promatranog objekta.

Ukoliko je za dalju obradu bitna boja, tada se u preprocesiranju mogu koristiti različiti prostorne transformacije, kao što je RGB u CIE L*a*b* koji predstavlja trodimenzionalni prostor boja i daje dobru aproksimaciju u skladu sa ljudskom percepcijom. Klasifikacija hrane u radu [14] je izvršena upravo pomoću CIE L*a*b* analize boja. Detaljne transformacije boja dobivene korištenjem različitih formata boja se nalaze u [2, 9, 15] kao što su RBG, CMY, YUV, HSV, L*a*b*.

Kao dio preprocesiranja slike može se izvršiti normalizacija, skaliranje, invarijantnost na rotaciju i slično, pri čemu slika postaje pogodnija za dalju obradu. Ove operacije su detaljnije obrađene u [5]. Na ulaznoj slici se također mogu promjeniti dimenzije na fiksiranu veličinu, jer u daljim koracima je potrebno raditi sa slikom konstantnih dimenzija.

2.2 Segmentacija slike

Segmentacija dijeli sliku na njene regije, ili objekte. Stepen detaljnosti zavisi od primjene, tj. segmentacija prestaje kada se detektuju regioni koji su od interesa. Stoga, uvijek treba definisati cilj, odnosno šta je potrebno detektovati da bi se smanjilo vrijeme segmentacije. Segmentacija netrivijalnih slika može predstavljati najteži dio obrade slike. Od tačnosti segmentacije zavisi uspjeh daljeg procesa analize slike. Metode segmentacije se dijele na: metode praga, konturno bazirane metode, regijski bazirane metode i segmentaciju klasterizacijom.

Metoda praga je najjednostavnija metoda segmentacije kod koje se određena konstanta osvjetljenja proglašava pragom i koristi za razdvajanje objekta od pozadine [16]. Prag može biti lokalni ili globalni, zavisno koja se metoda koristi. Lokalni prag varira ovisno o dijelu slike na koju je primijenjen i o lokalnim osobinama slike izračunatih za taj dio. Globalni prag ostaje konstantan za sve piksele na slici i zavisi samo od histograma nivoa sive na slici [16]. Postoje različite metode kao što je Niblackov, Otsu algoritam i drugi. Ipak, izbor vrijednosti praga još uvijek predstavlja težak zadatak i predmet istraživanja mnogih radova [2, 5, 15, 17]. Međutim ove metode nisu pogodne za složenije slike, odnosno slike koje imaju dosta detalja, kao i one gdje nivo osvjetljenja nije prilagođen.

Konturno bazirane metode se zasnivaju na tome da se odrede konture objekta pod pretpostavkom da je kontura zatvorena kriva koja definiše objekat. Ove metode nisu pogodne za slike gdje kontura nije jasno izražena. Ivice objekta se mogu odrediti različitim transformacijama i heurističkim tehnikama [2].

Regijske bazirane metode podrazumijevaju izdvajanje cijelog objekta od pozadine, gdje pikseli, čiji je intenzitet ispod nekog praga, pripadaju pozadini, a oni, čiji je intenzitet iznad praga pripadaju objektu [2], odnosno predstavljaju dijeljenje slike na homogene oblasti povezanih piksela primjenom kriterija homogenosti. Metoda je pogodna za slike koje nemaju jasno izražene ivice i gdje je prisutno dosta šuma.

Segmentacija klasterizacijom predstavlja jednu od najstarijih tehnika segmentacije. Klasterizacija predstavlja identifikovanje grupa (klastera) objekata, tako da se unutar jedne grupe nalaze međusobno slični i istovremeno različiti objekti u odnosu na druge grupe, pri čemu je potrebno da rastojanje između objekata unutar klastera bude minimalno, a rastojanje između klastera je maksimalno [18]. Postoje različite metode za ovu vrstu segmentacije, a jedna od najpoznatijih je *K-means* klasterizacija [18]. Rezultat ovog algoritma uveliko ovisi od odabrane mjere udaljenosti.

Ipak, koja metoda segmentacije će se primijeniti zavisi od dalje obrade slike. Prva dva koraka (preprocesiranje i segmentacija) će uveliko uticati na nastavak analize slike i rezultate klasifikacije, te je stoga potrebno izabrati adekvatan metod, ovisno o vrsti slike koja se obrađuje.

2.3 Izdvajanje i selekcija značajki

Da bi se prevazišli problemi koje sa sobom nosi visoka dimenzionalnost podataka, potrebno ju je smanjiti. Ovo se može uraditi tako što se izabere samo podskup relevantnih značajki, ili kreiranjem novih značajki koji sadrže maksimum informacija o dатој klasi. Prva metodologija se zove selekcija značajki, dok se druga zove ekstrakcija značajki.

Selekcija značajki (engl. *feature selection*) predstavlja važan korak u pripremi podataka za klasifikaciju, jer se reducira dimenzionalnost i olakšava kreiranje sistema za prepoznavanje oblika. Primjenjuje se kada je dimenzionalnost velika, a broj observacija mali, odnosno kada slika sadrži veliku količinu informacija koje nisu bitne za dalju analizu, a ovim korakom se biraju samo one informacije koje doprinose kvaliteti. U najjednostavnijem smislu selekcija osobina se može posmatrati kao proces traženja podskupa osobina za koje konačni model ima najmanju grešku [19]. Selekcija značajki uključuje: SFS (engl. *sequential forward selection*) sa Fisher diskriminantom, KNN (engl. *k-nearest neighbor*), LDA (engl. *linear discriminant analysis*) [20], QDA (engl. *quadratic discriminant analysis*), FOSMOD (engl. *forward orthogonal search algorithm to maximize the overall dependency*) i druge. Ove metode su objašnjene u [21, 22].

Ekstrakcija značajki se razlikuje od selekcije značajki jer selekcija značajki kao rezultat vraća podskup originalnih osobina, a ekstrakcija značajki kreira nove osobine na osnovu originalnih osobina i pri tome reducira ukupni broj osobina, odnosno dimenzionalnost [19]. Neke od najpoznatijih metoda za ekstrakciju značajki su: Haar-like-features, Histogram of oriented gradients (HOG), Scale invariant feature transform (SIFT), Speeded up robust feature (SURF), Houghova transformacija, Principal Component Analysis (PCA) [15, 23], samoorganizirajuće mape (SOM)[24], ICA (engl. *Independent component analysis* i drugi koji su obrađeni u [11, 12, 24, 25, 26]. Dobar metod ekstrakcije značajki uzima samo bitne osobine objekta. Tako na primjer, PCA transformiše skup observacija koji su potencijalno korelirane varijable u skup linearne nekoreliranih varijabli koje se nazivaju glavne komponente. Detaljan način rada PCA je dat u [19, 23].

Značajke se mogu svrstati u nekoliko kategorija: značajke boje, značajke teksture, značajke oblika i druge metode ekstrakcije značajki prikazane u tabeli 2.1. [4]. Postoji još mnogo podjela prema kojim je izvršena podjela ekstrakcije značajki. Tako prema [35] ekstrakcija značajki se svrstava u dvije kategorije: statičke i strukturalne ekstrakcije značajki. Strukturalne ekstrakcije podrazumijevaju da se koriste strukturalne osobine, bazirane na topografskim i geometrijskim osobinama objekta. Primjeri strukturalnih značajki su: horizontalne i vertikalne linije, broj krajnjih tačaka i sl. U statičkim ekstrakcijama značajki se podrazumijeva da se koriste statičke osobine objekta kao što su: *zoning*, projekcija histograma i slično. Ove metode se mogu lako detektovati i uporediti sa strukturalnim. U radu [36] za klasifikaciju lješnjaka korištene su metode standardne ekstrakcije značajki u koje spada srednja vrijednost, standardna devijacija, srednji gradijent i drugi. Potom slijedi invarijantnost momenata, te tekstualne značajke u koje spada Harlick i Gaborova metoda.

Tabela 2.1: Tehnike ekstrakcije značajki [4]

Ime	Metoda
PHOG	Pyramid Histogram of Oriented Gradients [27]
CONTRAST	Contrast Features[28]
FITELLIPSE	Ellipse Features [29]
FOURIER	Fourier Features [30]
FOURIERDES	Fourier Descriptors [31]
GABOR	Gabor Features [32]
GUPTA	Three Gupta Moments Features
HARALICK	Haralick Texture Feature
HUGEO	The Seven Hu Moments Features
HUINT	The Hu Moments with Intensit Features
LBP	Local Binary Patterns Features [33, 34]
MOMENTS	Moments and Central Moments Features
BASICGEO	Geometric Features
BASICINT	Basic Intensity Features

U radu [9] za opis objekta od interesa uzeta je veličina, opisana sa površinom, dužinom, širinom i obimom. Sljedeća osobina koja je uzeta u razmatranje je oblik, jer je objekte najlakše prepoznati pomoću oblika. Za ekstrakciju oblika korištene su različite metode navedene u [9]. Za razlikovanje dva objekta su također bitni boja i tekstura.

Nakon što su najvažnije karakteristike oblika predstavljene numerički, u obliku vektora, potrebno je istrenirati mrežu pomoću trenirajućeg seta podataka kako bi se izvršila klasifikacija.

2.4 Klasifikacija

U mašinskom učenju postoje tri vrste problema učenja:

- nadgledano/supervizirano učenje (engl. *supervised learning*) koje podrazumijeva da se algoritmu zajedno sa podacima daju i željeni izlazi (trening podaci). Primjer nadziranog/superviziranog učenja je klasifikacija,
- nenadgledano/nesupervizirano učenje (engl. *unsupervised learning*) podrazumijeva da se algoritmu koji uči daju samo podaci bez izlaza, a od algoritma se očekuje da sam uoči zakonitosti u podacima koji su mu dati. Tipični primjer nesuperviziranog učenja je klasterizacija,
- pojačano učenje (engl. *Reinforcement learning*) koje je blisko nadgledanom/superviziranom učenju, s tim da povratna informacija nije ispravka greške nego obavještenje da je došlo do nje.

Klasifikacija pokušava pronaći pravila za razdvajanje uz pomoć već poznatih klasa (grupa), bez obzira na distance i razlike među grupama. Stoga za klasifikaciju je bitno izabrati odgovarajući trenirajući skup, pomoću kojeg se treba naučiti algoritam kako klasificirati podatke [2]. U skupu za treniranje svaka instanca je opisana pomoću određenog broja atributa. Zadatak klasifikacije je da na osnovu skupa podataka za treniranje i odgovarajuće

labele, generiše klasifikator koji instance testnog skupa, za koji nije poznata labela, treba da svrsta u adekvatnu klasu [5]. Postoji nekoliko podjela klasifikatora na [9] SL (engl. *statical learning*), fuzzy klasifikacija, klasifikacija korištenjem neuronskih mreža.

SL podrazumijeva da postoji probabilistički model koji pruža statističke informacije pripadnosti određenoj klasi. Postoji više različitih metoda razvijenih u ovoj skupini kao što je Bayesov klasifikator, metod KNN, itd. Ove metode u radu [9] su primjenjene na klasifikaciju voća, gdje se njihovom primjenom dobili zadovoljavajući rezultati klasifikacije. U fuzzy klasifikaciji stepen funkcije pripadnosti može pružiti više informacija o pouzdanosti pripadnosti određenoj klasi. Detaljnija analiza fuzzy sistema je data u [2, 15]. Novije fuzzy klasifikacijske metode se najviše koriste za ocjenjivanje i klasifikaciju različite vrste hrane, dok se neuronske mreže koriste za ocjenjivanje žitarica poput pšenice, ječma, zobi, raž i druge [9].

Kriterij kojim se porede i ocjenjuju metode klasifikacije su: [37]

- tačnost - sposobnost klasifikatora da tačno klasificuje instancu nepoznate vrijednosti atributa klase,
- brzina - predstavlja broj operacija koje treba izvršiti pri konstrukciji i primjeni klasifikatora,
- robusnost - odnosi se na preciznost klasifikatora, kada se primjeni na podatke sa šumom, ili podatke kojima nedostaje vrijednosti nekih značajki,
- skalabilnost - potrebno je da klasifikator može da podnese veću količinu podataka, a da se pri tome ne mora znatno mijenjati.

Najpoznatije tehnike su: stablo odluke (engl. *decision tree*) [37, 38], neuronske mreže - ANN (engl. *Artificial neutral network*) [12, 38], Bayesove mreže [5, 14, 38], metod *K* najbližih susjeda - KNN (engl. *K-nearest neighbor*) [37, 38], SVM (engl. *Support vector machine*) [4, 9, 38], SOM (engl. *Self-organized maps*) [24], Naive Bayes klasifikator [38], LogR (engl. *Logistic Regression*), LR (engl. *linear regression*) [37], ARC (engl. *Area under curve*) [37] i drugi.

Najviše korištena tehnika je SVM sa razlicitim varijacijama. Prednosti ove metode su tačnost, pružanje boljih predviđanja neviđenih test podataka, postojanje manje parametara za optimizaciju, međutim može zahtijevati više vremena za obradu od drugih metoda [38]. U [4] je primjenjen SVM sa Gausovom jezgrom, linerarni SVM, LS SVM (engl. *Least-square SVM*). SVM vrši klasifikaciju koristeći nelinearno mapiranje za transformaciju originalnih podataka u veće dimenzije. U [10, 39] je detaljno objašnjen SVM i njegove varijacije za primjenu klasifikacije za više od dvije klase. Također, učenje na instancama (engl. *Instance learning*) daje pogodne rezultate za određene vrste slika. Ove metode spadaju u kategoriju statističkog učenja i zahtijeva manje vremena tokom treniranja od drugih algoritama, ali im je zato potrebno više vremena tokom klasifikacijskog procesa. Najpoznatiji algoritam u ovoj grupi je najbliži susjed [38]. Metoda najbližeg susjeda -KNN se zasniva na učenju po analogiji, odnosno poredi se testni skup sa trenirajućim. Klasifikacija novih instanci se obavlja na način da se nova instanca poredi sa memorisanim instancama iz skupa za učenje korištenjem definisane metrike [19, 37]. C4.5 je algoritam baziran na drvu odluke koji koristi princip "podijeli pa vladaj" za pravljenje stabla odluke. Princip rada je objašnjen u [37].

U radu [12] su navedene i opisane metode klasifikacije i koje su pogodne za određenu vrstu hrane zajedno se vrstom izdvojenih značajki za pojedinu hranu. Za klasifikaciju lješnjaka u pet klasa u radu [36] je, također korišteno supervizirano učenje i metode poput SVM,

KNN i PNN.

Dobre rezultate je pokazala i tzv. tehnika Prosječni L*a*b* (engl. *Average L*a*b* technique*) [14] koja koristi polazne aproksimacije za klasifikaciju. Tehnika koristi prosječne L*a*b* vrijednosti referentnih slika određene hrane i poredi sa prosječnim L*a*b* vrijednostima testne slike određene hrane.

U zavisnosti od broja klasa klasifikatori se dijeli na:

- binarnu klasifikaciju (engl. *binary classification*),
- višeklasnu klasifikaciju (engl. *multiclass classification*).

U binarnoj klasifikaciji postoje samo dvije klase. Na primjer, da je hrana ispravna, ili neispravna. U višeklasnoj postoji više od dvije klase. U radu [10] je korišten metod multiklasne SVM klasifikacije za sortiranje riže po kvaliteti u tri grupe. Višeklasna klasifikacija je objašnjena u [15].

Problem klasifikacije je još uvijek aktivna tema istraživanja mnogih radova i još uvijek niti jedna metoda ne daje najbolje rezultate za sve vrste problema. Izbor metode zavisi od osobina posmatranog skupa podataka.

Poglavlje 3

Preprocesiranje slike

Analiza slike je grana kompjuterske vizije koja se bavi detekcijom semantičkih objekata na slici. Tako dobijene informacije moguće je iskoristiti za čitav niz korisnih primjena, na primjer za analizu CT snimaka, poboljšanje kompresije videa, klasifikaciju i sl. Digitalno procesiranje slike je zanimljivo područje istraživanja, u kojem se postižu brojna poboljšanja u posljednje vrijeme. Prije same klasifikacije potrebno je izvršiti preprocesiranje slike. Preprocesiranje je naziv koji se koristi za operacije nad slikama. Sastoji se od različitih tehnika manipulacije nad ulaznom slikom kao što su: skaliranje, poboljšanje kontrasta, detekcija ivica, segmentacija, otklanjanje šuma i sl. Ovaj korak predstavlja ključnu operaciju u mašinskom učenju, jer su ovdje izražene najvažnije osobine slike, dok su one manje bitne informacije i šum potisnute ili eliminisane. Stoga, cilj ovog koraka je poboljšanje važnih informacija koje potiskuju neželjene distorzije i naglašavanje bitnih osobina slike za dalje procesiranje. Također, geometrijske transformacije slike kao što su: rotacija, skaliranje i translacija se svrstavaju u preprocesirajuće metode.

U literaturi postoje mnoge metode preprocesiranja slike raspoređene u različite kategorije. Tačnost i preciznost za ove tehnike je iznimno bitna kako bi se osiguralo pouzdano izdvajanje informacija. U [5] metode preprocesiranja su podijeljene na transformaciju piksela svjetline (engl. *pixel brightness transformation*), geometrijske transformacije (engl. *geometric transformation*), metode preprocesiranja koje koriste lokalne susjede i restauracija slike (engl. *image restauration*), koja zahtjeva poznavanje čitave slike.

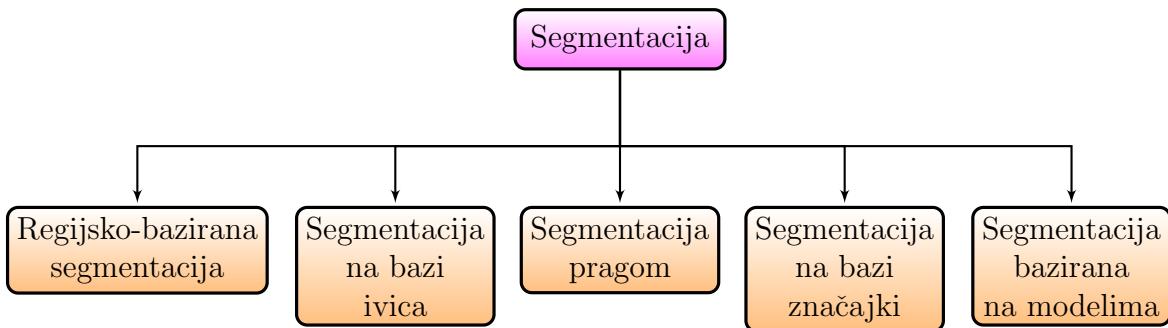
Prvi korak, u preprocesiranju slike, predstavlja akvizicija, jer bez postojanja slike dalja obrada nije moguća. Stoga pomoću kamere, skenera, ili nekog drugog alata potrebno je dobiti slike nad kojima će se vršiti obrada.

Originalna slika se može pojaviti u različitim stanjima, npr. može biti ispunjena šumom. Tada je potrebno primjeniti neku od metoda uklanjanja šuma kao što je filtriranje. Pomoću adekvatnih filtera može se na izlazu dobiti slika sa potpuno, ili djelomično uklonjenim šumom. Najpoznatiji filter je *median filter*, koji uklanja piksele koji se znatno razlikuju od drugih piksela u okolini [5].

Za klasifikaciju i adekvatno prepoznavanje badema prvo je potrebno izvršiti pretpocesiranje slike, a potom segmentaciju, odnosno izdvajanje badema i podloge. U narednoj sekciji je dat opis različitih metoda segmentacije, sa posebnim osvrtom na korištenu metodu u eksperimentalnim istraživanjima.

3.1 Segmentacija slike

Jedan od glavnih elemenata u prepozavanju oblika na slici predstavlja pronalaženje ključnih elemenata slike koji reprezentativno predstavljaju njen sadržaj. Nakon što se ključni elementi pronađu stvaraju se uslovi da se ti elementi porede, klasificiraju i na drugi način koriste u daljoj obradi. Zadatak pronalaženja ovih elemenata ima segmentaciju koja dijeli sliku na regione, ili objekte do određenog nivoa i predstavlja nezaobilazan korak pri automatskoj detekciji oblika i analizi scene. Tačnije, segmentacija slike je proces dodjeljivanja "oznake" svakom pikselu na slici tako da pikseli koji imaju iste oznake se svrstavaju u grupu koja ima iste vizuelne karakteristike. Na kraju rezultat segmentacije je skup kontura pronađenih na slici. Izbor tehnike segmentacije zavisi od same strukture slike, a u literaturi postoje razne podjele. Na slici 3.1 su prikazane neke od osnovnih metoda segmentacije.



Slika 3.1: Klasifikacija segmentacije

Tokom istraživanja se pokazalo da se niti jedan od razvijenih algoritama ne može primjeniti na sve slike, već svaki od njih je pogodan za različite vrste slika. Primjena različitih vrsta algoritama ovisi od rezultata segmentacije koje zavise od više faktora kao što su tekstura, sadržaj slike, boje piksela i slično. Stoga izbor odgovarajućih algoritama postaje važan zadatak. Tri najčešće korištene tehnike za segmentaciju su:

- Segmentacija pragom (engl. *Thresholding*) predstavlja grupu metoda zasnovanu na poređenju osvjetljenosti piksela sa jednim ili više pragova.
- Segmentacija na bazi diskontinuiteta (engl. *Edge-based*) metoda koja se bazira na pronalaženju ivica, odnosno izdvajaju piksela koji pripadaju rubovima objekata. Ovdje se mora voditi računa da su ivice neprekidne.
- Regijsko bazirana segmentacija (engl. *Region-based*) izdvajanje cijelog objekta od pozadine gdje pikseli čija je osvjetljenost ispod nekog praga pripada pozadini, a one čija je svjetlost iznad praga pripada objektu [2].

3.1.1 Segmentacija pragom

Segmentacija pragom (engl. *Thresholding*) predstavlja izdvajanje objekata od pozadine pomoću predefinisanog jednog ili više pragova. Segmentacija sa jednim pragom predstavlja

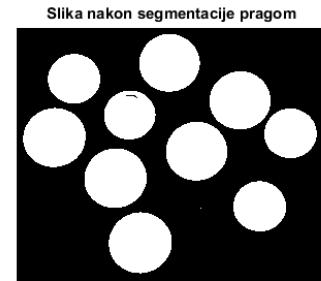
najjednostavniju metodu i zasniva se na tome da ukoliko je piksel veći od praga svrstava se u pozadinu, a ukoliko je manji od praga u objekt. Ovo je prikazano formulom 3.1.

$$B(x, y) = \begin{cases} 1, & \text{za } I(x, y) \leq U \\ 0, & \text{inače,} \end{cases} \quad (3.1)$$

gdje je $B(x, y)$ rezultat segmentacije, a $I(x, y)$ intenzitet određenog piksel na slici. Ova metoda je korisna kod izdvajanja pisanog teksta, analiza biomedicinskih slika i slično. Glavni problem ovog metoda jeste vrijednost praga. Za slike u kojima su objekti isti i gdje osvjetljenje ne varira, prag se može odrediti ručno vizuelnim putem. Međutim, mnogo bolja opcija je da se prag određuje automatski na osnovu karakteristika slike. Postoje dvije vrste pragova:

- lokalni prag - varira ovisno o dijelu slike na koju je primijenjen i o lokalnim osobinama slike izračunatih za taj dio.
- globalni prag - koji ostaje konstantan za sve piksele na slici i zavisi samo od histograma nivoa sive na slici. [16]

Na slici 3.2 je predstavljena je originalna slika uzeta za segmentaciju. Primijenjena je binarna segmentacija pragom, gdje je za prag vizuelnim putem uzeta vrijednost 70. Kod izbora praga mogu nastati problemi zbog šuma i smetnje, neuniformne pozadine i slično. Nakon primjene ove metode dobijena je slika 3.3



Slika 3.2: Originalna slika

Slika 3.3: Slika nakon segmentacije

Pored ove jednostavne metode postoje i razne druge koje se baziraju na upotrebi lokalnog ili globalnog praga. Jedna od njih je binarna segmentacija upotrebom Niblackovim algoritmom. Ova metoda koristi lokalni prag koji se računa prema formuli 3.2. Ideja algoritma se temelji da se prag računa na osnovu srednje vrijednosti m i standardne devijacije σ u prozoru, koji se pomjera na slici, dok k predstavlja parametar koji unosi korisnik i može imati i negativne vrijednosti[2].

$$T = m + k\sigma \quad (3.2)$$

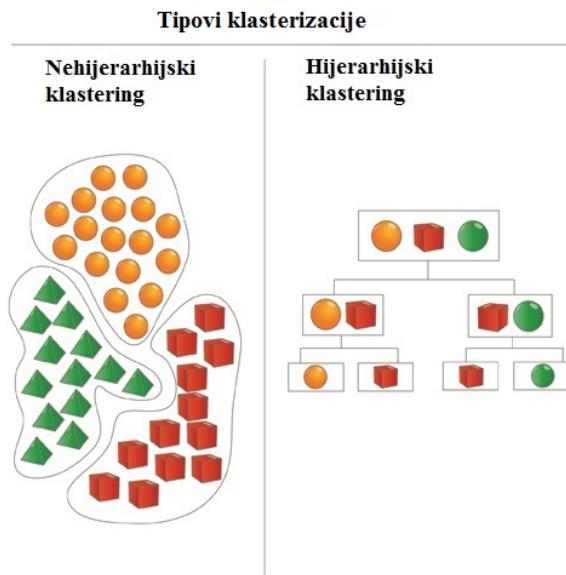
Metode bazirane na globalnom pragu kao što je Otsu metoda, također daju pogodne rezultate za pojedine vrste slika. Algoritmom se prepostavlja odabrana slika, koja je prethodno konvertovana u binarnu, sadrži dvije klase piksela, te se izračunava optimalan prag razdvajanja te dvije klase. Ova metoda je implementirana u MATLAB-u i može se pozvati funkcijom *graythresh*. Algoritam daje i dobre i loše rezultate ovisno od zadane slike (nivo osvjetljenja, broj objekata i slično). Ipak, algoritam se nije pokazao kao efikasan

za složenije slike jer da bi rezultati bili dovoljno dobri zahtijeva se ručno postavljanje praga.

Algoritmi zasnovani na pragu su također primjenjeni na bazu slika badema, međutim niti jedan od ovih algoritama nije dao zadovoljavajuće rezultate. Stoga za adekvatnu segmentaciju slika badema primjenjene su i druge metode opisane u nastavku.

3.1.2 Segmentacija klasterizacijom

Segmentacija klasterizacijom (engl. *Clustering*) predstavlja jednu od najstarijih tehnika segmentacije koja je primjenljiva kako za sive tako i za slike u boji. Klasterizacija predstavlja identifikovanje grupa (klastera) objekata tako da se unutar jedne grupe nalaze međusobno slični i istovremeno različiti objekti u odnosu na druge grupe. Pri čemu je rastojanje između objekata unutar klastera minimalno, a rastojanje između klastera je maksimalno [18]. Kao mjere sličnosti najčešće se koristi euklidska metrika, a mogu se odabrati i neke druge kao što je kosinusna sličnost, Manhattan i druge. Klasterizacija se dijeli na hijerarhijske i nehijerarhijske tehnike. Glavna osobina hijerarhijske tehnike je podjela skupa objekata na ugnježdene klastere koji su hijerarhijski predstavljeni pomoću stabla. Primjer hijerarhijske i nehijerarhijske tehnike je na slici 3.4.



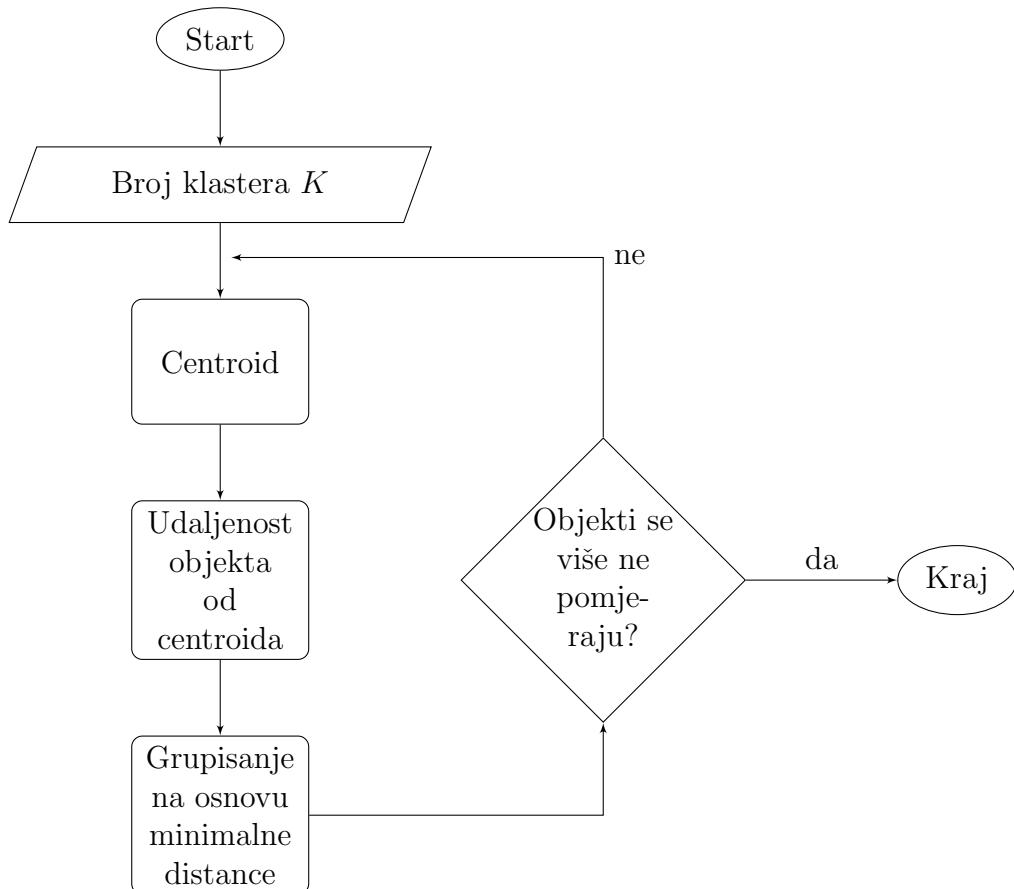
Slika 3.4: Tipovi klasterizacije

Svrha segmentacije je da značajke unutar jedne grupe budu uniformne. Postoje različite metode za ovu vrstu segmentacije, a jedna od najpoznatijih je *K-means* klasterizacija.

3.1.2.1 K-means segmentacija

Ovaj algoritam određuje klastere tako da se minimizira ukupna suma udaljenosti vektora od centroida klastera kojima su ti vektori dodijeljeni. Rezultat ovog algoritma uveliko ovisi od odabране mjere za udaljenost. Postupak *K-means* algoritma je dat na slici 3.5 i sastoji se od sljedećih koraka:

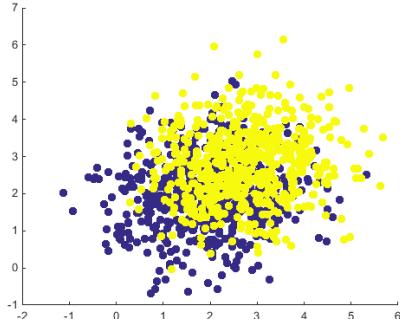
1. Prvo se definiše broj klastera K .
2. Zatim definiše se k centroida (pikseli) za svaki klaster. Ove centroide je potrebno postaviti na pravo mjesto jer od njih zavisi i krajnji rezultat. Stoga ih je bolje postaviti što dalje jedni od drugih.
3. Nakon toga se uzima svaka tačka i računa se njena udaljenost od centroida. Bira se onaj centroid s kojim ima minimalnu udaljenost i pridružuje se tom klasteru. S ovim je završena prva iteracija i napravljeno je prvobitno grupisanje.
4. Nakon toga se ponovo računa k centroida koji se postavljaju u nove položaje zavisno od rezultata prethodnog koraka i algoritam se ponavlja.
5. Kao rezultat ove petlje se može primijetiti da u svakom koraku centroidi mijenjaju svoju poziciju sve dok više ne bude bilo promjena. Drugim riječima došlo je do zasićenja i centroidi se više ne pomjeraju. Time je ovaj algoritam završen i izvršena je klasterizacija.



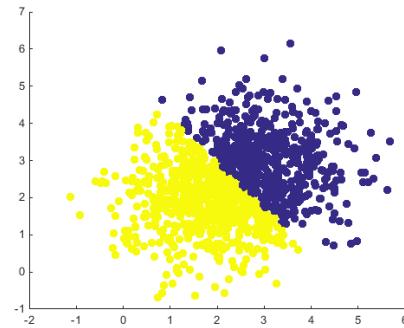
Slika 3.5: Procedura izvršavanja K -means algoritma

Neka su data dva vektora podataka specifičnih za jednu sliku i prikazani na slici 3.6. Potom je pomoću K-means algoritma sprovedena klasterizacija, gdje su ovi vektori razdvojeni u dvije grupe. Rezultat je prikazan na slici 3.7. Sa slike se vidi da je ovaj algoritam uspješno izvršio podjelu ulaznih podataka na dvije klase. Neophodni podaci za izvršavanje ovog algoritma su ulazni podaci i broj klastera na koje će se ulazni podaci podijeliti. Međutim, problem ovog algoritma je što može upasti u lokalne minimume. Jedan od načina da se

ovo izbjegne je da se algoritam pokrene više puta sa različitim početnim uslovima i da se na kraju odabere rezultat koji ima minimalnu distorziju. Drugi mogući problem je što se unaprijed mora znati broj klastera što u realnosti je često nepoznat podatak. Iako postoji algoritmi za procjenu ovog broja klastera još uvijek se oni najčešće određuju na osnovu znanja o konkretnom domenu primjene.



Slika 3.6: Ulagni podaci sa slike



Slika 3.7: Segmentacija K-means algoritmom

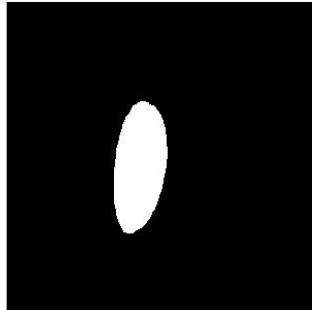
Ova metoda je primjenjena na bazu slika badema kako bi se izdvojio badem i podloga. Za dalje analize slike ovaj korak je jako važan, stoga i rezultati moraju biti adekvatni. Na slici 3.8 je prikazan originalna slika badema nad kojom je potrebno izvršiti segmentaciju.



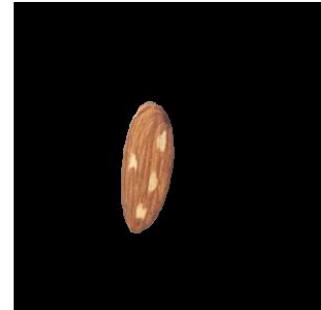
Slika 3.8: Originalni badem

Segmentacija je provedena *Kmeans fast color* algoritmom u $L^*a^*b^*$ prostoru boja. Za alternativu je ostavljena mogućnost segmentacije i u *RGB* prostoru boja navođenjem adekvatnih parametara. Međutim, eksperimentalno se pokazalo da algoritam postiže bolje rezultate u $L^*a^*b^*$ prostoru boja. Ovaj algoritam predstavlja modifikaciju originalnog algoritma jer vrši prealokaciju i obavlja više operacija paralelno kako bi ostvario što bolje performanse za što kraće vrijeme. Obzirom da je potrebno vršiti klasterizaciju u dvije grupe (badem i podloga), za ulazne parametre su postavljena 2 klastera, gdje su klasteri razdvojeni na osnovu boje (a i b komponenta). Na slikama 3.9 i 3.10 se jasno vidi da je ovaj algoritam dao jako dobre rezultate. Prva slika je pogodna za ekstrakciju značajki oblika, dok je druga slika pogodna za ekstrakciju boje. Ono što je još bitno uraditi u ovom koraku je invarijantnost na rotaciju. Obzirom da se bademi na slikama nalaze u različitim položajima, potrebno ih je sve postaviti u jedan isti položaj, radi adekvatne ekstrakcije značajki oblika poput dužine, širine, omjera i sl.

Sličan postupak je proveden i u programskom paketu OpenCV i rezultati segmentacije nad istim slikama su identični kao i u MATLAB-u.



Slika 3.9: Segmentirana slika



Slika 3.10: Objekt od interesa

3.1.3 Segmentacija na bazi diskontinuiteta

Segmentacija na bazi diskontinuiteta (engl. *Edge-based segmentation*) se zasniva na pretpostavci da su granice regiona dovoljno različite jedna od druge, kao i da su granice različite od pozadine, pa je moguće ustanoviti granice na osnovu diskontinuiteta u intenzitetu. Ivice se definiraju kao nagle promjene u intenzitetu susjednih piksela i u suštini čine granice objekata na slici. Postoji mnogo operatora za detekciju ivica, a neki od njih su:

- Robertov operator
- Laplaceov operator
- Prewittov operator
- Sobelov operator
- Robinsonov operator
- Kirschov operator
- Marr-Hildreth detekcija ivica
- Canny detekcija ivica [2]

3.1.4 Segmentacija na bazi regije

Sve prethodne metode su zasnovane na detekciji oblika na osnovu ivica, dok se segmentacija na bazi regije zasniva na detekciji oblasti objekta direktno. Ova metoda daje puno bolje rezultate od prethodnih metoda u slučajevima kada je na slici prisutan šum i veoma je otežana detekcija ivica oblika. Homogenost je važna osobina oblasti i osnovni je kriterij u ovoj metodi segmentacije. Ideja je podijeliti sliku u zone maksimalne homogenosti. Uobičajena su tri pristupa kod ove metode:

- spajanje regija (engl. *region merging*) - kreće se od slike podijeljene na mnogo oblasti, nakon čega se vrši spajanje određenih oblasti pod određenim uslovima,
- razdvajanje regija (engl. *region splitting*) - kreće se od slike podijeljene na mali broj oblasti, koje je potrebno dalje podijeliti po određenom kriteriju,
- podijeli i spoji (engl. *split-and-merge*) - kombinuje prednosti prethodnih pristupa.[2]

Poglavlje 4

Baza slika

U svrhu provođenja klasifikacije hrane, napravljena je baza slika različitih vrsta badema. Sve slike su slikane mobilnim uređajem pod istim uvjetima: pozicija kamere, pozadina i osvjetljenje. Baza se sastoji od 110 slika na kojima se nalaze različite vrste badema. Slike su podijeljene u pet osnovnih klasa:

- sirovi bademi,
- blanširani bademi,
- pečeni bademi,
- blanširani-pečeni bademi,
- ostalo.

Za svaku od navedenih klasa postoji dodatna podjela da li je u pitanju:

- čitav badem,
- slomljen badem,
- oštećen badem.

Svakom bademu se mogu pridružiti dva obilježja, kojoj klasi pripada (sirovi, pečeni, blanširan, blanširan-pečeni, ili nepoznato), a potom i dodatna osobina da li je čitav, slomljen ili oštećen. U klasi ostalo se nalazi ona hrana koja ne pripada niti jednoj navedenoj klasi, za primjer su uzeti lješnjaci. Također, ovoj klasi nije potrebno dodjeljivati dodatne osobine (slomljen, čitav i oštećen).

4.1 Klasa I

Ova klasa se sastoji od 30 slika. U prvih 10 slika se nalaze **sirovi čitavi bademi**, u narednih 10 slika su **sirovi slomljeni bademi**, dok su u posljednjih 10 slika **sirovi oštećeni bademi**. Reprezentativne slike klase I su prikazane na narednim slikama.



Slika 4.1: Sirovi čitavi bademi



Slika 4.2: Sirovi slomljeni badem



Slika 4.3: Sirovi oštećeni badem

4.2 Klasa II

Ova klasa se sastoji od 20 slika na kojim su prikazani blanširani bademi. U prvih 10 slika se nalaze **blanširani čitavi bademi**, a u narednih 10 slika su **blanširani slomljeni bademi**. Reprezentativne slike klase II su prikazane na narednim slikama.



Slika 4.4: Blanširani čitavi badem



Slika 4.5: Blanširani slomljeni badem

4.3 Klasa III

Ova klasa se sastoji od 30 slika na kojim su prikazani pečeni bademi. U prvih 10 slika se nalaze **pečeni čitavi bademi**, u narednih 10 slika su **pečeni slomljeni bademi** i u posljednjih 10 slika se nalaze **pečeni oštećeni bademi**. Reprezentativne slike klase III su prikazane na slikama 4.6-4.8.

4.4 Klasa IV

Ova klasa se sastoji od 20 slika na kojim su prikazani blanširani-pečeni bademi. U prvih 10 slika se nalaze **blanširani-pečeni čitavi bademi**, u narednih 10 slika su **blanširani-**



Slika 4.6: Pečeni čitavi badem



Slika 4.7: Pečeni slomljen badem



Slika 4.8: Pečeni oštećen badem

pečeni slomljeni bademi. Reprezentativne slike klase IV su prikazane na slikama 4.9 i 4.10.



Slika 4.9: Pečeni-blanširani čitavi badem



Slika 4.10: Pečeni-blanširani slomljeni badem

4.5 Klasa V

U ovoj klasi se nalazi ona vrsta hrane koja se ne može svrstati niti u jednu od prethodno navedenih klasa. Kao primjer nepoznate klase su uzeti lješnjaci. Klasa se sastoji od 10 različitih slika. Ovoj klasi nije potrebno pridruživati dodatno obilježje (slomljen, čitav, oštećen) iz razloga što nije bitno sa stanovišta klasifikacije jer ova klasa predstavlja škart. Reprezentativne slike klase V su prikazane na narednim slikama.



Slika 4.11: Nepoznata klasa - lješnjak



Slika 4.12: Nepoznata klasa- lješnjak

Sve slike su pohranjene u jedinstvenu bazu, koja se koristi za klasifikaciju. Radi lakšeg prepoznavanja klase i osobina, slike su numerisane tako da prvi broj predstavlja klasu badema, drugi predstavlja osobinu dok je treći broj predstavlja redni broj slike.

Prvi broj u numeraciji može predstavljati:

- broj 1 - sirove bademe,
- broj 2 - blansirane bademe,
- broj 3 - pečene bademe,
- broj 4 - blansirne-pečene bademe,
- broj 5 - nepoznate bademe.

Drugi broj u numeraciji može predstavljati:

- broj 1 - čitave bademe,
- broj 2 - slomljene bademe,
- broj 3 - oštećene bademe,

Na primjer, slika 3-2-5.jpg predstavlja petu sliku unutar klase pečenih slomljenih badema. Tabelarna organizacija baze je predstavljena tabelom 4.1.

klasa/stanje	čitav	slomljen	oštećen
Klasa I (sirovi)	10	10	10
Klasa II (blansirani)	10	10	
Klasa III (pečeni)	10	10	10
Klasa IV (blansirani-pečeni)	10	10	
Klasa V (nepoznato)	10		

Tabela 4.1: Broj slika po klasama u ovisnosti od stanja badema

Ukoliko bi se bademi predstavljali po različitim vrstama tada bi postojalo ukupno 11 klasa. U tabeli 4.2 je prikazano svih 11 klasa.

Klasa	Broj slika
Klasa 1 (sirovi čitav)	10
Klasa 2(sirovi slomljén)	10
Klasa 3 (sirovi oštećen)	10
Klasa 4 (blanširani čitav)	10
Klasa 5 (blanširani slomljén)	10
Klasa 6 (pečen čitav)	10
Klasa 7 (pečen slomljén)	10
Klasa 8 (pečen oštećen)	10
Klasa 9 (blanširani-pečeni čitav)	10
Klasa 10 (blanširani-pečeni slomljén)	10
Klasa 11 (nepoznato)	10

Tabela 4.2: Broj slika po klasama

Poglavlje 5

Ekstrakcija i selekcija značajki

U brojnim naučnim disciplinama i aplikacijama postoji velika količina podataka koji se trebaju analizirati kako bi se otkrilo koji od tih podataka su bitni za dalju analizu. Važan korak u pripremi podataka za proces klasifikacije i prepoznavanja oblika je selekcija osobina i redukcija dimenzionalnosti. Nakon preprocesiranja, slika može sadržavati veliku količinu informacija koji ne doprinose razlikovanju oblika ili sadrži redundantne zapise koji ne daju nikakvu bitnu informaciju u procesu učenju iz podataka. Redukcijom dimenzionalnosti se takve osobine iz skupa podataka uklanjaju i olakšava se kreiranje sistema za klasifikaciju.

Problem dimenzionalnosti se može prevladati:

- tako da se odabere samo podskup relevantnih značajki,
- stvaranjem novih značajki koji sadrže najviše informacija o objektu.

Prva metodologija se zove selekcija značajki, a druga se zove ekstrakcija značajki. Ekstrakciju značajki treba razlikovati od selekcije značajki jer selekcija osobina kao rezultat vraća podskup originalnih osobina, a ekstrakcija osobina kreira nove atribute na osnovu originalnog skupa značajki i pri tome reducira ukupni broj značajki.

Selekcija značajki iz postojećeg seta bira onaj podskup značajki koji su relevantni za ciljani problem istraživanja.

Prije same ekstrakcije i selekcije na osnovu slika su izdvojene značajke oblika i boje. Rezultati dobijeni predstavljaju skup atributa za dalju klasifikaciju.

5.1 Značajke oblika

Vizualne značajke su jako bitne za raspoznavanje slika. Za raspoznavanje ispravnosti hrane važnu ulogu ima njen oblik, jer i sami kupci validnost hrane prvo procjenjuju na osnovu njenog oblika. Hrana u industriji se vrlo lako može ošteti i slomiti, stoga je važno identificirati njen oblik i na osnovu njega donijeti relevantne zaključke da li se hrana treba ukloniti iz dalje upotrebe.

Efikasne značajke oblika se odlikuju sljedećim osobinama [1]:

- prepoznatljivost: oblici koji su perceptualno slični čovjeku imaju iste značajke koje se razlikuju od drugih,
- invarijantnost na translaciju, rotaciju i skaliranje: promjena lokacije, rotacije i skaliranja ne smije uticati na izdvojene značajke,
- invarijantnost na afinu transformaciju: ova transformacija izvršava linearno mapiranje iz 2D koordinata u druge 2D koordinate koje održavaju pravac i paralelnost linija. Izdvojene značajke moraju biti invarijantne na ove transformacije,
- otpornost na šum,
- invarijantnost na prikrivanje (engl. *occultation*): kada su neki djelovi objekta prekriveni nekim drugim objektom, značajke preostalog dijela objekta se ne smiju promjeniti u poređenju sa originalnim oblikom,
- statistička nezavisnost: dvije značajke moraju biti statistički neovisne,
- pouzdanost: sve dok je u pitanju isti uzorak, izdvojene značajke moraju ostati iste.

Za oblik badema izdvojene su sljedeće značajke opisane u nastavku i prikazane u tabeli 5.1.

Red. br.	Naziv značajke
1	Ekscentricitet
2	Dužina velike ose
3	Kružnost
4	Obim
5	Dužina
6	Širina
7	Omjer dužine i širine
8	Cirkularnost
9	Površina
10	Poluprečnik
11	Komponenta R boje
12	Komponenta G boje
13	Komponenta B boje
14	Komponenta R boje (originalne slike)
15	Komponenta G boje (originalne slike)
16	Komponenta B boje (originalne slike)

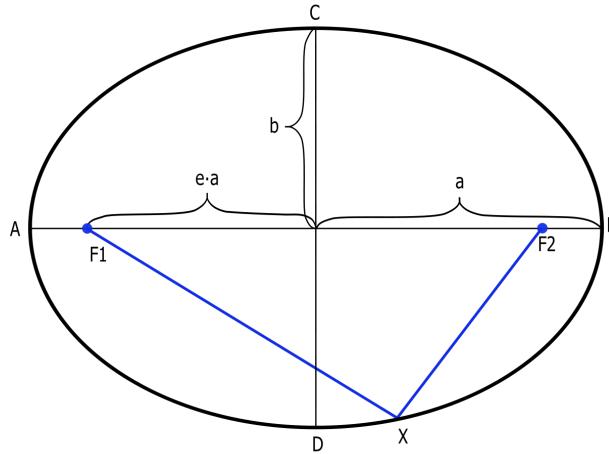
Tabela 5.1: Izdvojene značajke oblika i boje za bademe

5.1.1 Ekscentricitet

Ekscentricitet predstavlja omjer dužine velike ose i dužine male ose. Računa se prema relaciji 4.1.

$$e = \sqrt{a^2 - b^2}, a > b, \quad (5.1)$$

gdje a i b predstavljaju veliku i malu osu respektivno kao što je to prikazano na slici 5.1.

**Slika 5.1:** Osnovne oznake elipse

Može se izračunati na dva načina: koristeći metodu glavnih osa (engl. *principal axes method*) ili metodu minimuma ograničavanja pravougaonika (engl. *minimum bounding rectangle method*).

Sa aspekta osobina oblika badema ova značajka ima značajnu ulogu jer sam oblik badema predstavlja elipsu. Stoga računanje njenog ekscentriteta je bitno za dalju analizu.

5.1.2 Dužina velike ose

Dužina velike ose predstavlja najduži prečnik elipse. Računa se prema sljedećoj relaciji:

$$M = a + b, \quad (5.2)$$

gdje a i b predstavlja veliku i malu osu respektivno.

5.1.3 Površina i obim

Pod površinom se podrazumijeva ukupan broj piksela koji obuhvata promatrani oblik. Ovaj parametar ima važnu ulogu u raspoznavanju oblika badema, jer se na osnovu površine može lako zaključiti da li se radi o čitavom ili slomljenom bademu. Označava se sa P .

Obim daje informacije o ukupnom broju piksela koji čine obim promatranog oblika. Označava se sa O . Na osnovu ove informacije može se zaključiti kakav oblik ima promatrani badem (slomljen, čitav, oštećen).

5.1.4 Cirkularnost

Pod ovim pojmom se podrazumijeva mjera koliko promatrani oblik ima sličnosti sa kružnjem. Cirkularnost predstavlja omjer površine promatranog oblika i površine kruga koji ima isti obim kao i promatrani oblik, ovo je predstavljeno relacijom 5.3.

$$C = \frac{A_s}{A_c}, \quad (5.3)$$

gdje A_s predstavlja površinu promatranog oblika, a A_c predstavlja površinu kruga koji ima isti obim kao i promatrani oblik. Ukoliko se obim označi kao O tada je $A_c = \frac{O^2}{4\pi}$. Uvrštavanjem u relaciju 5.3 i koristeći činjenicu da 4π predstavlja konstantu dobije se relacija za cirkularnost predstavljena relacijom 5.4.

$$C = \frac{P}{O^2}, \quad (5.4)$$

gdje P predstavlja površinu promatranog oblika, a O obim oblika. Prema relaciji 5.4 cirkularnost predstavlja omjer površine oblika i kvadrata obima oblika. S obzirom da bademi imaju izduženi oblik kruga ova značajka može pomoći pri razlikovanju različitih stanja badema.

Cirkularnost se također naziva i kružna varijansa (engl. *circle variance*) i definira se kao:

$$C_{va} = \frac{\sigma_R}{\mu_R} \quad (5.5)$$

gdje σ_R i μ_R predstavljaju srednju i standardnu devijaciju radijalne udaljenosti od centroida (g_x, g_y) oblika do njegovih krajnjih tačaka (x_i, y_i). Računaju se prema sljedećim relacijama:

$$\mu_R = \frac{1}{N} \sum_{i=1}^{N-1} d_i \quad (5.6)$$

$$\sigma_R = \sqrt{\frac{1}{N} \sum_{i=1}^{N-1} (d_i - \mu_R)^2} \quad (5.7)$$

Kružna cirkularnost je prikazana na slici 5.2.



Slika 5.2: Kružna cirkularnost [1]

5.1.5 Poluprečnik

Poluprečnik promatranog oblika se računa prema sljedećoj relaciji:

$$R = \sqrt{\frac{P}{\pi}} \quad (5.8)$$

5.1.6 Kružnost

Kružnost predstavlja mjeru koliko je objekat okrugao, odnosno koliko promatrani oblik odudara da bude savršen krug. Računa se prema sljedećoj relaciji.

$$M = \frac{4\pi P}{O^2} \quad (5.9)$$

gdje P predstavlja površinu, a O obim promatranog oblika. Što je rezultat metrike bliži jedinici to indicira da je oblik sličniji krugu. Ovaj parametar je povezan sa cirkularnosti.

5.1.7 Omjer dužine i širine badema

Na osnovu segmentirane slike moguće je također dobiti informacije o dužini i širini badema. S obzirom da se bademi mogu naći na različitim pozicijama na slici (neki su postavljeni horizontalno, neki vertikalno i sl.), potrebno je prvo sve bademe rotirati na istu poziciju. Nakon što je to urađeno izdvojene su informacije o dužini i širini. Također na osnovu ovih podataka je izračunat omjer prema sljedećoj relaciji.

$$\text{omjer} = \frac{\text{duzina}}{\text{širina}} \quad (5.10)$$

5.2 Značajke boje

Boje su specijalno sredstvo za emocionalno i umjetničko izražavanje grafičkih sadržaja. Čovjek koristi opseg svjetlosnog zračenja sa talasnom dužinom od 380 [nm] do 700 [nm]. Ta oblast se naziva vidljivom svjetlošću elektromagnetskog spektra. Različite talasne dužine elektromagnetskog spektra definišu svjetlosno zračenje koje čovjek vidi kao boje.

Boja nekog objekta ne ovisi isključivo o samom objektu nego i o izvoru svjetlosti, boji okoline i čovjekovu sistemu vida. Neki objekti odbijaju svjetlost, a neki je propuštaju. Tako na primjer površina koja odbija samo plavu svjetlost, obasjana crvenom svjetlošću izgleda kao crna.

U klasifikaciji slika boja može predstavljati važnu značajku koja pomaže pri razlikovanju objekata. Efikasnost značajke boje se ogleda u činjenici da je ova značajka neovisna i neosjetljiva na promjenu veličine oblika, invarijantna je na rotaciju, translaciju i skaliranje. Postoji više različitih modela boja koje se koristi u praksi, koji su obrađeni u nastavku.

5.2.1 Modeli boja

Ljudsko oko vidi boju kao različitu kombinaciju primarnih boja; crvene, zelene i plave. Na osnovu ovoga je utvrđen standardni model apsorpcije svjetlosti u ljudskom oku. Primarne boje svjetlosti su:

- crvena (engl. *red*),
- zelena (engl. *green*),

- plava (engl. *blue*):

Miješanjem po dvije primarne boje svjetlosti se dobivaju sekundarne boje svjetlosti:

- ružičasta (engl. *magenta*) crvena+plava,
- cijan (engl. *cyan*) zelena+plava,
- žuta (engl. *yellow*) crvena+ zelena,

Karakteristike po kojima se razlikuju boje su:

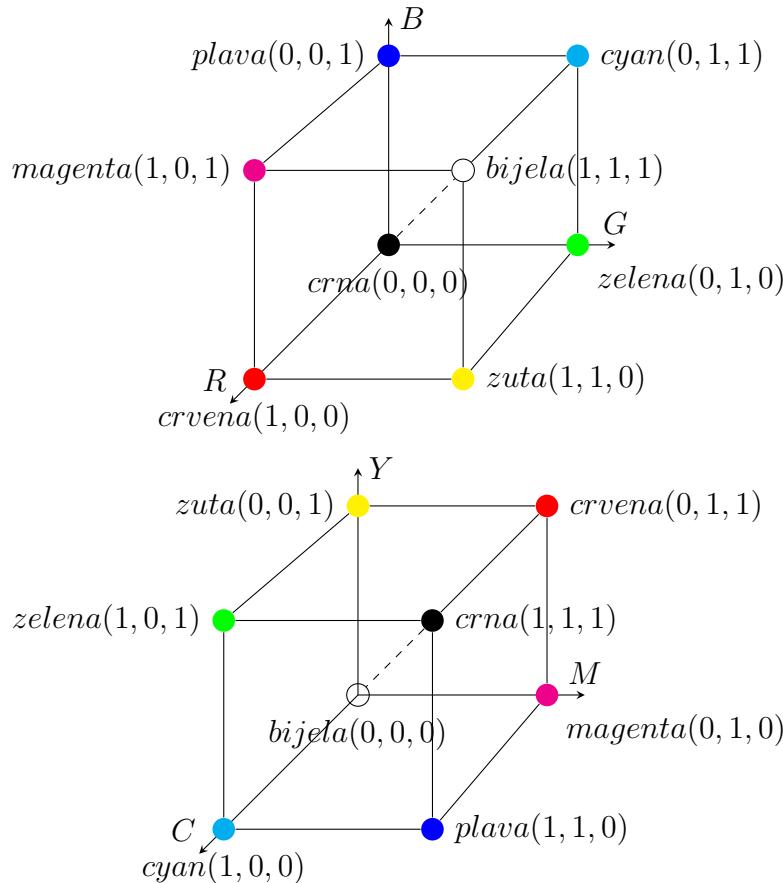
- Svjetlost (engl. *brightness*) - predstavlja intenzitet,
- nijansa boje (engl. *hue*) - predstavlja dominantnu talasnu dužinu u spektru svih talasnih dužina,
- zasićenje (engl. *saturation*) - relativna čistoća boje, odnosno količina bijele boje pomiješana sa *hue*.

Namjena modela boja je da na standardizovan način omogući predstavljane boje. Model boja je specifikacija koordinatnog sistema i podprostora toga sistema u kome je svaka boja predstavljena jednom tačkom [19].

Najviše korišteni modeli boja su:

- RGB (**R**ed, **G**reen, **B**lue) - koristi se u računarskoj grafici,
- CMYK (**C**yan, **M**agenta, **Y**ellow, **K**black) ili CMY - koristi se kod printera u boji,
- HSI (**H**ue, **S**aturation, **I**ntensity) - najbliži ljudskom oku jer razdvaja boju od intenziteta.

RGB model definisan je u Kartezijevom koordinatnom sistemu. Primarne veličine, crvena, zelena i plava boja, nazivaju se aditivnim primarnim veličinama jer se sabiraju da bi se dobila specificirana boja. Područje u kojem su boje definirane je oblika jedinične kocke. RGB model boje prikazan je na slici 5.3. U centru koordinatnog sistema je smještena crna boja, dok je suprotno od nje smještena bijela boja. Prostorna dijagonala koja ih povezuje predstavlja nivoe sive boje. Kombinacijom primarnih boja se dobijaju sekundarne boje, što je prikazano na slici 5.3.



Slika 5.3: RGB i CMY model boja [2]

Suprotan model aditivnom je **substraktivni** koji boju definiše oduzimanjem zadate boje od bijele boje [2]. Substraktivni model je primjenjen u štampi boja. Najpoznatiji substraktivni model je CMY model koji se formira od cijan, magente i žute boje. Na slici 5.3 je prikazana jedinična kocka modela CMY.

Moguće je preći iz jednog modela boja u drugi pomoću odgovarajućih formula. Ukoliko se prelazi iz RGB modela u CMY uz prepostavku da su vrijednosti normalizovane u opsegu $[0, 1]$ koristi se sljedeća relacija:

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (5.11)$$

Ukoliko je nešto obojeno cijan bojom onda to znači da će crvena komponenta svjetlosti biti apsorbovana a ostale dvije će biti odbijene. Printeri obično koriste CMY ili vrše internu konverziju RGB u CMY. Crna boja se formira miješanjem C, M i Y u jednakim omjerima što je teško i neisplativo. Zbog toga se kod printerova često koristi model boja koji uključuje i crnu boju (CMYK) kao odvojenu komponentu. Razlog za to je činjenica da se većina ispisa radi u crnoj boji.

Pored ova dva modela jedan od najzastupljenijih modela boja je model CIE L*a*b*. Ovaj model boja se zasniva na ljudskoj percepciji boje. On definiše širi opseg boja od ostalih modela. Lab boja se definije preko svjetline (engl. *luminance*) i još dvije komponente (a i b) koje idu od zelene do crvene (komponenta a), odnosno od plave do žute (komponenta b).

b). $L^*a^*b^*$ boja ne zavisi od uređaja, što znači da opseg boja definisan u ovom modelu nije ograničen na opseg boja koje se mogu odštampati ili prikazati na određenom uređaju. Ovaj model boje u ovom radu je korišten za segmentaciju slike pomoću klasterizacijskog algoritma $K - means$.

Za izdvajanje značajke boje badema je korišten RGB model boja. Pored ovog modela vršeni su eksperimenti sa ostalim modelima boje, ali ovaj model boja je pokazao najbolju tačnost.

Proces se zasniva na pronalasku srednjih vrijednosti pojedinih komponenti boja. Za izdvojeni badem nakon procesa segmentacija opisanog u poglavlju 3 i prikazanog na slici 3.10. izdvojena je boja i izračunate su srednje vrijednosti nenuktih piksela slike. Svaka komponenta boje je predstavljena srednjom vrijednosti.

Vrijednost crvene boje R badema prikazanog na slici 3.10 se može izračunati prema sljedećoj relaciji:

$$R = \frac{\sum_{i=0}^{N-1} R(p_i)}{N}, \quad p_i \in \Pi \quad (5.12)$$

gdje p_i predstavlja piksel, dok Π područje obuhvaćeno konturom oblika. R predstavlja srednju vrijednost crvene boje regije od interesa (u ovom slučaju badema), a $R(p_i)$ predstavlja vrijednost komponente crvene boje na poziciji piksela p_i . Analogno ovome se računaju i preostale vrijednosti za plavu (B) i zelenu (G) komponentu prema sljedećim relacijama.

$$G = \frac{\sum_{i=0}^{N-1} G(p_i)}{N}, \quad p_i \in \Pi \quad (5.13)$$

$$B = \frac{\sum_{i=0}^{N-1} B(p_i)}{N}, \quad p_i \in \Pi \quad (5.14)$$

5.3 Selekcija značajki

Prisustvo irelevantnih i redundantnih informacija u formulaciji problema negativno utiče na performanse samog algoritma. Stoga prije same klasifikacije potrebno je izvršiti selekciju značajki. Selekcija značajki se definiše kao proces koji bira minimalni podskup M značajki iz izvornog skupa N značajki, tako da je prostor značajki optimalno smanjen prema određenom kriteriju ocjenjivanja. Kako se dimenzionalnost domena širi, broj značajki N se povećava. Pronalaženje najboljeg podskupa značajki je nerijetko težak problem, pa se često koriste različite metode koje pomažu da se izaberu optimalni atributi. Selekcija značajki je osnovni problem u mnogim područjima a posebno u predikciji, klasifikaciji, bioinformatici, prepoznavanju objekata ili u modeliranju složenih tehnoloških procesa.

Često pri analizi se radi sa velikim skupovima podataka koji imaju i preko hiljadu značajki. Svi ovi atributi su važni za pojedine probleme, ali za neka ciljana istraživanja samo njihov mali podskup je bitan. Procesom selekcija značajki se smanjuje dimenzionalnost podataka, uklanjaju se suvišni i otklanja se šum. Nakon ovog procesa algoritam radi znatno brže, poboljšava se kvalitet podataka i performanse kao i poređenje dobijenih rezultata.

Algoritmi za selekciju značajki mogu se podijeliti na [22]:

- filtere (engl. *filters*) - bira podskup značajki nezavisno od algoritma učenja,
- metode prethodnog učenja (engl. *wrappers*) - vrši selekciju značajki prema estimaciji tačnosti predviđanja koju daje izabrani klasifikator,
- ugrađene (engl. *embedded*) - vrše selekciju značajki u sklopu osnovnog algoritma.

5.4 Ekstrakcija značajki

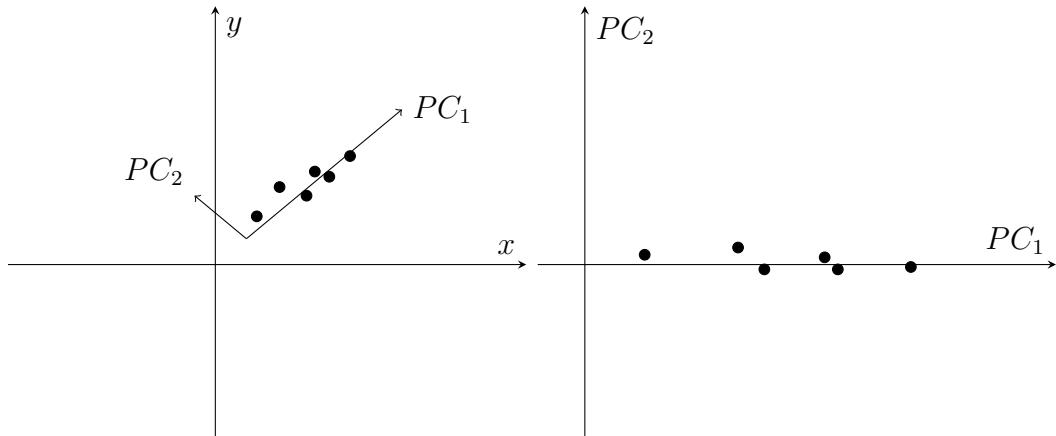
U današnjim raznim aplikacijama skupovi podataka sadrže preko hiljadu atributa. Može se desiti da nekad svi atributi budu značajni za određene probleme, ali za neke ciljane probleme samo mali podskup atributa je obično relevantan. Kako bi se prevazišli ovi problemi vrši se ekstrakcija atributa, gdje se na osnovu nekog kriterija i originalnih atributa, formira podskup novih atributa koji su relevantni za promatrani problem.

5.4.1 *Principal Component Analysis - PCA*

Karl Pearson je 1901. godine izložio u svom radu *On Lines and Planes of Closest Fit to Systems of Points in Space* metodu analize glavnih komponenata, ali je Hotelling dosta kasnije 1933. godine razradio praktične računske metode. Zbog kompleksnog računa, veća primjena ove tehnike, počela je sa razvojem računara. Ovaj pristup se u različitim oblastima primjene javlja pod različitim nazivima kao što su: diskretna Kosambi-Karhunen–Loëve transformacija, Hotelling transformacija, i drugi.

Principal component analysis (PCA) [23] transformiše skup opservacija, potencijalno koreliranih varijabli u skup vrijednosti linearne nekoreliranih varijabli koje se nazivaju glavne komponente (engl. *principal components*). Ovom metodom se postiže redukcija dimenzionalnosti velikog broja podataka, a također koristi se u svrhu postizanja preglednosti i pojednostavljenja velikog broja podataka. Maksimalan broj novih atributa odnosno broj glavnih komponenata koji se mogu formirati jednak je broju izvornih atributa gdje nove varijable nisu međusobno korelirane [3]. Obično broj glavnih komponenta je manji od broja originalnih atributa, pri čemu se postiže redukcija dimenzionalnosti.

Matematski PCA predstavlja ortogonalnu linearnu transformaciju koja transformiše skup podataka u novi koordinatni sistem takav da projekcija podataka sa najvećom varijansom predstavlja prvu osu (prvu glavnu komponentu), narednu projekciju sa najvećom varijansom na narednu ortogonalnu osu itd. (slika 5.4) [19]. Obično varijansa originalnih podataka se može opisati sa nekoliko prvih glavnih komponenata, dok ostatak može biti zanemaren. U ovom slučaju korištenjem nekoliko prvih glavnih komponenta se vrši redukcija dimenzionalnosti.



Slika 5.4: Glavne komponente su ortogonalne. Ukoliko je varijansa PC_2 mala, onda se zanemaruje i dimenzionalnost se redukuje sa dva na jedan [3]

Baza podataka u PCA metodi sastoji se od n mjerena, objekata ili slično (vektori) na kojima se mjeri p obilježja, atributa, značajki, parametara (variables). Ovi se podaci mogu interpretirati kao n tačaka u p -dimenzionalnom vektorskem prostoru R^p i imaju oblik matrice $n \times p$.

Ukoliko su dvije variable visoko korelisane onda se smatra da imaju isti ili sličan sadržaj. Ovom metodom se veći broj takvih varijabli zamjenjuje manjim brojem varijabli. U tu svrhu se vrši transformacija koordinatnog sistema. Projekcije varijabli baze podataka na koordinatne ose novog koordinatnog sistema predstavljaju nove variable – glavne komponente (*principal component - PC*) koje se dobivaju kreiranjem p linearnih kombinacija izvornih varijabli.

$$\epsilon_1 = w_{11}X_1 + w_{12}X_2 + \dots + w_{1p}X_p \quad (5.15)$$

$$\epsilon_2 = w_{21}X_1 + w_{22}X_2 + \dots + w_{2p}X_p \quad (5.16)$$

$$\vdots \quad (5.17)$$

$$\epsilon_p = w_{p1}X_1 + w_{p2}X_2 + \dots + w_{pp}X_p \quad (5.18)$$

gdje $\epsilon_1, \epsilon_2..,\epsilon_p$ predstavlja p glavnih komponenata i w_{ij} su težine odnosno konstante koje čine koeficijente j -te varijable za p -tu komponentu. Konstante w_{ip} se nazivaju svojstveni ili latentni vektori (engl. *eigenvectors*) i u geometrijskom smislu su, u dvodimenzionalnoj strukturi, ustvari, sinusi i kosinusi uglova novih osa, tj. glavnih komponenata. Transformisane vrijednosti izvornih varijabli predstavljaju skorove glavnih komponenata (engl. *principal component scores*).

Svojstvena vrijednost (engl. *eigenvalue*) je zapravo varijansa izračunata iz seta skorova glavne komponente. Ona je najveća u prvoj glavnoj komponenti i u svakoj sljedećoj njeni vrijednost je sve manja. Prva glavna komponenta, ϵ_1 , objašnjava maksimum varijanse iz podataka, druga glavna komponenta, ϵ_2 , objašnjava maksimum varijanse koja je ostala neobjašnjena prvom i tako dalje. Obzirom da je suma svih svojstvenih vrijednosti jednaka ukupnoj varijansi cilj je iteracijskim postupkom, izdvojiti veći dio ukupne varijanse u nekoliko prvih glavnih komponenata i time reducirati broj izvornih varijabli.

Neka je

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1p} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{d3} & \dots & x_{np} \end{bmatrix}$$

matrica izvornih varijabli sa p atributa.

Kovarijansa za dvije varijable matrice X zadana je relacijom:

$$c_{jk} = \frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{n-1}, \quad j, k = 1, 2, \dots, p \quad (5.19)$$

gdje je

$$\bar{x}_j = \frac{\sum_{i=1}^n x_{ij}}{n} \quad (5.20)$$

aritmetička sredina j -te originalne varijable.

Matrica kovarijansi je data sljedećom relacijom:

$$Cov(X) = c_{jk}, \quad j, k = 1, 2, \dots, p \quad (5.21)$$

Neka je E matrica svojstvenih vrijednosti uređena po veličini (najveća je prva):

$$E = (\lambda_1, \lambda_2, \dots, \lambda_p)^T \quad (5.22)$$

Koordinate svojstvenog jediničnog vektora e koji pripada svojstvenoj vrijednosti λ dobivaju se iz homogenog sistema jednačina.

$$|cov(x) - \lambda I|e = 0 \quad (5.23)$$

Koordinate svojstvenih jediničnih vektora su kolone matrice:

$$A = \begin{bmatrix} e_{11} & e_{12} & e_{13} & \dots & e_{1p} \\ e_{21} & e_{22} & e_{23} & \dots & e_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{p1} & e_{p2} & e_{d3} & \dots & e_{pp} \end{bmatrix}$$

Prva kolona je svojstveni vektor koji odgovara prvoj svojstvenoj vrijednosti matrice E , drugi drugoj, itd. Elementi ove matrice se nazivaju zasićenja (engl. *loadings*). Traženu novu bazu čine svojstveni vektori, redom kako su složeni u matrici A . Matrica

$$Y = AX^T \quad (5.24)$$

je matrica vektora u transformisanom koordinatnom sistemu.

Udio varijanse glavnih komponenta u ukupnoj varijansi je kriterij koji određuje koliko će se prvih komponenata koristiti, a da se pri tome ne izgube značajne informacije. Na ovaj način se iz velikog broja originalnih atributa kreira nekoliko glavnih komponenta koje nose većinu informacija i čine glavni oblik. Tako na primjer ukoliko je oko 98% varijanse sadržano u prve tri glavne komponente za analizu je dovoljno koristiti samo njih dok se ostale zanemaruju.

Međutim ima situacija kada originalni atributi nisu međusobno korelisani, tada analiza ne daje povoljne rezultate, odnosno ne postiže se redukcija dimenzionalnosti jer su svi atributi značajni za razmatrani problem. U slučaju kada su izvorni atributi visoko pozitivno ili negativno korelisani postižu se najbolji rezultati. Tada se može očekivati da će na primjer 20-30 atributa biti obuhvaćeno sa samo 2 ili 3 glavne komponente.

Pri formiranju novog koordinatnog sistema on zadovoljava sljedeće uslove:

- prva glavna komponenta ima maksimalno moguću vrijednost varijanse, druga ima maksimalnu moguću vrijednost od ostatka ukupne varijanse, analogno ovome tako i treća do zadnje p-te glavne komponente,
- zbir varijansi originalnih varijabli jednak je zbiru varijansi glavnih komponenata,
- glavne komponente su međusobno ortogonalne i nekorelisane.

Osnovni koraci u analizi glavnih komponenta su:

- Potrebno je standardizirati varijable tako da im je srednja vrijednost 0, a varijansa 1 kako bi sve bile na jednakom nivou u analizi, jer je većina setova podataka konstruisana iz varijabli različitih skala i jedinica mjerjenja,
- izračunati matrice korelacije između svih izvornih standardizovanih varijabli,
- pronaći svojstvene vrijednosti glavnih komponenta,
- na kraju, odbaciti one komponente koje nose malo udjela varijanse (obično prvih nekoliko nose 80% - 90% ukupne varijanse).

Za interpretaciju glavnih komponenata osnovu čine svojstveni vektori. Njihove vrijednosti u prvoj glavnoj komponenti su ravnomjerno raspoređene po svim izvornim varijablama, dok u drugoj dolazi do veće disproporcije, što omogućava izdvajanje izvornih varijabli sa većom važnošću i pomaže u objašnjavanju i sažimanju ukupne varijabilnosti.

5.4.2 Analiza značajki badema pomoću metode glavnih komponenata

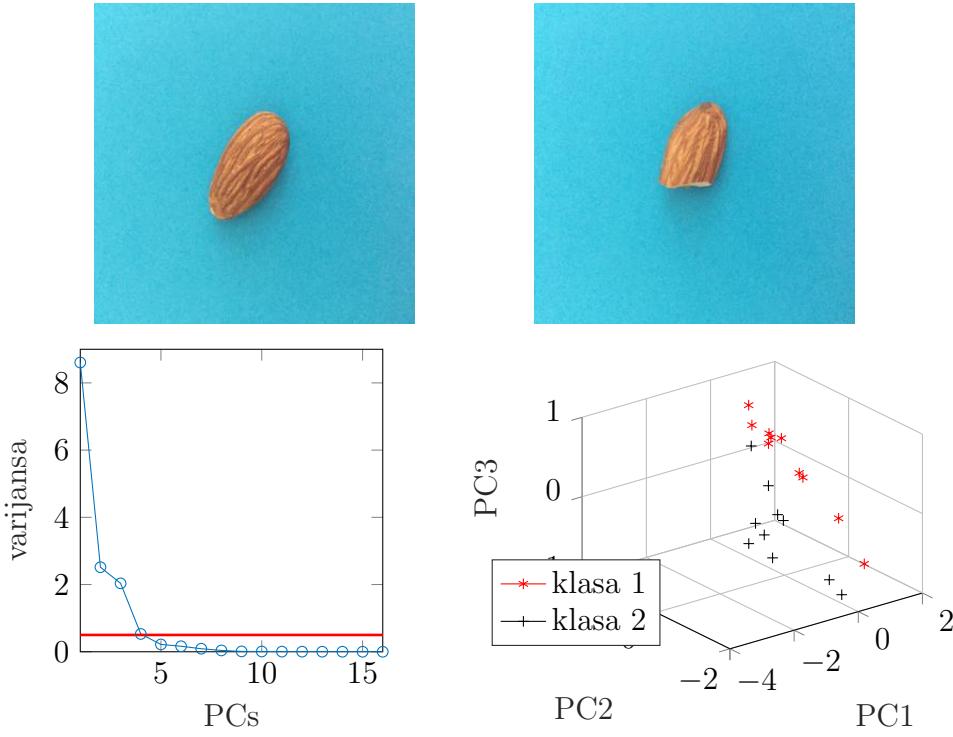
Prije same klasifikacije u okviru ispitivanja ispravnosti različitih vrsta i stanja badema provedena je analiza izdvojenih značajki boje i oblika za cjelokupnu postojeću bazu badema.

Izdvojeno je ukupno 16 značajki prikazanih u tabeli 5.1 za svaku sliku badema (ukupno njih 110). Iz skupa ovih značajki potrebno je odrediti koje značajke su relevantne za raspoznavanje pojedinih klasa badema.

5.4.2.1 Usporedba i analiza klase 1 i 2

Klase 1 se sastoji od čitavih sirovih badema ukupno njih 10, dok su u klasi 2 slomljeni sirovi bademi. Primjenom metode PCA dobiju se glavne komponente tzv. matrica zasićenja dimenzija 16×16 i matrica skorova glavnih komponenata dimenzija 20×16 jer postoji ukupno 20 slika iz klase 1 i 2 i ukupno 16 atributa. Rekonstrukcija originalnih podataka se može izvršiti jednostavnim množenjem ovih dviju varijabli.

Analizom varijanse predstavljene na slici 5.5 može zaključiti da prve tri glavne komponente nose najviše informacija. Tačnije prve tri glavne komponente sadrže ukupno 92.4779% varijanse. Na osnovu ovog podatka može se zaključiti da prve tri komponente igraju glavnu ulogu pri analizi dok se sve preostale mogu zanemariti.



Relevantne značajke su: komponente R, dužina velike ose, površina i kružnost.

Slika 5.5: Usporedba i analiza klase 1 i 2

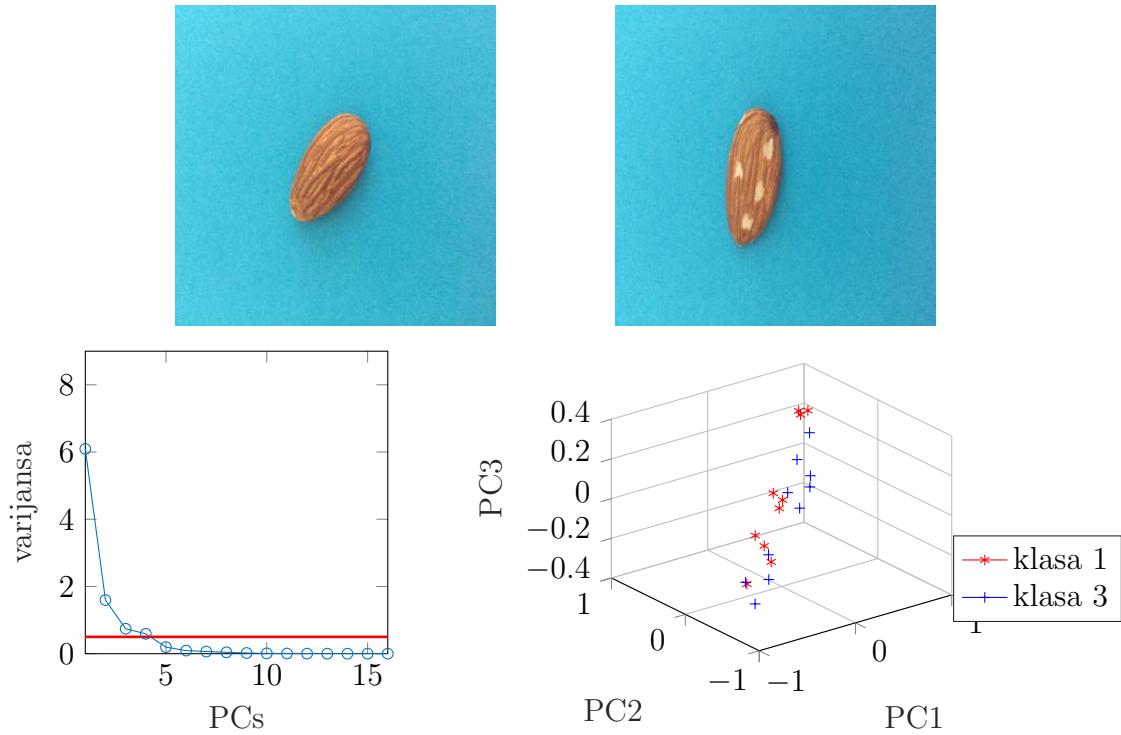
Analizom rezultata nakon zanemarivanja irrelevantnih informacija se može zaključiti da najbolje značajke koje pomažu pri razlikovanju klase 1 i 2 se ostvaruju značajkama: komponenta R, dužina velike ose, površina, obim, omjer, kružnost i ekscentricitet. Ovi rezultati su i u skladu sa intuitivnim razmišljanjem jer se čitavi i slomljeni bademi mogu najbolje razlikovati po značajkama oblika.

Grafički prikaz ove dvije klase nakon zanemarivanja irrelevantnih značajki je prikazan na slici 5.5. Sa slike se vidi da su značajke razdvojene dovoljno da se između njih može provući ravan tako da se sa jedne strane ravni nalaze instance jedne klase, dok su sa druge strane ravni instance klase 2. Ovakva raspodjela je pogodna za primjenu metode mašina vektorske podrške - SVM.

5.4.2.2 Usporedba i analiza klase 1 i 3

Analogno prethodnom izlaganju, izvršena je analiza klase 1 i 3 odnosno čitavih sirovih badema i oštećenih sirovih badema. Na slici 5.6 je prikazana varijansa, gdje se može zaključiti da su samo prve četiri glavne komponente bitne za dalju analizu dok se ostale zanemaruju. Odnosno, u prve četiri komponente sadržano je 89.2198% ukupne varijanse.

Grafički prikaz nakon zanemarivanja irelevantnih značajki je prikazan na slici 5.6. Relevantne značajke za razlikovanje ove dvije klase su komponente R, G i B.

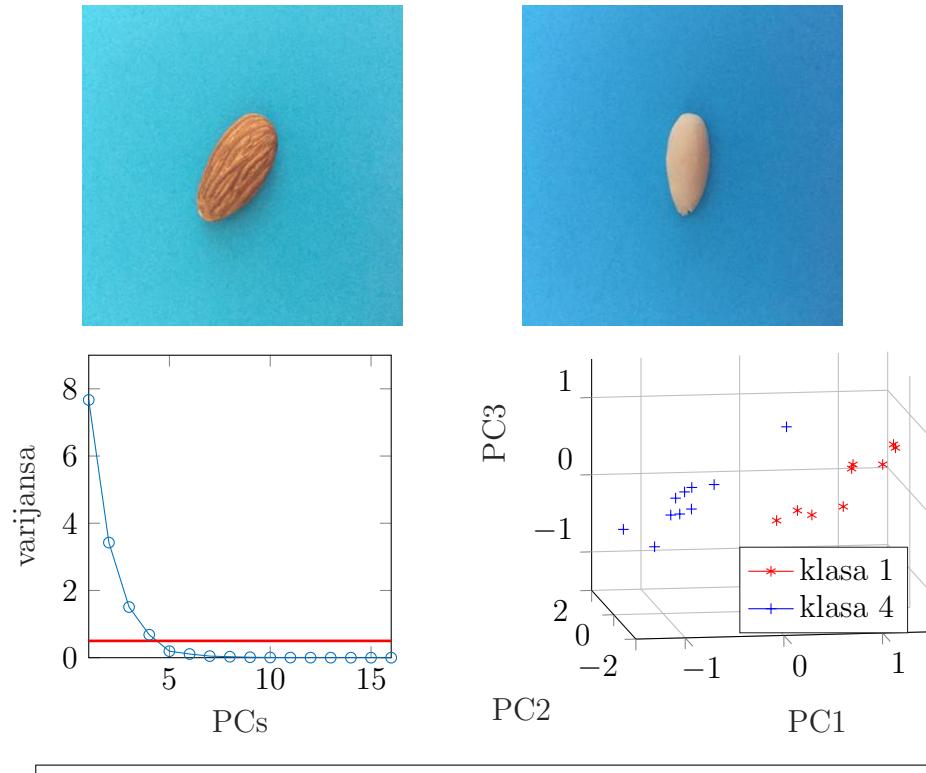


Slika 5.6: Usporedba i analiza klase 1 i 3

5.4.2.3 Usporedba i analiza klase 1 i 4

Analogno prethodnom izlaganju, izvršena je analiza klase 1 i 4 odnosno čitavih sirovih badema i čitavih blanširanih badema. Na slici 5.7 je prikazana varijansa, gdje se može zaključiti da su samo prve četiri glavne komponente bitne za dalju analizu dok se ostale zanemaruju. Odnosno u prve četiri komponente sadržano je 97.0348% ukupne varijanse.

Grafički prikaz nakon zanemarivanja irelevantnih značajki je prikazan na slici 5.7. Instance pojedinih klasa su grupisane u trodimenzionalnom prostoru i može se provući takva ravan da se instance klase nalaze sa suprotne strane ravni. U skladu sa analizom PCA najpogodnije značajke koje je potrebno koristiti za razlikovanje instanci ovih klasa su komponente R, G i B boje i kružnost. Naravno, s obzirom da su sirovi bademi smeđe boje, a blanširani bijele može se i intuitivno zaključiti da je boja najpogodnija značajka za razlikovanje ove dvije klase. Ovo je potvrđeno i pomoću PCA metode.



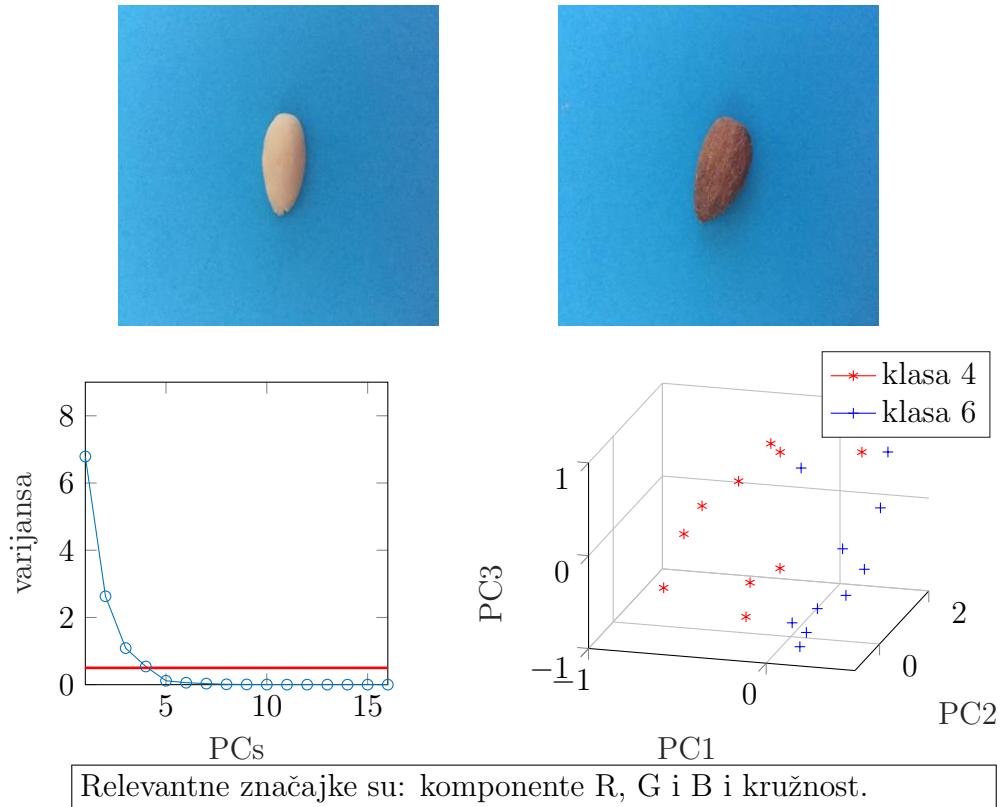
Relevantne značajke su: komponente R, G i B i kružnost.

Slika 5.7: Usporedba i analiza klase 1 i 4

5.4.2.4 Usporedba i analiza klase 4 i 6

Analogno prethodnom izlaganju, izvršena je analiza klase 4 i 6 odnosno čitavih blanširanih badema i pečenih čitavih badema. Na slici 5.8 je prikazana varijansa, gdje se može zaključiti da su samo prve četiri glavne komponente bitne za dalju analizu dok se ostale zanemaruju. Odnosno u prve četiri komponente sadržano je 97.9832% ukupne varijanse.

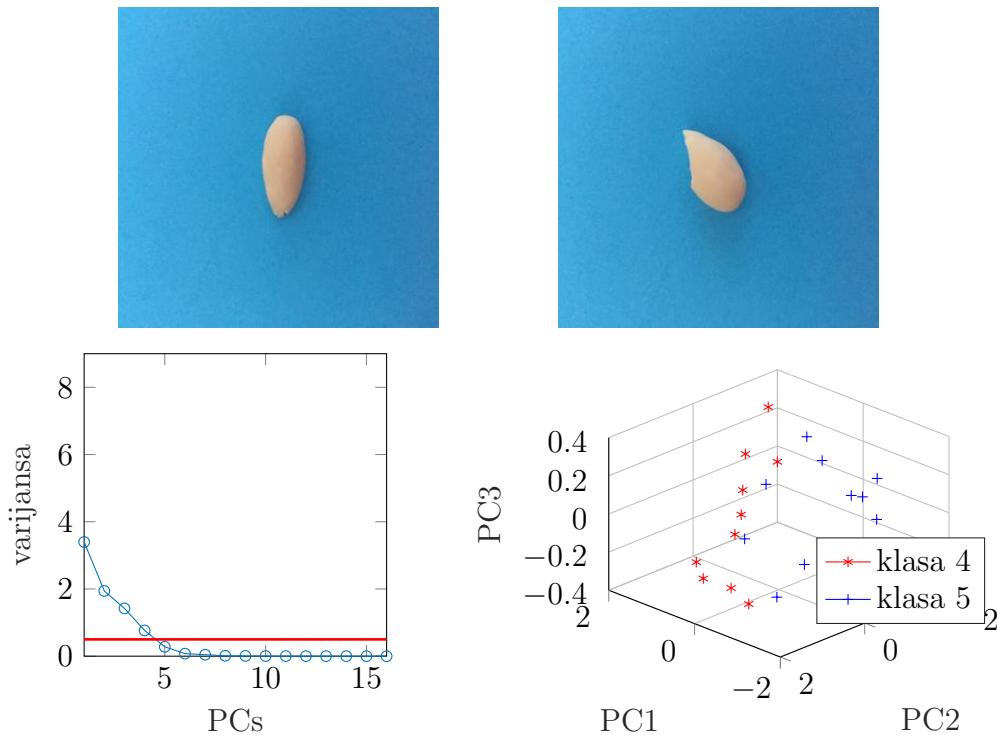
Grafički prikaz nakon zanemarivanja irelevantnih značajki je prikazan na slici 5.8. Na osnovu rekonstruisanih vrijednosti glavnih komponenata i matrice zasićenja nakon zanemarivanja irelevantnih značajki može se vidjeti da značajke koje najviše utiču na razlikovanje ove dvije klase su komponente R, G i B dobijene usrednjavanjem piksela originalne slike i kružnost.

**Slika 5.8:** Usporedba i analiza klase 4 i 6

5.4.2.5 Usporedba i analiza klase 4 i 5

Slično kao i u prethodnim razmatranjima izvršena je analiza klase 4 i 5 odnosno čitavih blanširanih badema i slomljenih blanširanih badema. Na slici 5.9 je prikazana varijansa, gdje se može zaključiti da su samo prve četiri glavne komponente bitne za dalju analizu dok se ostale zanemaruju. Odnosno, u prve četiri komponente sadržano je 94.6168% ukupne varijanse.

Grafički prikaz nakon zanemarivanja irelevantnih značajki je prikazan na slici 5.9. Na osnovu rekonstruisane vrijednosti glavnih komponenata i matrice zasićenja nakon zanemarivanja irelevantnih značajki može se vidjeti da značajke koje najviše utiču na razlikovanje ove dvije klase su: površina, obim, dužina, širina, cirkularnost, kružnost i omjer. Ovi rezultati su bili i za očekivati jer se slomljeni i čitavi intuitivno mogu razlikovati po površini, obimu i drugim značajkama koji ukazuju na nepravilnosti oblika. Intuitivni rezultati su potvrđeni PCA metodom.



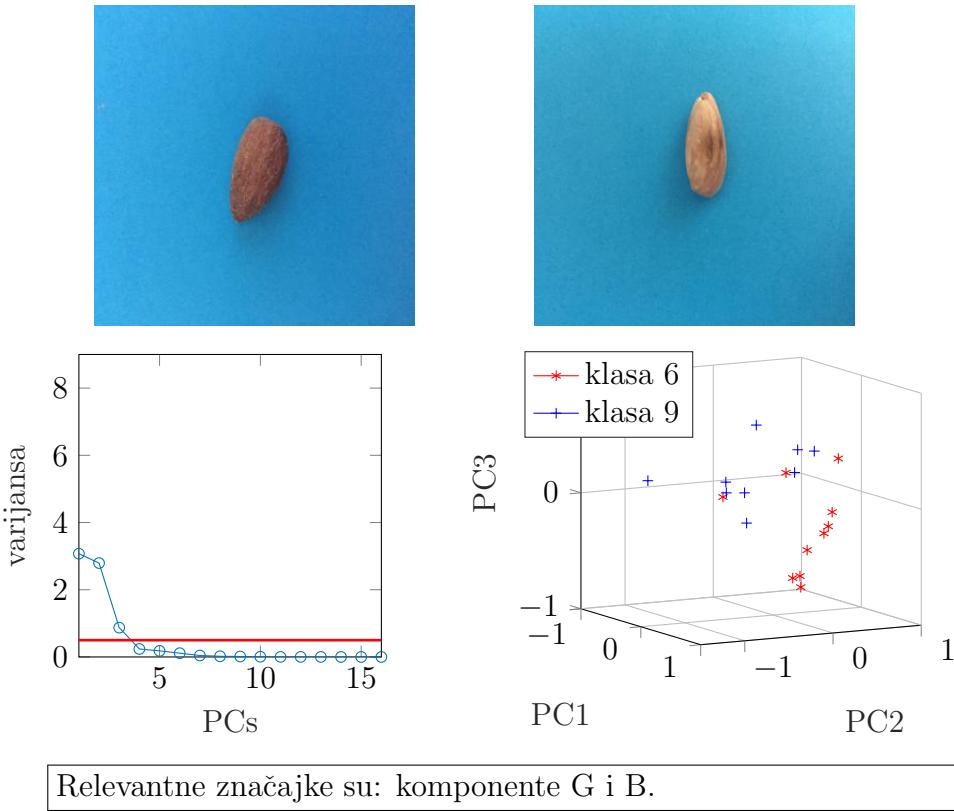
Relevantne značajke su: površina, obim, dužina, širina, cirkularnost, kružnost i omjer.

Slika 5.9: Usporedba i analiza klase 4 i 5

5.4.2.6 Usporedba i analiza klase 6 i 9

U okviru analize značajki izvršena je analiza klase 6 i 9 odnosno čitavih pečenih badema i čitavih blansiranih - pečenih badema. Na slici 5.10 je prikazana varijansa, gdje se može zaključiti da su samo prve tri glavne komponente bitne za dalju analizu dok se ostale zanemaruju. Odnosno u prve tri komponente sadržano je 91.5733% ukupne varijanse.

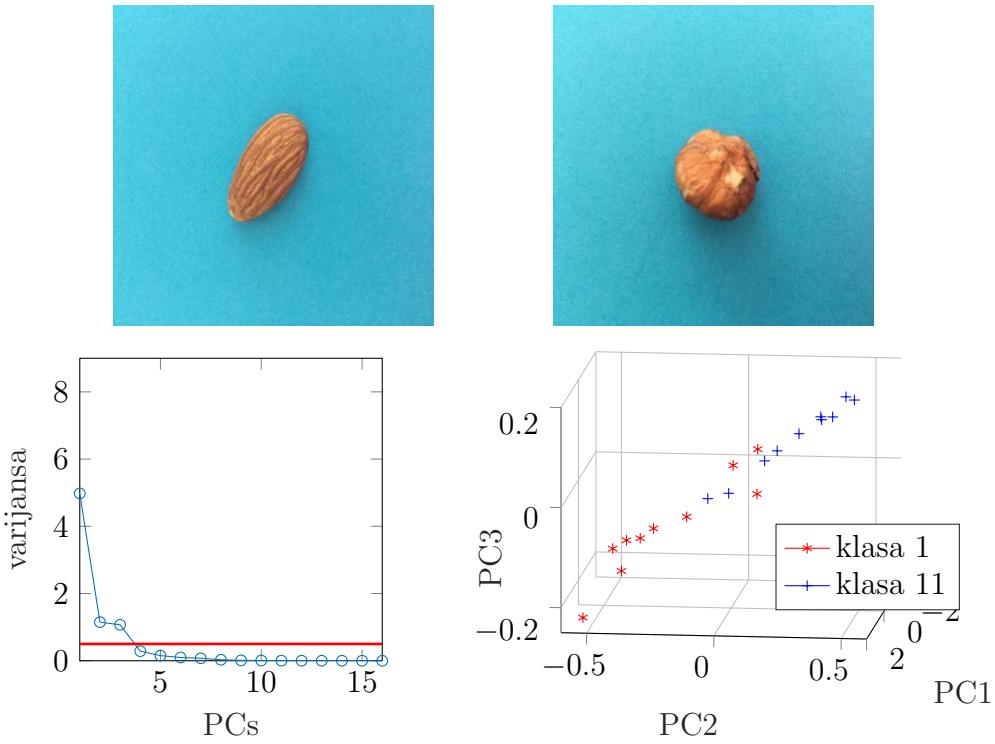
Grafički prikaz nakon zanemarivanja irrelevantnih značajki je prikazan na slici 5.10. Na osnovu rekonstruisane vrijednosti glavnih komponenata i matrice zasićenja nakon zanemarivanja irrelevantnih značajki može se vidjeti da značajke koje najviše utiču na razlikovanje ove dvije klase su komponente G i B dobivene usrednjavanjem piksela originalne slike. Također značajnu ulogu pri razlikovanju instanci ovih klasa ima komponente B dobivena usrednjavanjem piksela segmentiranog badema. Rezultati su intuitivni jer za razlikovanje pečenih badema koji imaju smeđu boju od blansirano - pečenih koji su bijelo-smeđi je najpogodnije koristiti značajke boje.

**Slika 5.10:** Usporedba i analiza klase 6 i 9

5.4.2.7 Usporedba i analiza klase 1 i 11

U okviru analize značajki izvršena je analiza klase 1 i 11 odnosno čitavih sirovih badema i nepoznate klase u kojoj se nalaze lješnjaci. Na slici 5.11 je prikazana varijansa, gdje se može zaključiti da su samo prve tri glavne komponente bitne za dalju analizu dok se ostale zanemaruju. Odnosno u prve tri komponente sadržano je 91.5690% ukupne varijanse.

Grafički prikaz nakon zanemarivanja irelevantnih značajki je prikazan na slici 5.11. Na osnovu rekonstruisane vrijednosti glavnih komponenata i matrice zasićenja nakon zanemarivanja irelevantnih značajki može se vidjeti da značajke koje najviše utiču na razlikovanje ove dvije klase su: cirkularnost, površina, ekscentricitet i boja dobivena usrednjavanjem piksela originalne slike. Odnosno za razlikovanje sirovih čitavih badema i lješnjaka je najpogodnije koristiti boju i značajke oblika cirkularnosti i ekscentriciteta što se može i intuitivno zaključiti.



Relevantne značajke su: cirkularnost, površina, ekscentricitet i boja dobivena usrednjavanjem piksela originalne slike.

Slika 5.11: Usporedba i analiza klase 1 i 11

Identičan postupak se može provesti i za preostale kombinacije klasa badema. Dobiveni rezultati nakon provedene analize značajki su korišteni za metode klasifikacije, kako bi se ispravno prepoznale klase badema. Rezultati dobiveni pomoću PCA, odnosno koje značajke su najpogodnije za razlikovanje pojedinih klasa badema su korišteni za formiranje hibridne strukture za klasifikaciju pomoću metoda mašine vektorske podrške (engl. *support vector machine*) i neuronske mreže (engl. *neural networks*). U narednom poglavljju su detaljno opisane ove metode kao i njihova primjena na klasifikaciju postojeće baze badema.

Poglavlje 6

Metode klasifikacije

U ovom poglavlju, razmatra se problem klasifikacije, koji predstavlja razvrstavanja nepoznate instance u jednu od unaprijed ponuđenih kategorija. U ovom dijelu biće riječi o klasifikacijskim algoritmima, koji su kasnije korišteni u eksperimentalnim istraživanjima za raspoznavanje različitih vrsta i stanja badema.

6.1 Pojam klasifikacije

Klasifikacija je operacija koja smješta svaku instancu iz skupa podataka koji se izučava u jednu od specifičnih, unaprijed određenih klasa, na osnovu karakteristika (osobina) te instance [40]. Pojedinačna instanca se dodjeljuje klasi koristeći algoritam odnosno skup pravila koji formira model. Zadatak je da se na osnovu karakteristika objekata čija klasifikacija je unaprijed poznata, napravi takav model na osnovu koga će se vršiti klasifikacija novih objekata. U problemu klasifikacije, broj klasa je unaprijed poznat i ograničen.

Klasifikacija koristi skup podataka za treniranje i testiranje. U skupu podataka za treniranje svaka instanca je opisana određenim brojem atributa. Cilj klasifikacije je pronaći model za atribut kao funkciju vrijednosti drugih atributa. Algoritam klasifikacije na osnovu ulaznog trening seta i odgovarajuće labele klase proizvodi klasifikator. Pri klasifikaciji se svaki objekat svrstava u neku od klasa sa određenom tačnošću. Cilj klasifikacije je da se kada dođe nova instanca kojoj nije poznata labela klase na osnovu izgrađenog modela izvrši dodjeljivanje instance jednoj od poznatih klasa onoliko tačno koliko je to moguće.

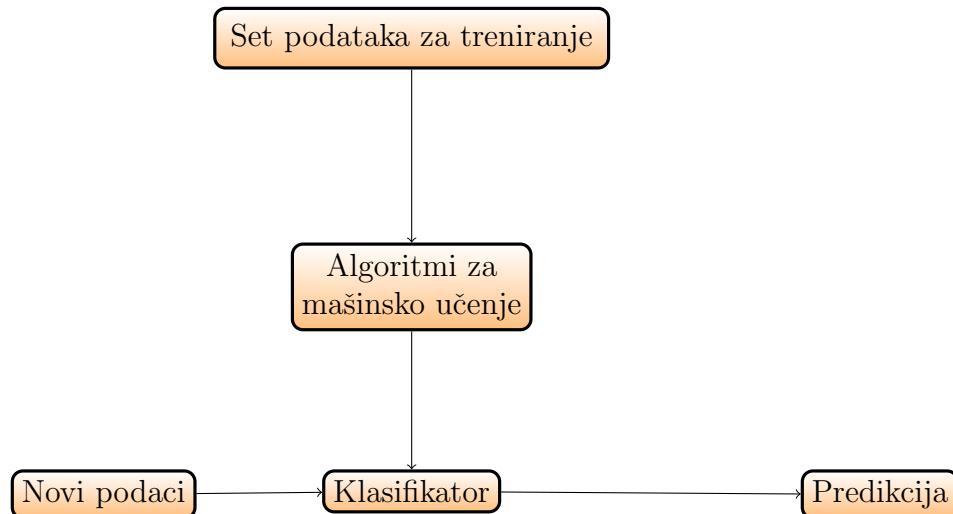
Proces klasifikacije se izvršava u četiri koraka [40]

1. faza treniranja - ovaj korak se radi nad skupom podataka koji je nasumično odabran iz skupa podataka koji se modelira. Sve instance u ovom koraku imaju poznatu labelu (klasu). S obzirom da je poznata labela klase za svaku trening instancu, ovaj korak je poznat kao korak nadziranog učenja u kojem se pravi klasifikator "učeći" iz trening skupa načinjenog od instanci iz baze podataka i njihovih dodijeljenih labela klase,
2. faza testiranja - u ovom koraku se testira model koji je nastao u prethodnom koraku.

Postoji više metoda za procjenu tačnosti klasifikatora. Ako se tačnost klasifikatora smatra prihvatljivom onda se on može koristiti za klasifikaciju budućih instanci podataka za koje labela nije poznata,

3. faza validacije - mjerjenje performansi odabranog modela,
4. primjena - rezultirajući model se primjenjuje na cijeli set podataka koji se modelira.

U praksi trening, test i validacijski set podataka mogu biti u istom fizičkom setu podataka ili u odvojenim setovima podataka. Za sve algoritme klasifikacije bitno je izvršiti predprocesiranje podataka i metode čišćenja podataka. Na slici 6.1 je prikazan model za izgradnju klasifikatora i njegovo testiranje.



Slika 6.1: Pojednostavljena blok struktura modela za izgradnju klasifikatora

Najpoznatije metode klasifikacije su:

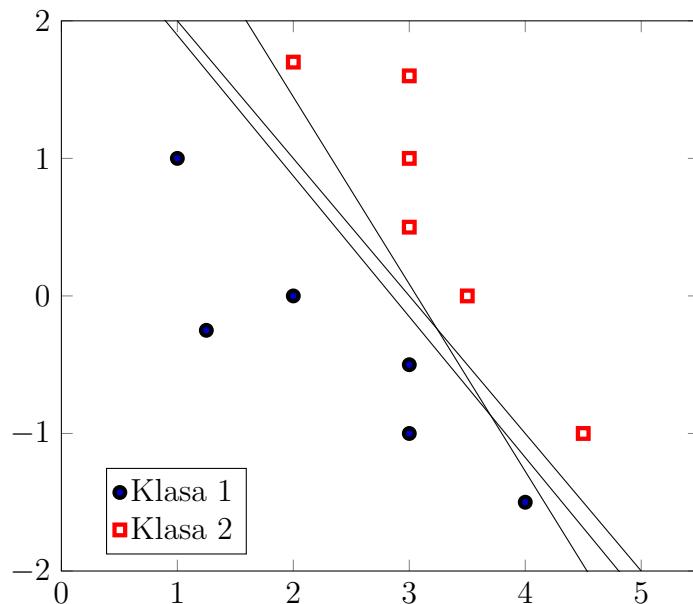
- stablo odlučivanja (engl. *decision tree*),
- maštine vektorske podrške (engl. *support vector machine*),
- Naive Bayesovi klasifikatori,
- neuronske mreže,
- k - najbližih susjeda,
- Bayesove mreže,
- genetski algoritmi,
- indukcijska pravila.

U okviru ovog rada za klasifikaciju badema u 11 klasa korištena je metoda maština vektorske podrške (SVM) i neuronskih mreža. Zadatak višeklasne klasifikacije se može svesti na više zadataka binarnih klasifikacija, koje je moguće efikasno riješiti koristeći binarne

klasifikatore. Najuspješniji i najčešće korišteni binarni klasifikatori su mašine vektorske podrške. Neuronske mreže za razliku od SVM podržavaju mogućnost višeklasne klasifikacije te stoga nije potrebno kombinovati više binarnih klasifikatora kao u slučaju SVM.

6.2 Mašine vektorske podrške - SVM

SVM (eng. *Support Vector Machine*) klasifikatori predstavljaju popularnu metodu klasifikacije, koja se bazira prvenstveno na klasifikaciji podataka u dvije klase. Ukratko se ideja klasifikatora može predstaviti kao određivanje hiperravnih koja će razdvojiti dvije klase [41]. SVM klasifikatori su relativno nov algoritam, u poređenju sa ostalim metodama nadziranog/superviziranog učenja, a početak se veže za 1962. godinu i rad ruskih naučnika Vladimira Naumovicha Vapnika i Alexandra Yakovlevicha Leinera [42]. Ideja algoritma se može predstaviti na slici 6.2.



Slika 6.2: Klasifikacija korištenjem SVM klasifikatora

Na slici 6.2 je moguće primjetiti da su granice oblasti zapravo prave, a da je klasifikaciju moguće izvršiti različitim pravim. U opštem slučaju ovo može biti i hiperravan za n -dimenzionalni problem, kao što je ranije navedeno. Za razliku od ostalih algoritama klasifikacije, gdje je cilj minimizirati grešku na izlazu klasifikatora (kao kod npr. neuronskih mreža), cilj SVM klasifikatora je da pronađe takvu granicu između klasa, koja će dati maksimalno odstojanje od analiziranih podataka [43]. Kao što je moguće primjetiti na slici 6.2 ispravnu klasifikaciju je moguće izvršiti na više načina (pomoću više različitih pravih), međutim klasifikator je tačniji i pouzdaniji što je granica više udaljena od podataka. U skladu sa ovim problem SVM klasifikacije se može formulisati na način opisan u nastavku.

Neka je dat skup uređenih parova $A = x_i, y_i$, gdje su x_i ulazni podaci i predstavljaju n -dimenzionalne vektore, dok su y_i pripadajuće klase koje mogu imati vrijednost 0, ili 1 (općenito klasa 1 i klasa 2). Cilj klasifikacije je naći hiperravan koja će podijeliti prostor

na dva dijela u skladu sa ulaznim podacima. Ako se ta ravan opiše sa svojim parametrima \mathbf{W} (vektor normale na ravan) i b (slobodni član), onda se može pronaći funkcija pripadnosti D za svaki ulazni podatak u skladu sa relacijom 6.1.

$$D(\mathbf{x}) = \mathbf{W}\mathbf{x} + b \quad (6.1)$$

Ako se radi o linearно-separabilnom problemu i ako je klasifikator ispravno treniran, onda za podatak \mathbf{x} vrijedi relacija 6.2.

$$\mathbf{x} \in \begin{cases} 0, & D(\mathbf{x}) < 0 \\ 1, & D(\mathbf{x}) \geq 0 \end{cases} \quad (6.2)$$

Postupak treniranja se ponavlja dok se svi podaci ispravno klasificiraju (odnosno dok se ne postigne odgovarajuća granica). U praksi međutim može doći do slučajeva, gdje podaci nisu linearno odvojivi. U takvim situacijama prvo rješenje je povećanje dimenzije (npr. ako se podaci ne mogu razdvojiti pravom, onda se kao granica između podataka može usvojiti ravan) [44]. Povećanje dimenzionalnosti problema dovodi do povećanja i kompleksnosti algoritma. Stoga, kako bi se smanjila računarska kompleksnost uvode se *kernel* funkcije. Pomoću njih se može postići da hiperravan bude linearna kombinacija polaznih podataka, odnosno da jednačina hiperravnih bude u skladu sa relacijom 6.3:

$$\sum_i \alpha_i K(x_i, x) = const, \quad (6.3)$$

gdje je K funkcija kernela. Neke od najpoznatijih funkcija kernela uključuju:

- homogene $K(x_i, x) = (x_i x)^d$
- radijalne $K(x_i, x) = e^{\lambda \|x_i x\|^2}$
- Gaussove $K(x_i, x) = e^{-\frac{\|x_i x\|^2}{2\sigma^2}}$
- sigmoidne $K(x_i, x) = (\tanh(x_i x) + c), c < 0$

Drugi problem SVM klasifikatora je što je predviđen za binarnu klasifikaciju. Iz same definicije se moglo primjetiti da je ovo klasifikator koji vrši klasifikaciju za dvije klase. Poseban izazov jeste primjena ovog klasifikatora za probleme više klasa, pa postoje određeni načini na koji se to može riješiti [45]. Jedan od načina je da se za svaku klasu kreira poseban klasifikator, pa da se onda izlazi posmatraju kao pripadnost klasi, odnosno nepripadnost klasi. Za kraj se može zaključiti da SVM klasifikatori predstavljaju moćan alat, ali koji može biti dosta kompleksan, te se teško nosi sa višeklasnim problemima.

Mašine vektorske podrške imaju nekoliko prednosti u odnosu na ostale modele :

- jednostavno operiraju nad velikim brojem varijabli,
- otpornije na problem "preučavanja" (engl. *overfitting*),
- pružanje boljih predviđanja neviđenih test podataka,
- pružanje jedinstvenih optimalnih rješenja za problem u treniranju,
- postojanje manje parametara za optimizaciju u poređenju sa drugim metodama.

6.3 Neuronske mreže

Najvažnija stvar koja ljudi razlikuje od ostalih jeste svijest i sposobnost racionalnog razmišljanja. Kako je čovjek u svojoj historiji nastojao da oponaša pa čak i kreira život, tako ne trebaju da čude i naporu ljudske civilizacije za razvojem sistema koji ima sposobnost racionalnog razmišljanja. Iako je ovo još uvijek u sferi naučne fantazije i *SF* filmova, područje vještačke inteligencije se sve više razvija, pa su stoga nastali razni algoritmi koji su na neki način učinili da robot "razmišlja" i bude svjestan svoje okoline. Stoga, sama ideja vještačkih neuronskih mreža je dočekana sa velikim oduševljenjem i očekivanjima, koja su znatno spala tokom godina. Motivacija uvođenja neuronskih mreža je uzrokovana problemima koji se u opštem slučaju ne mogu formulirati nekim algoritmom. Npr. procjena vrijednosti nekog proizvoda je uzrokovana dosadašnjim iskustvima osobe. Osoba može procijeniti vrijednost proizvoda, bez da se ranije susrela sa njim, poučena ranijim iskustvima. Ako bi se sličan problem postavio računaru, bez nekih ulaznih informacija, pa čak i algoritma, računar ne bi bio u stanju procijeniti vrijednost nekog proizvoda (računar može primiti informacije o proizvodu, ali bez postojanja nekog algoritma, neće moći ocijeniti vrijednost proizvoda). Iskustvo osoba stiče učenjem, te se prilagođava novim situacijama, dok računar, iako nesumnjivo kvalitetnije i brže može riješiti kompleksniji problem, nema mogućnost adaptacije. Upravo, je ovo dovelo do razvoja vještačkih neuronskih mreža, koje bi omogućili računaru da uči, usvaja nova znanja i prilagođava se situaciji. Osnovni cilj neuronskih mreža je da uče, npr. iz skupova za treniranje, pa se ne moraju eksplicitno programirati. Nakon procesa učenja, neuronske mreže mogu razviti sposobnost da povezuju i generaliziraju podatke. Ako je učenje uspješno, neuronska mreža može pronaći adekvatna rješenje za slične probleme prisutne u skupovima za treniranje, čak ako ove podatke ranije nije analizirala. Posebna prednost neuronskih mreža je velik nivo neosjetljivosti na šumove [46]. Najčešće primjene neuronskih mreža su u aproksimaciji funkcije, uključujući probleme predikcije, zatim klasifikaciji (eng. *pattern recognition*), procesiranju signala (eng. *signal processing*, npr. filtriranje, kompresija, klasterizacija), robotici, upravljanju, itd. Posebni primjeri neuronskih mreža su u video-igramama (npr. šah, ili poker), zatim identifikaciji sistema, medicini, data miningu, zaustavljanje spamova u okviru e-maila, itd.

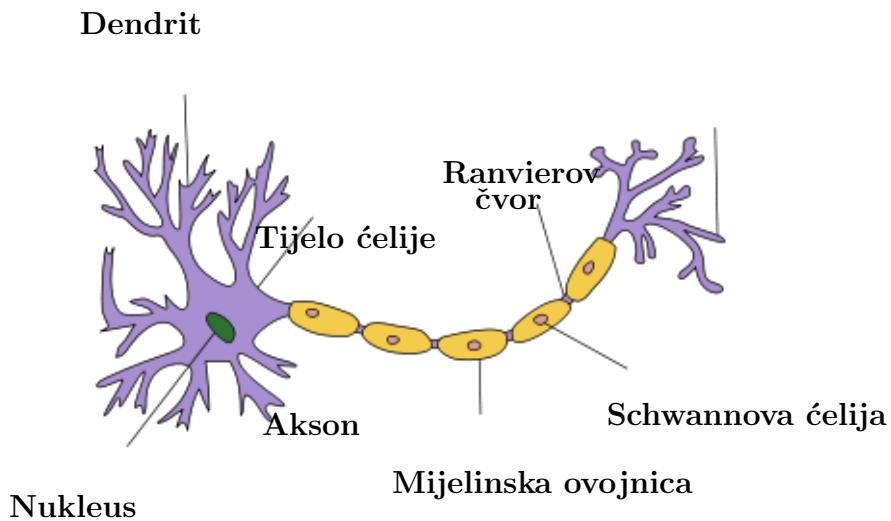
6.3.1 Historijski razvoj neuronskih mreža

Početak neuronskih mreža veže se za 1943. godinu, kada su naučnici Warren McCulloch i Walter Pitts objavili rad [47], kojim su dokazali da jednostavne neuronske mreže mogu izračunati bilo koju aritmetičku ili logičku funkciju. Ovaj rad je dočekan sa velikim oduševljenjem, imao je veliki utjecaj, a inspirisao je i rade Norberta Wienera i von Neumanna u kojem se obrađuju mogućnosti računara dizajniranog na bazi ljudskog mozga. Tako je 1951. godine konstruisan prvi neuroračunar pod nazivom *Snark* od strane Marvina Minskyog. Iako je ispravno radio, računar nije bio u stanju da obavi procesiranje informacija od interesa. Frank Rosenblatt, Charles Wightman i ostali su 1957. kreirali novi neuroračunar pod nazivom (*Mark I*). Nešto kasnije Bernard Widrow i grupa njegovih studenata su kreirali ADALINE (Multi ADAptive LINear Elements), procesor na bazi neuronskih mreža. Veliki uspjeh neuronskih mreža u početku doveo je do prevelikih očekivanja, koji su ipak bili nerealni čak i za današnje vrijeme (npr. očekivalo se da će vještački mozgovi biti kreirani za samo nekoliko godina), a sam interes i en-

tuzijazam izazvan neuroskim mrežama su učinili da se ovaj period zove i zlatni period neuronskih mreža. Minsky i Seymour Pappert su prvi prepoznali probleme neuronskih mreža. 70-ih godina prošlog stoljeća neuronske mreže se počinju primjenjivati u raznim aplikacijama, kao što su adaptivna obrada signala, prepoznavanje oblika, itd. 80-ih godina ponovo je porastao interes za neuronske mreže, prije svega kada je John Hopfield pokazao mogućnosti neuronskih mreža na stvarnim uređajima, a 1987. održana i je i prva IEEE naučna konferencija na temu neuronskih mreža sa 1800 učesnika. Danas se neuronske mreže primjenjuju u raznim oblastima i obzirom da su modelirani na osnovu ljudskog neurona i mozga, uvijek imaju poseban značaj u oblasti vještačke inteligencije. Međutim, postavlja se pitanje do koje mjere su neuronske mreže u stanju da obnašaju mozak. Sa druge strane druge metode, kao što su *SVM klasifikatori* postepeno zamjenjuju neuronske mreže u oblasti mašinskog učenja (eng. *machine learning*). Istovremeno, uređaji na bazi *CMOS* tehnologije se u posljednje vrijeme koriste za simulaciju biofizičkih procesa. Obzirom da se uređaji pokušavaju smanjiti na nano nivo, uređaji bazirani na neuronskim mrežama bi mogli zamijeniti digitalne računare, obzirom da se zasnivaju na učenju, ne na programiranju. U novije vrijeme (od 2009. godine) na raznim takmičenjima u oblasti pattern recognitiona (prepoznavanje rukopisa), machine learning algoritmi na bazi neuronskih mreža su osvajali prva mesta, bazirani na rekurentnim feedforward algoritmima [48].

6.3.2 Arhitektura neuronskih mreža

Neuronska mreža se sastoji od mnoštva neurona, koji su organizovani u slojeve. Pri tome izgradnja čitave neuronske mreže počinje od neurona. Primjer neurona je dat na slici 6.3.



Slika 6.3: Primjer jednog neurona

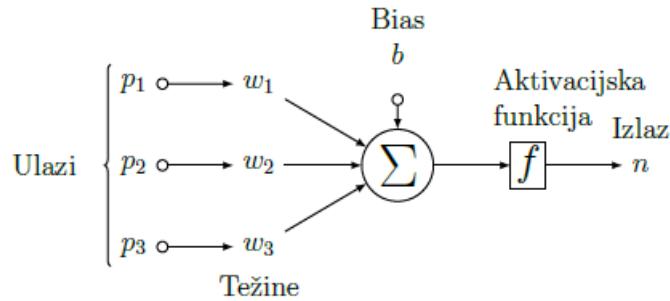
Ulez u neuron je dat sa p i predstavlja ulaz u neuronsku mrežu, ili izlaz iz susjednog neurona. Ovaj ulaz se množi težinskim koeficijentom w i sabira sa internim ulazom pomaka b (eng. *bias*). Kao što se primjeti, na osnovu ovih koeficijenta se formira a , odnosno aktivacijsko djelovanje. Djelovanje a na osnovu funkcije f proizvodi izlaz neuronske mreže dat relacijom 6.4:

$$n = f(a) = f(Wp + b) \quad (6.4)$$

Pri tome neuronska mreža može imati više ulaza p , pa je u tom slučaju relacija data, kao 6.5:

$$n = f(\mathbf{a}) = f(\mathbf{W}\mathbf{p} + b), \quad (6.5)$$

obzirom da su \mathbf{p} i \mathbf{W} predstavljeni vektorima. Iz strukture pojedinačnog neurona se može uočiti da svojstvo neurona ovisi od aktivacijske funkcije, a da se podešavanja vrši preko težinskih koeficijenata \mathbf{W} i pomakom b . Pri tome se isti ulaz može dovoditi na više pojedinačnih neurona. Ovakva struktura, gdje se više ulaza istovremeno dovodi na ulaz više neurona, naziva slojem. Na slici 6.4 je prikazan primjer jednoslojne neuronske mreže.

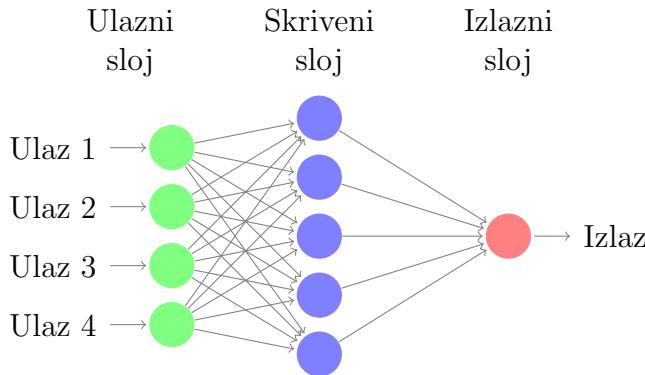


Slika 6.4: Struktura jednog neurona

Ovo je neuronska mreža sa jednim neuronom. Međutim, u praksi se koriste složenije neuronske mreže. Moguće je da se neuronska mreža sastoji od više neurona, stoga je moguće imati neuronsku mrežu sa n ulaza i m izlaza, odnosno m neurona, što čini jedan sloj neuronske mreže. Ipak, dosta češća situacija je da se izlazi jedne ovakve neuronske mreže dovode na ulaz druge neuronske mreže. Ovo su višeslojne neuronske mreže, koje se i najčešće primjenjuju. Ovakve mreže se sastoje od:

- **ulaznog sloja**, čiji su ulazi zapravo realni ulazi (slika iz stvarnog svijeta, ulaz sa senzora, neki zvuk, ili sl.),
- **skrivenog sloja**, ulazi su zapravo izlazi prethodnog sloja neuronske mreže, a izlazi su ulazi u naredni sloj neuronske mreže,
- **izlaznog sloja**, ulazi su izlazi prethodnog sloja neuronske mreže, a izlazi neuronske mreže su izlazi u stvarni svijet.

Neuronska mreža ima po jedan ulazni i izlazni sloj, a može imati proizvoljan broj skrivenih slojeva. Primjer jedne višeslojne neuronske mreže je dat na slici 6.5.

**Slika 6.5:** Struktura višeslojne neuronske mreže

U skladu sa ovim neuronska mreža se može definisati kao: "uređena trojka (N, V, ω) sa dva skupa N i V i funkcijom ω , gdje je N skup neurona, a V uređeni skup $V = (i, j) \mid i, j \in \mathbb{N}$ čiji elementi predstavljaju veze između neurona i i j . Funkcija $w : V \rightarrow \mathbb{R}$ definiše težine gdje je $w(i, j)$, ili drugačije napisano $w_{i,j}$ težina između neurona i i j . Težine su predstavljene matricom W , gdje red predstavlja neuron na početku, a kolona neuron na kraju veze. U slučaju da nema veze između neurona težina $w_{i,j}$ postaje nula" [46].

6.3.3 Aktivacijske funkcije

Kada se kreira neuronska mreža, da bi se ona upotrijebila za odgovarajuću svrhu neophodno je odrediti ispravne vrijednosti težinskih koeficijenata i bias ulaza. Iz prethodne slike moguće je primijetiti da izlaz neuronske mreže ovisi od aktivacijske funkcije f i težinskih koeficijenata W . Ideja treniranja je da se vrši modifikacija težina, dok neuronska mreža na izlazu ne generiše željeni izlaz, obzirom da jednom usvojene aktivacijske funkcije nije moguće mijenjati. Ukoliko na početku nisu odabrane zadovoljavajuće aktivacijske funkcije, tada nije moguće dobiti željeni izlaz procesom treniranja (kao što je prikazano na primjeru perceptronu i XOR funkcije). Što se tiče aktivacijskih funkcija, postoji nekoliko tipičnih aktivacijskih funkcija koje se najčešće koriste, prikazanih na slici 6.6, a to su:

- **hardlim**

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (6.6)$$

- **purelin**

$$f(x) = x \quad (6.7)$$

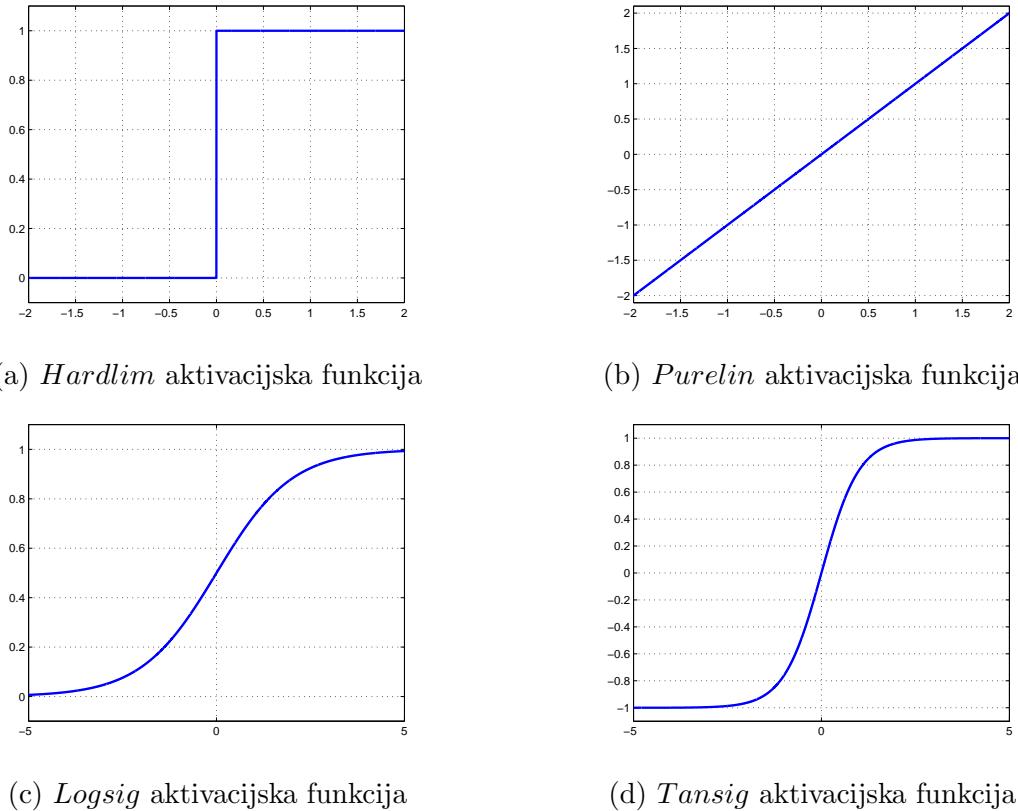
- **logsig**

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6.8)$$

- **tansig**

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (6.9)$$

Naravno, moguće je koristiti i drugačije definisane aktivacijske funkcije poput *hardlims*, *satlin*, koje predstavljaju prethodno navedene funkcije za zasićenjima.



Slika 6.6: Primjeri aktivacijskih funkcija

6.3.4 Učenje i treniranje neuronske mreže

Kao što je i ranije navedeno, osnovna motivacija neuronskih mreža je njihova sposobnost da oponašaju ljudske neuronske mreže. Osnovna osobina neuronskih mreža jeste da uče iz podataka, te učenjem poboljšavaju svoju performansu. Ranije je naglašeno da se prilagođavanje izlaza neuronske mreže vrši podešavanjem težinskih koeficijenata w , pa samim tim algoritam učenja se svodi na podešavanje ovih koeficijenata u skladu sa određenim kriterijem. Učenje se može podijeliti na tri vrste:

- Nadzirano/supervizirano učenje (eng. *supervised learning*) - kod ove vrste učenja se koristi znanje vanjskog supervizora, odnosno učitelja. Neuronskoj mreži se daju izlazne vrijednosti, koje bi se morale dobiti na osnovu određenih ulaza, odnosno mreži se daje uređeni par (x, y) , $x \in X, y \in Y$, a cilj je pronaći funkciju $f : X \rightarrow Y$. Informacije o željenim izlazima su apriori informacije i dolaze od supervizora. Mjera poklapanja stvarnih izlaza za željenim se najčešće daje srednjom kvadratnom greškom (eng. *Least Mean Square error - LMS*), pa je cilj naći f za koju je ova greška minimalna. Obično se ovo učenje koristi za probleme klasifikacije (pattern recognition) i probleme aproksimacije.
- Nenadzirano/nesupervizirano učenje (eng. *unsupervised learning*) - kod ovakve vrste učenja se ne koristi znanje supervizora, već se na osnovu lokalnih informacija

pokušava pronaći sličnost u ulaznim podacima. Obično se daje neka funkcija cijene J , koja ovisi od ulaza x i funkcije izlaza $f(x)$, a cilj je da se ova funkcija minimizira. Npr. funkcija cijene J može biti definisana relacijom 6.10

$$J = (f(x) - x)^2 \quad (6.10)$$

Primjeri nenadziranog učenja su Hebbianovo učenje (eng. *Hebbian learning*), kompetitivno učenje (eng. *Competitive learning*), a koriste se za probleme estimacije, probleme klasterizacije, filtriranja i slično. Ono što je važno istaći je da se nadzirano učenje koristi u off-line aplikacijama, dok se nenadzirano učenje koristi za on-line aplikacije.

- Pojačano učenje (eng. *reinforcement learning*) - kod ovog vrste učenja posmatraju se izlazi neuronske mreže, te se ocjenjuju. Problem se može posmatrati na način da se u nekom trenutku t na osnovu ulaza x_t formira izlaz y_t , a akcija proizvodi nagradu c_t . Cilj ovog učenja je da se pronađu izlazi takvi da se dobija najveća moguća nagrada. Također, neuronske mreže se često koriste kao dio većeg algoritma za *reinforcement learning*, kao što je kombinacija dinamičkog programiranja i neuronskih mreža (Neuro-dinamičko programiranje) i koristi se za nelinearne probleme, kao što su problem upravljanja resursima, problemi u medicini, ili problem putanja vozila. Pored toga ovo učenje se koristi za probleme upravljanja, video-igre, itd.

Poglavlje 7

Primjena metoda klasifikacije na postojeću bazu

U okviru klasifikacijskih metoda za postojeću bazu badema odabrane su prethodno obrađene metode mašine vektorske podrške i neuronske mreže.

7.1 Primjena mašina vektorske podrške

S obzirom da ova metoda predstavlja binarnu klasifikaciju, potrebno je koristiti više binarnih klasifikatora kako bi se postigla višeklasna klasifikacija.

Postoje tri pristupa na koje je to moguće ostvariti:

- jedan naspram jedan (engl. *one - vs - one*) - ukoliko postoji N različitih klasa u ovom pristupu potrebno je napraviti $\frac{N(N-1)}{2}$ klasifikatora. Svi binarni klasifikatori se treniraju sa jednim parom klasa od mogućih, $\frac{N(N-1)}{2}$ parova, čime se uče razlikovati dvije klase. Na kraju se primjenjuje shema glasovanja, pri čemu je pobjednik klase koja ima najveći broj glasova i nepoznati podaci dobivaju oznaku iste,
- jedan naspram svih (engl. *one - vs - all*) - strategija uključuje treniranje jednog binarnog klasifikatora po klasi, pri čemu se instance te klase uzimaju kao pozitivne, a sve ostale kao negativne. Ukoliko imamo N klasa tada je potrebno istrenirati ukupno N klasifikatora.
- hibridni pristup - u ovom pristupu se koristi kombinacija OvO i OvA gdje broj klasifikatora varira u ovisnosti od problema koji treba riješiti.

7.1.1 ”Jedan naspram svih” pristup

Kako je ranije objašnjeno u poglavlju 3, postojeća baza se sastoji od ukupno 110 slika na kojima su različite vrste badema u različitim oblicima, slikani pod istim uvjetima. Nakon što je izvršena segmentacija badema, a potom i ekstrakcija osobina slijedi postupak klasifikacije, u okviru kojeg je potrebno prepoznati kojoj klasi pripada analizirani badem.

Pristup jedan naspram svih se sastoji od ukupno 11 klasifikatora. Prvi klasifikator prepoznaže da li uzorak pripada klasi 1 ili ne, drugi da li pripada klasi 2 i tako redom. Glavni nedostatak ovog pristupa jeste da se može desiti da uzorak koji se klasificira može pripadati više od jednoj klasi. S obzirom da nije moguće utvrditi kojoj od prepoznatih klasa uzorak pripada, uzeto je da prva klasa koja se prepozna označava pripadnost klase razmatranog uzorka. Iako tačnost i pouzdanost ovog pristupa upravo zbog ovog problema nije izražena, ovom pristupu zbog malog broja klasifikatora je potrebno dosta manje vremena za klasifikaciju jedne slike od preostalih pristupa.

Za analizu su razmatrana dva omjera raspodjele slika iz baze. U omjeru 50 : 50, 55 slika koje su nasumično izabrane, pri čemu je 5 slika iz svake klase za treniranje, dok je preostalih 55 slika, uzeto za testiranje. U drugom slučaju je odabran omjer 80 : 20 pri čemu je 80% odnosno 88 slika (8 slika iz svake klase) uzeto za treniranje, dok je preostalih 20%, odnosno 22 slike uzeto za testiranje.

U sljedećim tabelama 7.1 i 7.2 su prikazani rezultati klasifikacije. Slike za treniranje i testiranje su birane nasumično i u ovisnosti od odabranih slika rezultati blago variraju u opsegu od 49 – 60% za omjer 50 : 50, dok je za omjer 80 : 20 tačnost u opsegu 50 – 70%. Vrijeme potrebno za klasifikaciju jedne slike je oko $3.6[ms]$, što predstavlja jako dobro vrijeme, pogodno za klasifikaciju u realnom vremenu. Također, može se primjetiti da su rezultati bolji kod omjera 80 : 20 što se moglo i očekivati obzirom da u ovom omjeru ima više slika za treniranje nego za testiranje.

Pri klasifikaciji nije vršena analiza značajki za pojedine klasifikatore, već su korištene sve izdvojene značajke. Ovaj prisutn može biti pogodan da se izdvoje predstavnici jedne klase npr. da li je proizvod škart ili nije.

Tabela 7.1: Nasumično odabrani uzorci slika u omjeru 50 : 50

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	60	5.78
2	52.72	6.1
3	52.72	4.8
4	49.09	3.6
5	43.63	4.37

Tabela 7.2: Nasumično odabrani uzorci slika u omjeru 80 : 20

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	68.18	5.6
2	59.09	5.37
3	54.54	5.38
4	54.54	5.8
5	52.72	5.1

Tačnost ovim pristupom ipak nije zadovoljavajuća i rezultati nisu dovoljno pouzdani, obzirom da se sa sigurnošću ne može utvrditi kojoj od rezultujućih klasa pripada razmatrani uzorak. Stoga ovaj pristup nije pogodan za korištenje. Jedina prednost je vrijeme potrebno za klasifikaciju i mali broj klasifikatora potreban za klasifikaciju. Pseudokod pristupa jedan naspram svih je prikazan u nastavku.

Algorithm 1: SVM pristup *One Vs All*

- 1 Odabir broja trenirajućih i testnih slika iz baze
- 2 Segmentacija slike
- 3 Ekstrakcija značajki boje i oblika i normalizacija izdvojenih značajki
- 4 Treniranje klasifikatora
 - 1: **if** pripadnost klasi 1 **then**
 - 2: return rezultat
 - 3: **end if**
 - 4: **if** pripadnost klasi 2 **then**
 - 5: return rezultat
 - 6: **end if**
 - 7: **if** pripadnost klasi 3 **then**
 - 8: return rezultat
 - 9: **end if**
 - 10: **if** pripadnost klasi 4 **then**
 - 11: return rezultat
 - 12: **end if**
 - 13: **if** pripadnost klasi 5 **then**
 - 14: return rezultat
 - 15: **end if**
 - 16: **if** pripadnost klasi 6 **then**
 - 17: return rezultat
 - 18: **end if**
 - 19: **if** pripadnost klasi 7 **then**
 - 20: return rezultat
 - 21: **end if**
 - 22: **if** pripadnost klasi 8 **then**
 - 23: return rezultat
 - 24: **end if**
 - 25: **if** pripadnost klasi 9 **then**
 - 26: return rezultat
 - 27: **end if**
 - 28: **if** pripadnost klasi 10 **then**
 - 29: return rezultat
 - 30: **end if**
 - 31: **if** pripadnost klasi 11 **then**
 - 32: return rezultat
 - 33: **end if**

Result: Prikaz rezultata

7.1.2 "Jedan naspram jedan" pristup

Glavni nedostatak prethodno obrađenog pristupa je nemogućnost odlučivanja kojoj od prepoznatih klasa pripada analizirani uzorak. U ovom pristupu izgrađeno je $\frac{N(N-1)}{2}$ klasifikatora. Obzirom da postoji 11 klasa to je ukupno 55 klasifikatora. Svaki klasifikator raspoznaće kojoj od dvije klase pripada razmatrani uzorak. Prvi klasifikator vrši klasifikaciju uzorka u klasu 1 ili 2, drugi u klasu 1 ili 3, treći klasu 1 ili 4 i tako redom svaku kombinaciju ukupno njih 55. Na kraju se vrši prebrojavanje rezultata i ona klasa koja ima najviše glasova se uzima kao pobjednik i razmatranom uzorku se dodjeljuje labela te klase.

Isto kao i u prethodnom slučaju su razmatrana dva omjera raspodjele slika iz baze, omjer 50 : 50 i 80 : 20. U sljedećim tabelama 7.3 i 7.4 slike su odabrane nasumično i u ovisnosti od raspodjele slika varira i tačnost. Za razliku od prethodnog slučaja gdje su se koristile sve izdvojene značajke, ovdje su odabrane različite značajke za različite klasifikatore. Zavisno od klasifikatora korištene su značajke boje i značajke oblika.

Tabela 7.3: Nasumično odabrani uzorci slika u omjeru 50 : 50

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	78.18	20.18
2	78.18	19.31
3	74.54	19.47
4	72.72	20.63
5	63.63	19.63

Tabela 7.4: Nasumično odabrani uzorci slika u omjeru 80 : 20

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	95.45	18.3
2	81.81	20.15
3	81.81	17.86
4	77.27	15.27
5	68.18	18.8

Ovaj pristup se pokazao dosta bolji od prethodnog, gdje je tačnost dosta bolja, i u ovisnosti od rasporeda slika za treniranje i testiranje tačnost se kreće u opsegu od 60% – 80% za omjer 50 : 50 i 70% – 95% za omjer 80 : 20. Glavni nedostatak ovog pristupa su vremenske performanse jer je potrebno dosta više vremena za samu klasifikaciju s obzirom da svaki uzorak mora proći kroz 55 klasifikatora, a potom je potrebno izvršiti shemu glasovanja kako bi se utvrdilo kojoj klasi pripada razmatrani uzorak. Ovim se nameće potreba da se uvede hibridni metod koji bi kompenzirao nedostatke oba pristupa. Hibridni metod bi trebao biti dovoljno tačan kao pristup jedan naspram jedan, a opet dovoljno brz i jednostavan sa malim brojem klasifikatora kao pristup jedan naspram svih.

Pseudokod pristpa "jedan naspram jedan" je opisan u nastavku.

Algorithm 2: SVM *One vs One* princip

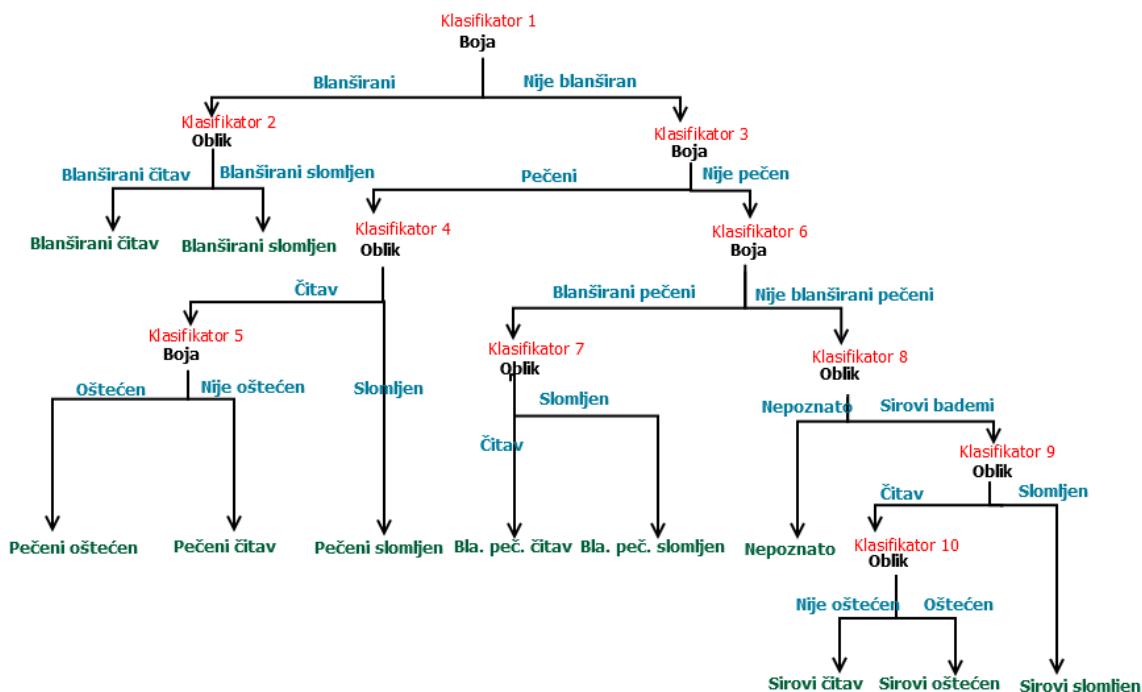
- 1 Odabir broja trenirajućih i testnih slika iz baze
- 2 Segmentacija slike
- 3 Ekstrakcija značajki boje i oblika i normalizacija izdvojenih značajki
 - 1: **for** $i = 1 : 11$ **do**
 - 2: **for** $j = i + 1 : 11$ **do**
 - 3: treniraj klasifikatore
 - 4: **end for**
 - 5: **end for**
 - 6: **for** $i = 1 : 11$ **do**
 - 7: **for** $j = i + 1 : 11$ **do**
 - 8: testiraj klasifikatore
 - 9: **end for**
 - 10: **end for**

Izvrši prebrojavanje rezultata

Prikaz rezultata

7.1.3 Hibridni pristup

Pored navedena dva pristupa u okviru ovog rada napravljen je hibridni metod koji prevaziča nedostatke prethodno dvije opisane metode. Na slici 7.1 prikazano je stablo hibridnih klasifikatora. Za ovaj pristup potrebno je ukupno 10 klasifikatora koji odlučuju o pripadnosti klasi razmatranog uzorka kao što je to prikazano na slici ispod. Prvi klasifikator provjerava da li se radi o blanširanom ili neblanširanom bademu na osnovu karakteristika boje. Ukoliko se radi o blanširanom bademu klasifikator 2 provjerava da li je navedeni uzorak čitav ili slomljen na osnovu značajki oblika. Ukoliko se ne radi o blanširanom bademu onda se provjerava da li je razmatrani badem pečen ili nije. Ukoliko je pečen klasifikator 4 provjerava da li je čitav ili slomljen. Ukoliko je badem čitav klasifikator 5 provjerava da li je razmatrani badem oštećen ili nije i tako redom na osnovu stabla prikazanog na slici 7.1. Na osnovu ove procedure dolazi se do pripadnosti badema jednoj od 11 mogućih klasa.



Slika 7.1: Stablo hibridnih klasifikatora za SVM i neuronske mreže

Na osnovu provedene analize značajke pomoću metode glavnih komponenata opisane u poglavljju 4, korištene su rezultujuće značajke za pojedine klasifikatore. U tabelama 7.5 i 7.6 za klasifikaciju uzeti su omjeri kao i u prethodnim pristupima sa nasumično odabranim slikama za treniranje i testiranje. Značajke za pojedine klasifikatore su većinom odabrane u skladu sa rezultatima metode glavnih komponenata.

Tabela 7.5: Nasumično odabrani uzorci slika u omjeru 50 : 50

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	92.72	5.83
2	81.81	3.67
3	80	3.22
4	78.18	4.37
5	72.72	4.18

Tabela 7.6: Nasumično odabrani uzorci slika u omjeru 80 : 20

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	90.90	5.89
2	90.90	6.87
3	81.81	5.12
4	77.27	3.26
5	72.72	4.19

Na osnovu rezultata može se vidjeti da tačnost varira u opsegu od 80% – 100% za oba omjera. Ovaj pristup se sastoji od samo 10 klasifikatora pa je i vrijeme obrade pojedine slike dosta manje od prethodna dva metoda koji imaju po 11 i 55 klasifikatora te je ovo čini pogodnom je za klasifikaciju u realnom vremenu. U svrhu povećanja tačnosti algoritma korištene su različite kernel funkcije u svim SVM klasifikatorima prikazane u tabeli 7.7. U tabeli 7.7 su također prikazane korištene značajke oblika i boje.

U nastavku je prikazan pseudokod hibridnog pristupa.

Input : Odabir broja trenirajućih i testnih slika iz baze

```
1 Segmentacija slike
2 Ekstrakcija značajki boje i oblika i normalizacija izdvojenih značajki
3 Treniranje klasifikatora i odabir značajki za klasifikaciju
4 if blansirani badem then
5   if čitav blansiran then
6     | return čitav blansiran
7   else
8     | return slomljen blansiran
9   end
10 else
11   if pečen then
12     if slomljen pečen then
13       | return pečen slomljen
14     else
15       if pečen ostecen then
16         | return pečen ostecen
17       else
18         | return pečen čitav
19       end
20     end
21   else
22     if blansirani pečeni then
23       if blansirani pečeni čitav then
24         | return blansirani pečeni čitav
25       else
26         | return blansirani pečeni slomljen
27       end
28     else
29       if sirovi then
30         if sirov slomljen then
31           | return sirov slomljen
32       else
33         if sirov oštećen then
34           | return sirov oštećen
35       else
36         | return sirov čitav
37       end
38     end
39   else
40     | return nepoznato
41   end
42 end
43 end
44 end
45 Testiranje klasifikatora
```

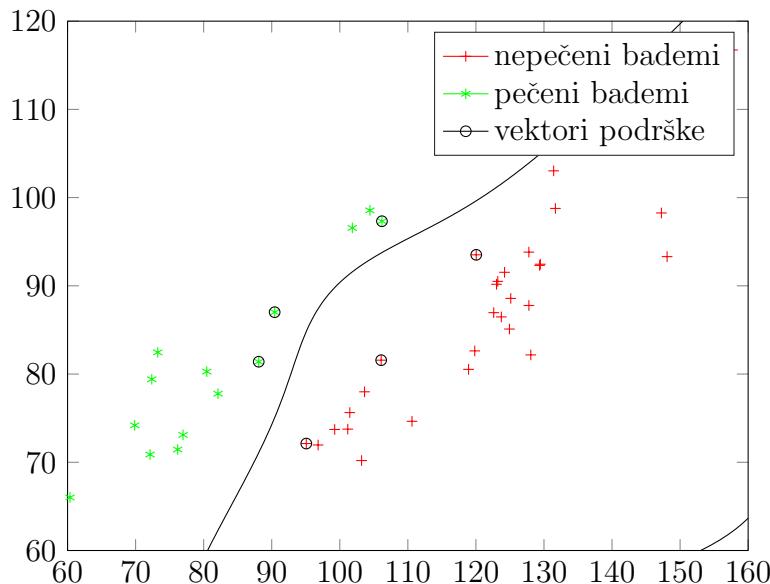
Output: Prikaz rezultata

Tabela 7.7: Kernel funkcije i značajke korištene za hibridnu SVM baziranu klasifikaciju

Klasifikator / Stanje	Kernel funkcije	Značajke
blansirani	polynomial ($n=4$)	R i G komponenta (boja)
blansirani čitavi	polynomial ($n=3$)	površina i kružnost (oblik)
pečeni	polynomial ($n=4$)	G i B komponenta (boja)
pečeni čitav	polynomial ($n=4$)	površina, kružnost i omjer dužine i širine (oblik)
pečeni oštećen	Gaussian	G i B komponenta (boja)
blansirani pečeni	polynomial ($n=5$)	srednja vrijednost R i srednja vrijednost B komponente (boja)
blansirani pečeni čitavi	Gaussian	površina, kružnost and ekscentricitet (oblik)
sirovi	polynomial ($n = 5$)	površina i kružnost (oblik)
sirovi slomljeni	Gaussian	površina i kružnost (oblik)
sirovi čitavi	Gaussian	R i B komponenta (boja)

Ovaj pristup daje bolju tačnost i vremenske performanse od pristupa jedan naspram jedan i jedan naspram svih. Na osnovu rezultata tačnosti klasifikacije u prethodnim tabelama može se zaključiti da je ovaj pristup bolji nego pristupi "jedan naspram jedan" i "jedan naspram svih". Ukoliko se porede vremenske performanse iz prethodnih tabela može se vidjeti da za klasifikaciju jedne slike je potrebno u prosjeku oko $4ms$ za razliku od pristupa "jedan naspram jedan" kome u prosjeku treba oko $20ms$ za iste rezultate. Pristup "jedan naspram svih" zahtjeva oko $5ms$ za klasifikaciju slike međutim tačnost algoritma je značajno manja nego kod hibridnog pristupa. Ovime hibridni pristup prednjači u odnosu na prethodna dva pristupa, a s obzirom da je $4ms$ vrijeme koje je tolerantno za čekanje rezultata klasifikacije jedne slike, algoritam je pogodan za korištenje u realnom vremenu.

Pri samom razdvajanju klasa, svaki klasifikator pronalazi takvu pravu, odnosno ravan koju je potrebno provući između instanci klasa, tako da su instance jedne klase na jednoj strani prave ili ravni. Na slici 7.2 je prikazan treći klasifikator sa slike 7.1 koji ispituje pripadnost klasi III, odnosno da li je badem pečen ili nije. Na grafiku su prikazane instance klase III i preostale klase (I, IV i V). Između njih je provučena kriva koja daje maksimalno odstojanje od analiziranih podataka i daje najbolje rezultate.

**Slika 7.2:** Primjer razdvajanja klase pečenih i nepečenih badema pomoću SVM metode

7.1.4 Poređenje rezultata SVM metoda

Za usporedbu sva tri pristupa uzet je isti skup trening i test slika za sve tri metode u omjeru 50 : 50 i 80 : 20. Rezultati su prikazani u tabeli 7.8 i 7.9. Najbolje rezultate je dao hibridni metod, dok su preostale dvije metode nešto manje tačne, što se moglo i očekivati jer je hibridni metod dizajniran sa ciljem da maksimizira tačnost. Druga prednost ovog algoritma jest da se mogu odabrati značajke u skladu sa rezultatima PCA metode, dok u drugim metodama relevantne značajke za klasifikatore se ne mogu odrediti, što za posljedicu ima manju tačnost. Metoda jedan naspram jedan također daje dobre rezultate međutim treba joj dosta više vremena za klasifikaciju, što se moglo i očekivati obzirom da svaka slika mora proći kroz 55 klasifikatora. Hibridni metod u prosjeku treba samo 5.5ms za klasifikaciju slike što je znatno manje od SVM One-vs-one pristupa kojem u prosjeku treba 22.8ms.

Tabela 7.8: Rezultati klasifikacije SVM metoda za omjer 50(%):50(%) za isti skup testnih i trening slika

Metoda klasifikacije	Tačnost (%)	Br. pogodjenih slika	Pros. vri. [ms]/sl.
SVM Hibrid	92.72	51	5.5
SVM One vs One	87.27	48	22.8
SVM One vs All	43.63	24	4.2

Ukoliko se promjeni omjer na 80 : 20 rezultati su prikazani u tabeli 7.9. Tačnosti su dosta slične prethodnim, dok se u pojedinim slučajevim tačnost poboljšala. Najbolji algoritam je i dalje hibridni SVM, potom ide SVM One-vs-one i na kraju SVM One-vs all.

Tabela 7.9: Rezultati klasifikatora sa trening i test slikama u omjeru 80(%):20(%)

Metoda klasifikacije	Tačnost (%)	Br. pogodjenih slika	Pros. vri. [ms]/sl.
SVM Hibrid	90.90	20	5.5
SVM One vs One	81.81	18	22.8
SVM One vs All	50	11	4.2

Matrica konfuzije za SVM hibrid je prikazana tabelom 7.10. Redovi matrice predstavljaju stvarnu klasu, dok kolone predviđene klase od strane izgrađenog modela. Vrijednosti na dijagonali predstavljaju broj slika koje su ispravno klasificirane. U ovom slučaju sve instance klase V, odnosno nepoznata klasa je ispravno prepoznata. Najviše grešaka ima u klasi I, odnosno sirovim bademima kojih ima ukupno 15. Od njih 12 je prepoznao ispravno, a jedan badem prepoznao kao blanširan, jedan kao blanširani pečen i jedan kao nepoznat.

Tabela 7.10: Matrica konfuzije klasa za SVM hibridni pristup

Stvarna/Prediktovana	Klasa I	Klasa II	Klasa III	Klasa IV	Klasa V
Klasa I	12	1	0	1	1
Klasa II	0	10	0	0	0
Klasa III	0	0	15	0	0
Klasa IV	1	0	0	9	0
Klasa V	0	0	0	0	5

U tabeli 7.11 prikazana je matrica stanja (čitav, slomljen ili oštećen badem). Redovi odgovaraju stvarnom stanju, dok kolone odgovaraju prediktovanom stanju od strane izgrađenog modela. Vrijednosti na dijagonali predstavljaju broj stanja koji su ispravno klasificirani. Tako se može vidjeti da su svi čitavi bademi ispravno klasificirani, dok je jedan oštećeni badem prepoznat kao čitav.

Tabela 7.11: Matrica konfuzije stanja za SVM hibridni pristup

Stvarno stanje/ Prediktovano stanje	Čitav	Slomljen	Oštećen
Čitav	25	0	0
Slomljen	0	20	0
Oštećen	1	0	9

7.2 Primjena neuronskih mreža

Za klasifikaciju postojeće baze primijenjene su i neuronske mreže. Za razliku od SVM-a, neuronske mreže mogu vršiti višeklasnu klasifikaciju bez potrebe uvođenje više binarnih klasifikatora. U okviru primjene ove metode klasifikacije obrađena su tri pristupa:

- neuronske mreže sa N klase,
- neuronske mreže sa M klase i P stanja,
- hibridne neuronske mreže.

7.2.1 Neuronske mreže sa N klasa

Ovaj pristup se bazira na direktnom određivanju klasa koristeći neuronske mreže. Značajke trening skupa su korištene kao ulazi u neuronsku mrežu, dok su izlazi klase badema. Stoga će svaki badem biti odmah direktno klasiran u jednu od 11 klasa. Ovaj pristup koristi samo jednu neuronsku mrežu sa dva skrivena sloja za klasifikaciju.

U tabeli 7.12 prikazani su rezultati 5 puta nasumično odabranih slika za treniranje i testiranje u omjeru 50 : 50 , gdje se iz svake klase uzima po 5 slika, ukupno za treniranje i testiranje po 55 slika. Za razliku od SVM metode klasifikacije gdje pri odabiru istih trening i test slika pri svakom pokretanju se dobiju isti rezultati u neuronskim mrežama to nije slučaj. Neuronske mreže pri treniranju istih podataka mogu dati različite rezultate, stoga je za iste rezultate potrebno snimiti istreniranu mrežu. U tabeli 7.13 su prikazani rezultati za omjer 80 : 20 odnosno 88 slika za treniranje i 22 slike za testiranje. Rezultati su nešto bolji u ovoj tabeli što se moglo i očekivati jer ima više slika za treniranje što utiče na bolju tačnost.

Opseg tačnosti varira od 50% – 80% zavisno od odabira trening i test slika. Vrijeme potrebno za klasifikaciju jedne slike je u prosjeku oko $10ms$. S obzirom da su korištene sve značajke bez odabira relevantnih za pojedinu klasifikaciju rezultati su zadovoljavajući.

Tabela 7.12: Nasumično odabrani uzorci slika u omjeru 50 : 50 za neuronsku mrežu sa N klasa

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	72.72	9.18
2	61.81	11.5
3	54.54	10.9
4	54.54	8.37
3	52.72	10.22

U tabeli 7.13 prikazani su rezultati nasumično odabranih slika pet puta u omjeru 80 : 20.

Tabela 7.13: Nasumično odabrani uzorci slika u omjeru 80 : 20 za neuronsku mrežu sa N klasa

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	81.81	9.18
2	81.81	8.37
3	72.72	11.5
4	72.72	10.22
5	45.45	10.9

Najveća prednost ovog pristupa je njegova jednostavnost i mala kompleksnost jer koristi samo jednu neuronsku mrežu. Međutim korištenjem svih značajki za klasifikaciju čini proces treniranja nešto težim. Također rezultati nisu dovoljno pouzdani obzirom da su za klasifikaciju korištene sve značajke i nije primijenjena analiza rezultata značajki za klasifikaciju.

7.2.2 Neuronske mreže sa M klasa i P stanja

Za ovaj pristup svaki badem je opisan jednom od V klasa (sirov, blanširan, pečen, blanširan-pečen, nepoznato) i stanjem (čitav, slomljen, oštećen). Korištena je jedna neuronska mreža za prepoznavanje klase i stanja. Ovaj algoritam kao i prethodni još uvijek je jednostavan i lakše je odabratи značajke za klasifikaciju od prethodnog pristupa sa N klasa. U ovom slučaju je korištena *logsig* aktivacijska funkcija za drugi sloj.

U tabelama 7.14 i 7.15 su prikazani rezultati 5 puta nasumično odabranih slika u omjerima 50 : 50 i 80 : 20. Rezultati su nešto bolji u drugoj tabeli, gdje ima više slika za treniranje nego testiranje.

Tabela 7.14: Nasumično odabrani uzorci slika u omjeru 50 : 50 za neuronsku mrežu sa M klasa i P stanja

Red. br.	Tačne klase %	Tačna stanja %	Uk. tačnost %	Pros. vri. po slici [ms]
1	80	80	70.90	9.55
2	74.54	80	67.27	10.3
3	74.54	74	63.63	11.3
4	74.54	68	49.09	9.5
5	78.18	62	40	10.4

Tabela 7.15: Nasumično odabrani uzorci slika u omjeru 80 : 20 za neuronsku mrežu sa M klasa i P stanja

Red. br.	Tačne klase %	Tačna stanja %	Uk. tačnost %	Pros. vri. po slici [ms]
1	90.90	85	81.81	9.55
4	81.81	90	77.27	9.5
3	81.81	85	68.18	11.3
2	72.72	70	54.54	10.3
5	72.72	65	45.45	10.4

Ovaj pristup je nešto bolji od prethodnog, a još uvijek je dovoljno jednostavan za korištenje. Međutim potrebno je izvršiti dodatnu modifikaciju ovog algoritma koji će moći izabrati značajke bitne za pojedinu klasifikaciju. U svrhu ovog napravljen je hibridni pristup pomoću neuronskih mreža.

7.2.3 Hibridne neuronske mreže

Slično kao u SVM hibridnoj metodi izvršena je klasifikacija badema putem neuronskih mreža. Napravljeno je ukupno 10 neuronskih mreža sa dva sloja prema slici 7.1. Tako prva neuronska mreža raspoznaće da li je badem blanširan ili nije, ako je blanširan druga neuronska mreža provjerava da li je badem čitav ili slomljen i tako redom prema stablu na slici 7.1.

U tabeli 7.16 prikazan je broj neurona korištenim u neuronskim mrežama. Za trenirajuću funkciju odabrana je RP (engl. *Resilient propagation*). Značajke za svaku neuronsku mrežu su podešene u skladu sa tabelom 7.7, čime se maksimizira tačnost.

Tabela 7.16: Parametri neuronskih mreža

Klasa / Stanje	Broj neurona
NN-blanširani	100
NN-blanširani čitavi	100
NN-pečeni	100
NN-pečeni slomljeni	100
NN-pečeni čitavi	1500
NN-blanširani-pečeni	1000
NN-blanširani-pečeni čitavi	1000
NN-sirovi	1000
NN-sirovi slomljeni	100
NN-sirovi čitavi	100

Slično kao i u prethodnim metodama za ispitivanje tačnosti uzeta su dva omjera 50 : 50 i 80 : 20. Slike su birane nasumično pet puta i rezultati su predstavljeni u tabelama 7.17 i 7.18. Rezultati su bolji nego u prethodnim metodama, jer su izabrane samo ciljane značajke koje doprinose razlikovanju dvije klase. Glavni nedostatak ove metode je vrijeme klasifikacije, s obzirom da postoji 10 neuronskih mreža vrijeme potrebno za klasifikaciju jedne slike je znatno veće nego kod prethodna dva pristupa. U prosjeku to vrijeme iznosi oko 150ms po slici.

Rezultati kod omjera 80 : 20 su nešto bolji nego kad se koristi isti broj slika za treniranje i testiranje. Raspon tačnosti se kreće od 70% – 90% što ovaj pristup čini pouzdanim za korištenje. Kako ove neuronske mreže rade sa relativno malim brojem podataka, korištene značajke nisu normalizirane već su korištene u izvornom obliku.

Tabela 7.17: Nasumično odabrani uzorci slika u omjeru 50 : 50 za hibridnu neuronsku mrežu

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	81.81	151.15
2	80	160.72
3	77.27	180.34
4	74.54	155.3
5	65.45	170.9

Tabela 7.18: Nasumično odabrani uzorci slika u omjeru 80 : 20 za hibridnu neuornsku mrežu

Red. br.	Tačnost %	Prosječno vrijeme po slici [ms]
1	81.81	151.15
2	77.27	160.72
3	72.72	180.34
4	81.81	155.3
5	86.36	170.9

7.2.4 Poređenje metoda neuronskih mreža

U svrhu poređenja metoda neuronskih mreža u tabeli 7.19 i 7.20 su prikazani rezultati sa sva tri pristupa nad istim trening i test skupovima u omjerima 50 : 50 i 80 : 20. Iz tabele se može vidjeti da u oba slučaja hibridni metod daje najbolje rezultate, ali i najlošiju vremensku performansu. U prosjeku vrijeme obrade jedne slike u hibridnom metodu je oko 150ms dok je za preostala dva metoda potrebno oko 11ms za klasifikaciju jedne slike. Ovo je ujedno i glavni nedostatak ove metode.

Tabela 7.19: Rezultati klasifikatora sa trening i test skupom u omjeru 50(%):50(%)

Klasifikacijska metoda	Tačnost (%)	Br. pogodjenih slika	Pros. vri. [ms]/sl.
Hibridna NN	85.45	47	177.1
NN sa N klasa	72.72	40	10.9
NN sa M klasa i P stanja	70.90	39	10.9

Tabela 7.20: Rezultati klasifikatora sa trening i test skupom u omjeru 80(%):20(%)

Klasifikacijska metoda	Tačnost (%)	Br. pogodjenih slika	Pros. vri. [ms]/sl.
Hibridna NN	86.36	19	177.1
NN sa N klasa	81.81	18	10.9
NN sa M klasa i P stanja	81.81	18	10.9

Za tačnost 86.3636% u omjeru 80 : 20 matrica konfuzije klasa je prikazana tabelom 7.21 i matrica konfuzije stanja u tabeli 7.22. Redovi predstavljaju stvarnu klasu, dok kolone predstavljaju prediktovanu klasu. Vrijednosti na dijagonali predstavljaju broj tačno prediktovanih slika. Tako su sve instance klase I, klase II, klase III i klase IV prepoznate tačno, dok je jedna instance klase V prepoznata kao klasa I. Slično i u matrici konfuzije

stanje, svi oštećeni bademi su prepoznati tačno, dok je jedan čitav badem prepoznat kao slomljen i jedan slomljen badem prepoznat kao čitav.

Tabela 7.21: Matrica konfuzije klasa za NN hibridni pristup

Stvarna/Prediktovana klasa	Klasa I	Klasa II	Klasa III	Klasa IV	Klasa V
Klasa I	6	0	0	0	0
Klasa II	0	4	0	0	0
Klasa III	0	0	6	0	0
Klasa IV	0	0	0	4	0
Klasa V	1	0	0	0	1

Tabela 7.22: Matrica konfuzije stanja za NN hibridni pristup

Stvarno stanje/ Prediktovano stanje	Čitav	Slomljen	Oštećen
Čitav	9	1	0
Slomljen	1	7	0
Oštećen	0	0	4

Poglavlje 8

Zaključak

Tema ovog završnog rada je bila klasifikacija objekata na slici korištenjem superviziranog učenja. U prvom dijelu je dat uvod u analizu i klasifikacijske metode, te pregled istraživanja u ovoj oblasti. Opisane su metode segmentacije i pretpresiranja slike. Posebna pažnja je posvećena metodama pretpresiranja slike i klasifikacijskim metodama koje su korištene u eksperimentalnim istraživanjima. Također je izvršena analiza značajki pomoću metoda glavnih komponenta, čiji su rezultati kasnije korišteni sa klasifikaciju badema.

U drugom dijelu rada objašnjen je sistem mašinskog učenja, kao i metode koje se koriste za klasifikaciju podataka. Posebna pažnja je posvećena mašinama vektorske podrške i neuronskim mrežama, koje su korišteni kao klasifikatori. Izgrađen je hibridni metod koristeći SVM i neuronske mreže, koji se pokazao dosta boljim po pitanju tačnosti i brzine od standardnih metoda SVM za višeklasnu klasifikaciju (jedan naspram jedan i jedan naspram svih) i neuronskih mreža.

Tačnost klasifikacije je moguće povećati pravilnim odabirom značajki oblika i boje. Pokazano je da određeni skupovi parametara različito utiču na sposobnost razlikovanja dvije instance klase pomoću metode glavnih komponenata (PCA). Zbog ovog je od izuzetne važnosti odrediti koje značajke će se koristiti u kojem klasifikatoru. U hibridnom metodu korištene su samo one značajke koje doprinose razlikovanju instanci klase u pojedinim klasifikatorima. Rezultati tačnosti su testirani na nasumično odabranim slikama u omjerima za treniranje i testiranje 50 : 50 i 80 : 20. U skoro svim slučajevima hibridni metod SVM i hibridni metod neuronskih mreža se pokazao jednim od boljih po pitanju tačnosti, dok je hibridni metod SVM također najbolji po pitanju brzine i pogodan je za korištenje u realnom vremenu. Ipak, postavlja se pitanje relevantnosti rezultata dobivenih pomoću algoritma hibridnih neuronskih mreža jer broj uzoraka s kojim je vršeno testiranje je relativno mali. Navedena testiranja su vršena u programskom paketu MATLAB.

Razvoj svih algoritama i analiza značajki je rađena u programskom paketu MATLAB, dok je hibridni metod SVM razvijen i u programskom jeziku C++ koristeći biblioteku programskih funkcija u realnom vremenu OpenCV. Na kraju radi bolje preglednosti i uvida u rezultate klasifikacije izgrađen je korisnički interfejs u programskom paketu MATLAB, za hibridne metode SVM i neuronskih mreža. Također u programskom jeziku C++ izgrađen je jednostavni korsnički interfejs koji omogućava pregled rezultata klasifikacije odabrane slike.

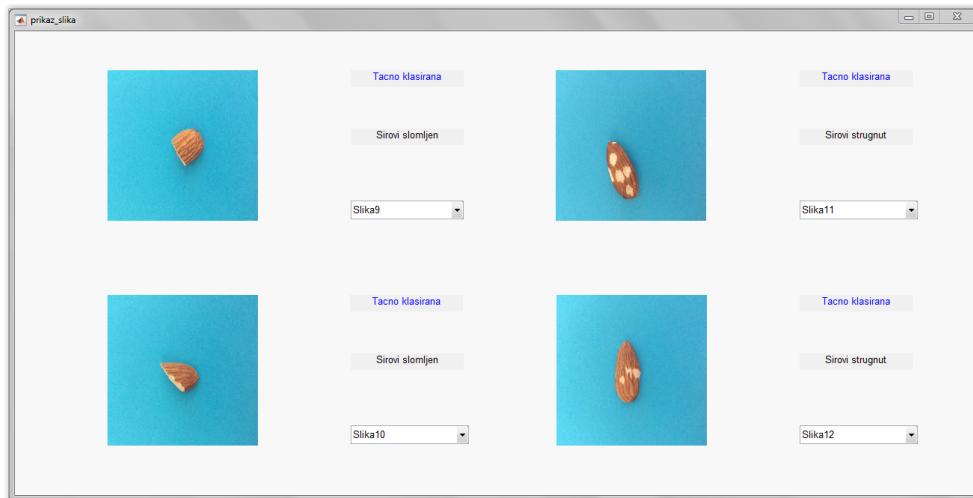
Prilozi

Prilog A

Prilog

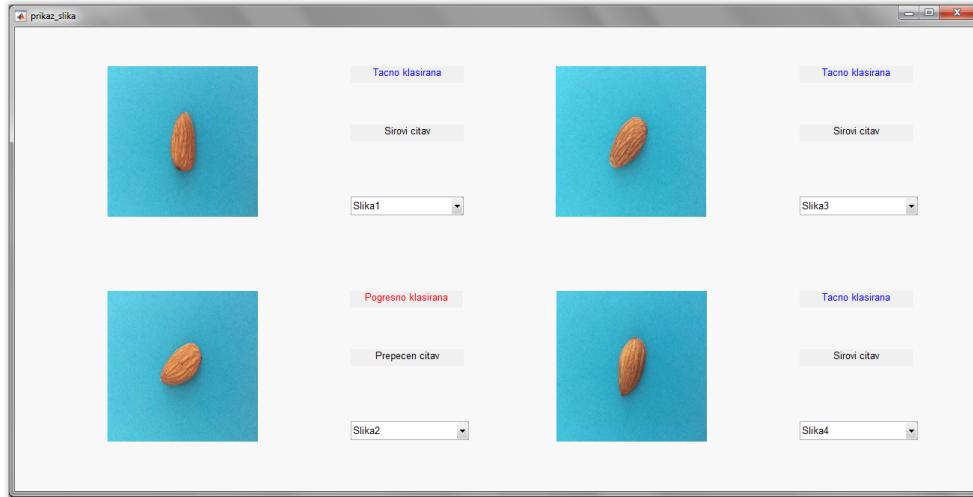
A.1 Razvoj korisničkog interfejsa za SVM i NN hibridni pristup

U okviru ovog rada također je razvijen korisnički interfejs za SVM i NN hibridni pristup, kako bi se ostvarila vizualizacija rezultata klasifikacije. Na slici A.1 je prikazan izgled jedne stranice korisničkog interfejsa koji je razvijen u programskom paketu MATLAB. Prelazak s jedne stranice na drugu se jednostavno ostvaruje pomoću srednjeg klizača (engl. *scroll wheel*) miša.



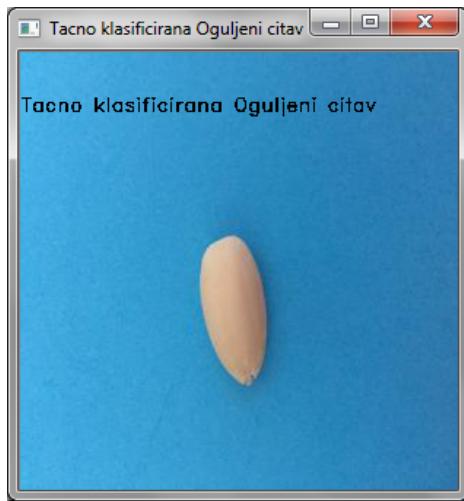
Slika A.1: Izgled korisničkog interfejsa za SVM i NN hibridni metod

Na svakoj stranici je moguće odabratи 4 slike badema iza baze i prikazati rezultate klasifikacije. Ukoliko je badem svrstan u pogrešnu klasu, tada se ispod slike prikaže tekst u koju klasu je klasificiran, kao što je to prikazano na slici A.2.

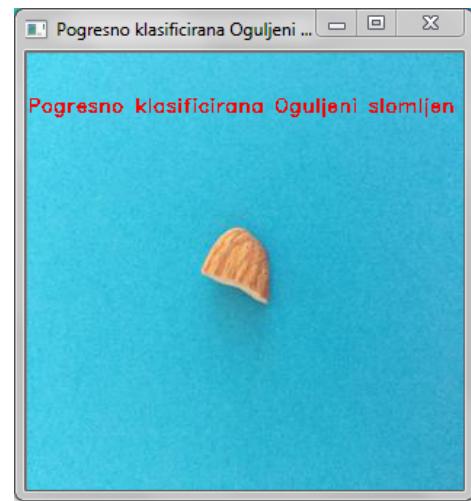


Slika A.2: Izgled korisničkog interfejsa za SVM i NN hibridni metod

U okviru programskog jezika C++ u Visual Studio programskom paketu napravljen je jednostavni korisnički interfejs za pregled rezultata klasifikacije željene slike. Korisnik bira sliku koju želi klasificirati, tako što se otvara prozor za odabir slike, nakon toga se slika prikazuje i daju se rezultati da li je ispravno klasificirana ili nije zajedno sa pripadnošću klase. Na slikama ispod prikazan je izgled dvije odabrane slike badema i njeni rezultati klasifikacije.



Slika A.3: Prikaz tačno klasificirane oda-brane slike u C++



Slika A.4: Prikaz netačno klasificirane odabrane slike u C++

A.2 Programske kodovi

Implementacija algoritama je rađena u programskom paketu MATLAB i u programskom jeziku C++ korištenjem skupine biblioteka OpenCV za obradu slike. U nastavku su dati neki od najvažnijih segmentata koda korištenih u ovom radu.

Kod ispod prikazuje segmentaciju slike pomoću algoritma *K-means* u L*a*b* prostoru boja u programskom paketu MATLAB.

Program A.1: Kmeans segmentacija u MATLABU

```
1 function [label_im,vec_mean] =
2     kmeans_fast_Color(im,no_of_cluster,varargin)
3 if isempty(varargin)
4 cform = makecform('srgb2lab');
5     lab = double(applycform(im,cform));
6     a = lab(:,:,2);
7     b = lab(:,:,3);
8
9     veca= (a(:)-min(a(:))+1);
10    vecb= (b(:)-min(b(:))+1);
11    veca = repmat(veca,[1,no_of_cluster]);
12    vecb = repmat(vecb,[1,no_of_cluster]);
13    meana1=(1:no_of_cluster).*max((veca(:)))/(no_of_cluster+1);
14    meanb1=(1:no_of_cluster).*max((vecb(:)))/(no_of_cluster+1);
15
16    num = length(veca);
17    itr = 0;
18    while(true)
19
20        itr = itr+1;
21        olda=meana1;
22        oldb=meanb1;
23
24        meana = repmat(meana1,[num,1]);
25        meanb= repmat(meanb1,[num,1]);
26
27        dista=((veca-meana).^2);
28        distb=((vecb-meanb).^2);
29
30        clear meana;clear meanb;
31
32        distance = sqrt(dista+distb);
33
34        [~,label_vector] = min(distance,[],2);
35
36        for i=1:no_of_cluster
37            index=(label_vector==i);
38
39            meana1(:,i)=ceil(mean(veca(index)));
40            meanb1(:,i)=ceil(mean(vecb(index)));
41
42        end
43
44        if ((meana1==olda & meanb1==oldb))
45            break;
46        end
47    end
48
49    label_im = reshape(label_vector,[m,n]);
50    vec_mean = [meana1;meanb1];
51    end
51
```

Kod za segmentaciju slike pomoću *K-means* algoritma u programskom jeziku C++.

Program A.2: Kmeans segmentacija u OCV

```
1 #ifndef KMEANS_SEG_H_INCLUDED
2 #define KMEANS_SEG_H_INCLUDED
3
4 #include "opencv2/imgproc/imgproc.hpp"
5 #include "opencv2/highgui/highgui.hpp"
6 using namespace cv;
7 using namespace std;
8
9 Mat kmeans_segmentation(Mat img, int num_clust, int max_iter, vector <
10   float > & color_feat, vector <float> & color_feat_b) {
11   /* Funkcija za k-means segmentaciju;
12    * vrsti se konverzija slike u 3 vektora u double format, zatim se
13    * primjenjuje
14    * k-means algoritam, a onda se vrsti binarna segmentacija */
15   // Konvertovanje u La*b* format
16   Mat img_lab;
17   cvtColor(img, img_lab, CV_BGR2Lab);
18
19   // Konvertovanje slike u float radi proracuna (dvije kolone a* i b*)
20   Mat img_vec(img_lab.rows * img_lab.cols, 2, CV_32F);
21
22   for (int y = 0; y < img_lab.rows; y++)
23     for (int x = 0; x < img_lab.cols; x++)
24       for (int z = 1; z < 3; z++)
25         img_vec.at < float > (y + x * img_lab.rows, z-1) = img_lab.at
26 < Vec3b > (y, x)[z];
27
28   /* Par puta se desilo da se klasterizacija ne izvrsti (vrati se samo
29    * slika). Kako bi se to izbjeglo program upada
30    * u beskonacnu petlju, koja se prekida ako broj piksela u
31    * segmentiranoj slici je veci od 1 */
32   while(1 == 1){
33     // Labele i centri klastera
34   Mat label;
35   Mat cluster_centers;
36   // K-means algoritam u skladu sa dokumentacijom
37   kmeans(img_vec, num_clust, label, TermCriteria(CV_TERMCRIT_ITER +
38     CV_TERMCRIT_EPS, 10, 0.1), max_iter, KMEANS_PP_CENTERS,
39     cluster_centers);
40   // Pretpostavka da je objekt klaster 0
41   int cluster = 0; int non_cluster = 1;
42   float s1; float s2;
43   /* Racuna se zbir centara klastera; kako podloga (ima vece a* i b*
44    * komponente onda
45    * se moze znati u kojem klasteru je podloga */
46   s1 = cluster_centers.at <float>(cluster,0) + cluster_centers.at
47 <float>(cluster,1);
48   s2 = cluster_centers.at <float>(non_cluster,0) + cluster_centers.at
49 <float>(non_cluster,1);
50   if(s2 < s1){
51     cluster = 1;
52     non_cluster=0;
53   }
54   // Slika sa ekstraktovanim objektom
55   Mat ext_img(img.size(), img.type());
56   // Segmentovana slika
57   Mat kmeans_seg_img(img.size(), img.type());
58   // Vektor sa color_feature
```

```
49 color_feat.push_back(0.0);
50 color_feat.push_back(0.0);
51 color_feat.push_back(0.0);
52 /* Vektor sa prosjecnu vrijednost komponente na citavoj slici
53 * ekvivalent trainf2 i b u MATLAB-u*/
54 color_feat_b.push_back(0.0);
55 color_feat_b.push_back(0.0);
56 color_feat_b.push_back(0.0);
57 int br_pix = 0;
58     for (int y = 0; y < img.rows; y++) {
59         for (int x = 0; x < img.cols; x++) {
60             label.at < int > (y + x * img.rows, 0);
61             color_feat_b[0] = (float) img.at < Vec3b > (y, x)[2] +
62                 color_feat_b[0];
63             color_feat_b[1] = (float) img.at < Vec3b > (y, x)[1] +
64                 color_feat_b[1];
65             color_feat_b[2] = (float) img.at < Vec3b > (y, x)[0] +
66                 color_feat_b[2];
67             int cluster_pix = label.at < int > (y + x * img.rows, 0);
68             if (cluster_pix != cluster) {
69                 kmeans_seg_img.at < Vec3b > (y, x)[0] = 0;
70                 kmeans_seg_img.at < Vec3b > (y, x)[1] = 0;
71                 kmeans_seg_img.at < Vec3b > (y, x)[2] = 0;
72                 br_pix++;
73                 /* kako bi se usaglasili formati matlaba (RGB) i OpenCV (BGR) uzima
74                 se da je
75                 * color_feat[0] zapravo img[2]*/
76                 ext_img.at < Vec3b > (y, x)[0] = img.at < Vec3b > (y, x)[0];
77                 color_feat[0] = (float) img.at < Vec3b > (y, x)[2] + color_feat[0];
78                 ext_img.at < Vec3b > (y, x)[1] = img.at < Vec3b > (y, x)[1];
79                 color_feat[1] = (float) img.at < Vec3b > (y, x)[1] + color_feat[1];
80                 ext_img.at < Vec3b > (y, x)[2] = img.at < Vec3b > (y, x)[2];
81                 color_feat[2] = (float) img.at < Vec3b > (y, x)[0] + color_feat[2];
82             }
83         }
84     }
85     if(br_pix > 1){
86
87         color_feat[0] = color_feat[0]/br_pix;
88         color_feat[1] = color_feat[1]/br_pix;
89         color_feat[2] = color_feat[2]/br_pix;
90         color_feat_b[0] = color_feat_b[0]/(img.rows * img.cols);
91         color_feat_b[1] = color_feat_b[1]/(img.rows * img.cols);
92         color_feat_b[2] = color_feat_b[2]/(img.rows * img.cols);
93         return kmeans_seg_img;
94     }
95 }
96 }
```

Ekstrakcija značajki oblika u MATLABU je prikazana kodom ispod.

Program A.3: Ekstrakcija oblika

```

1 function [osobine]=ekstrakcijaOblika2(IM2)
2 %funkcija prima segmentiranu sliku
3 bw = bwareaopen(IM2,30);
4 cmp = bwconncomp(bw);
5 [B,L] = bwboundaries(bw,'noholes');
6 statsBez = regionprops(L,'Centroid','Orientation');
7 angle=statsBez.Orientation;
8 angleToRrotateBy = 90- angle;
9 rotatedImage = imrotate(bw, 30);
10 [B2,L2] = bwboundaries(rotatedImage,'noholes');
11 stats = regionprops(L2,'Area','Centroid','Orientation',
12 'Perimeter','EquivDiameter','Eccentricity','MajorAxisLength',
13 'MinorAxisLength','BoundingBox');
14 Centroid = stats.Centroid;
15 Perimeter = stats.Perimeter;
16 Area = stats.Area;
17 Circularity=(Area)./(Perimeter.^2);
18 ecc = stats.Eccentricity;
19 ext=stats.MajorAxisLength;
20 diameters = sqrt(4 * [stats.Area] / pi);
21 radius = diameters/2;
22 widths = [stats.BoundingBox(3)];
23 heights = [stats.BoundingBox(4)];
24 boundary= B{1};
25 CircleMetricMat = 1./((Perimeter.^2)./(4*pi*Area));
26 sq = diff(boundary).^2;
27 pre = sum(sqrt(sum(sq,2)));
28 metric = 4*pi*Area/pre^2;
29 osobine=struct('ex',{ext}, 'elipsi',{ecc}, 'centroid',{Centroid},
30 'obim',{Perimeter}, 'povrsina',{Area},
  'circularity',{Circularity}, 'precnik',{diameters},
  'poluprecnik',{radius}, 'sirina',{widths}
  , 'duzina',{heights}, 'kruznaMetrika',{metric});

```

Ekstrakcija značajki boje u MATLABU je prikazana kodom ispod.

Program A.4: Ekstrakcija boje

```

1 function [tempBin,rgb]=kmeansBrzi(I,N)
2 [label_im,vec_mean] = kmeans_fast_Color(I,N,1);
3 images = cell(1,N);
4 rgb_label_im = repmat(label_im,[1 1 3]);
5 mr=[];mg=[];mb=[];
6 for i=1:N
7   pom=I;
8
9   pom(rgb_label_im~=i)=0;
10  images{i} = pom;
11  pom=images{i};
12  pom2=pom(:,:,1);
13  pom3=pom(:,:,2);
14  pom4=pom(:,:,3);
15  mr=[mr mean(pom2(pom2~=0))];
16  mg=[mg mean(pom3(pom2~=0))];
17  mb=[mb mean(pom4(pom2~=0))];
18
19 temp = rgb_label_im;

```

```

20 temp(temp~=i) = 0;
21 tempBin=im2bw(temp);
22
23 rgb=[mr;mg;mb];
24 end
25 img=images{2};
26
27 %provjera koja od dvije slike je badem , a koja podloga
28 if(sum(rgb(:,1)>190)>2)
29     rgb=rgb(:,1);
30 elseif(sum(rgb(:,2)>190)>2)
31     rgb=rgb(:,2);
32 else
33 if( (rgb(3,1) > rgb(3,2)))
34     rgb=rgb(:,2);
35 else
36     rgb=rgb(:,1);
37 end
38 end

```

U okviru klasifikacijskih algoritama implementirane su mašine vektorske podrške i neuronske mreže u MATLABU i OpenCV. Ukupno šest algoritama čiji je princip detaljno objašnjen u radu. Obzirom na obim kodova u oba programska jezika u nastavku su dati isječci kodova.

Kod ispod prikazuje princip višeklasne klasifikacije po principu "jedan naspram svih".

Program A.5: One vs All SVM princip u MATLABU

```

1 function [result] = multisvm(TrainingSet,GroupTrain,TestSet)
2 u=unique(GroupTrain);
3 numClasses=length(u);
4 result = zeros(length(TestSet(:,1)),1);
5 res=[];
6 %izgradnja modela
7 for k=1:numClasses
8     G1vAll=(GroupTrain==u(k));
9     models(k) = svmtrain(TrainingSet,G1vAll);
10 end
11
12 %klasifikacija testnih slučajeva
13 for j=1:size(TestSet,1)
14     for k=1:numClasses
15         if(svmclassify(models(k),TestSet(j,:)))
16             break;
17
18         end
19     end
20     result(j) = k;
21
22 end

```

Program A.6: Isječak koda za SVM princip *One Vs One* u programskom paketu MATLAB

```

1 svm_s=cell(1,55);
2 br = 0;
3 N = max(size(trainf))/11;
4 for i=1:11

```

```

5   for j=i+1:11
6     if(i~=j)
7       br1= i * 10;
8       br2 = j * 10;
9       k=[find(train_class_color==br1)
10      find(train_class_color==br2)];
11      svm_label = train_class_color(k);
12      if( i == 2 && j == 5)
13        svm_data = trainf(k,1:3);
14      elseif( i== 2 && j == 7)
15        svm_data = trainf(k,1:3);
16      elseif( i== 2 && j == 10)
17        svm_data = trainf(k,1:3);
18      elseif( i== 5 && j == 7)
19        svm_data = trainf(k,1:3);
20      elseif( i== 5 && j == 10)
21        svm_data = trainf(k,1:3);
22      elseif( i== 7 && j == 10)
23        svm_data = trainf(k,1:3);
24      else
25        if(j==3 || j==4 || j==6 || j==8 || j==9) %boja
26          svm_data = trainf(k,1:3);
27        elseif(j==2 || j==5 || j==7 || j==10 || j==11)
28          svm_data = trainf(k,4:11);
29        end
30      end
31      svm = svmtrain(svm_data,svm_label);
32      br=br+1;
33      svm_s{br} = svm;
34    end
35  end
36
37 br=0; uk_broj_tac = 0;
38 mat = zeros(max(size(test_class_color)),11);
39 mat_pojavljivanja = zeros(max(size(test_class_color)),55);

```

Kreiranje i treniranje SVM i neuronskih mreža u OpenCV.

Program A.7: SVM i NN za oguljene i neoguljene bademe u OpenCV

```

1 vecfind(train_class_color, train_pom, 2, f_ind);
2
3 /* Kako se SVM proslijeduju nizovi, izvrsava se konverzija.
4 * U sustini se vektor pom (sa 2 i 0) pretvara u niz,
5 * koji predstavlja labele. Iz featura se uzimaju R i G komponenta.*/
6 int* pom_ar = new int[ar_size];
7 float* pom_ar_ann = new float[ar_size];
8 vec2array(train_pom, pom_ar, pom_ar_ann);
// Kreiranje i treniranje SVM-a
10
11 Mat trainData_og = vec2Mat(train_color_feature, 0, 1);
12 Mat train_clas_mat_og(train_num * 11, 1, CV_32S, pom_ar);
13 Mat train_clas_mat_ogl(train_num * 11, 1, CV_32FC1, pom_ar_ann);
14 //postavljanje parametara SVM
15 svm_og->setKernel(ml::SVM::POLY);
16 svm_og->setDegree(4);
17 svm_og->setGamma(3);
18 svm_og->setCoef0(0);

```

```
19 svm_og->setTermCriteria(TermCriteria(TermCriteria::MAX_ITER, 1000,
20 1e-12));
//treniranje SVM-a
21 svm_og->train(trainData_og, ml::ROW_SAMPLE, train_clas_mat_og);
22 cout << "Kreiran SVM Oguljeni" << endl;
//postavljanje parametara neuronske mreze oguljenih i neoguljenih
23 //badema
24 Mat<int> layers(4, 1);
25 layers(0) = 2;
26 layers(1) = 100;
27 layers(2) = 100;
28 layers(3) = 1;
29 ann_og->setLayerSizes(layers);
30 ann_og->setActivationFunction(ml::ANN_MLP::SIGMOID_SYM, 1, 1);
31 ann_og->setTermCriteria(TermCriteria(TermCriteria::MAX_ITER +
32 TermCriteria::EPS, 1000, 1e-15));
33 ann_og->setTrainMethod(ml::ANN_MLP::RPROP, 0.07, .75);
\\treniranje podataka sa neuronskom mrezom
34 ann_og->train(trainData_og, ml::ROW_SAMPLE, train_clas_mat_og1);
35 cout << "Kreirana NN Oguljeni" << endl;
36
```

Literatura

- [1] Yang, M., Kpalma, K., Ronsin, J., “A survey of shape feature extraction techniques”, 2008.
- [2] Sonka, M., Hlavac, V., Boyle, R., Image processing, analysis, and machine vision. Cengage Learning, 2014.
- [3] Dougherty, G., Pattern recognition and classification: an introduction. Springer Science & Business Media, 2012.
- [4] Medjahed, S. A., “A comparative study of feature extraction methods in images classification”, International Journal of Image, Graphics and Signal Processing, Vol. 7, No. 3, 2015, str. 16.
- [5] da Fona Costa, L., Cesar Jr, R. M., Shape classification and analysis: theory and practice. CRC Press, 2009.
- [6] Ding, K., Gunasekaran, S., “Shape feature extraction and classification of food material using computer vision”, Transactions-american society of agricultural engineers, Vol. 37, 1994, str. 1537–1537.
- [7] Mohri, M., Rostamizadeh, A., Talwalkar, A., Foundations of machine learning. MIT press, 2012.
- [8] Krig, S., “Image pre-processing”, in Computer Vision Metrics. Springer, 2016, str. 35–74.
- [9] Du, C.-J., Sun, D.-W., “Recent developments in the applications of image processing techniques for food quality evaluation”, Trends in food science & technology, Vol. 15, No. 5, 2004, str. 230–249.
- [10] Kaur, H., Singh, B., “Classification and grading rice using multi-class svm”, International Journal of Scientific and Research Publications, Vol. 3, No. 4, 2013, str. 1–5.
- [11] Nixon, M. S., Aguado, A. S., Feature extraction & image processing for computer vision. Academic Press, 2012.
- [12] Du, C.-J., Sun, D.-W., “Learning techniques used in computer vision for food quality evaluation: a review”, Journal of food engineering, Vol. 72, No. 1, 2006, str. 39–55.
- [13] Novini, A. R., Fundamentals of machine vision component selection. Society of Manufacturing Engineers, 1985.
- [14] Somatilake, S., Chalmers, A., “An image-based food classification system”, in Image and Vision Computing New Zealand. Citeseer, 2007, str. 260–265.

- [15] Ross, L., Russ, J. C., “The image processing handbook”, Microscopy and Microanalysis, Vol. 17, No. 5, 2011, str. 843.
- [16] Torras, C., Computer vision: theory and industrial applications. Springer Science & Business Media, 2012.
- [17] Cheriet, M., Said, J. N., Suen, C. Y., “A recursive thresholding technique for image segmentation”, IEEE transactions on image processing, Vol. 7, No. 6, 1998, str. 918–921.
- [18] Dehariya, V. K., Shrivastava, S. K., Jain, R., “Clustering of image data set using k-means and fuzzy k-means algorithms”, in Computational Intelligence and Communication Networks (CICN), 2010 International Conference on. IEEE, 2010, str. 386–391.
- [19] Omanovic, S., Prepoznavanje oblika i obrada slike. Nacionalana i univerzitetska biblioteka BiH Sarajevo, 2016.
- [20] Harrag, A., Mohamadi, T., “Pca, sfs or lda: What is the best choice for extracting speaker features”, Int. J. Comput. Appl, Vol. 15, No. 3, 2011, str. 1–3.
- [21] Jain, A. K., Duin, R. P. W., Mao, J., “Statistical pattern recognition: A review”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, No. 1, 2000, str. 4–37.
- [22] Guyon, I., Elisseeff, A., “An introduction to variable and feature selection”, Journal of machine learning research, Vol. 3, No. Mar, 2003, str. 1157–1182.
- [23] Shlens, J., “A tutorial on principal component analysis”, arXiv preprint arXiv:1404.1100, 2014.
- [24] Jin, F., Qin, L., Jiang, L., Zhu, B., Tao, Y., “Novel separation method of black walnut meat from shell using invariant features and a supervised self-organizing map”, Journal of Food Engineering, Vol. 88, No. 1, 2008, str. 75–85.
- [25] Szeliski, R., Computer vision: algorithms and applications. Springer Science & Business Media, 2010.
- [26] Choudhary, D., Singh, A. K., Tiwari, S., Shukla, V., “Performance analysis of texture image classification using wavelet feature”, International Journal of Image, Graphics and Signal Processing, Vol. 5, No. 1, 2013, str. 58.
- [27] Bosch, A., Zisserman, A., Munoz, X., “Representing shape with a spatial pyramid kernel”, in Proceedings of the 6th ACM international conference on Image and video retrieval. ACM, 2007, str. 401–408.
- [28] Mery, D., Filbert, D., “Classification of potential defects in automated inspection of aluminium castings using statistical pattern recognition”, in Proceedings of 8th European Conference on Non-Destructive Testing (ECNDT 2002), Jun, 2002, str. 17–21.
- [29] Fitzgibbon, A., Pilu, M., Fisher, R. B., “Direct least square fitting of ellipses”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 21, No. 5, 1999, str. 476–480.

- [30] Stromberg, W. D., Farr, T. G., "A fourier-based textural feature extraction procedure", IEEE Transactions on geoscience and remote sensing, No. 5, 1986, str. 722–731.
- [31] Zhang, G., Ma, Z.-m., Niu, L.-q., Zhang, C.-m., "Modified fourier descriptor for shape feature extraction", Journal of Central South University, Vol. 19, No. 2, 2012, str. 488–495.
- [32] Kumar, A., Pang, G. K., "Defect detection in textured materials using gabor filters", IEEE Transactions on industry applications, Vol. 38, No. 2, 2002, str. 425–440.
- [33] Ojala, T., Pietikainen, M., Maenpaa, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", IEEE Transactions on pattern analysis and machine intelligence, Vol. 24, No. 7, 2002, str. 971–987.
- [34] Kabbai, L., Azaza, A., Abdellaoui, M., Douik, A., "Image matching based on lbp and sift descriptor", in Systems, Signals & Devices (SSD), 2015 12th International Multi-Conference on. IEEE, 2015, str. 1–6.
- [35] Saeed, U., "Automatic recognition of handwritten arabic text: A survey", Life Science Journal, Vol. 11, No. 3s, 2014.
- [36] Donis-González, I. R., Guyer, D. E., Leiva-Valenzuela, G. A., Burns, J., "Assesment of chestnut (castanea spp.) slice quality using color images", Journal of Food Engineering, Vol. 115, No. 3, 2013, str. 407–414.
- [37] Entezari-Maleki, R., Rezaei, A., Minaei-Bidgoli, B., "Comparison of classification methods based on the type of attributes and sample size", Journal of Convergence Information Technology, Vol. 4, No. 3, 2009, str. 94–102.
- [38] Kotsiantis, S. B., Zaharakis, I., Pintelas, P., "Supervised machine learning: A review of classification techniques", 2007.
- [39] He, H., Kong, F., Tan, J., "Dietcam: Multiview food recognition using a multikernel svm", IEEE journal of biomedical and health informatics, Vol. 20, No. 3, 2016, str. 848–855.
- [40] Dzonko, D., Skladište podataka i analize podataka. Univerzitet u Sarajevu, 2013.
- [41] Cortes, C., Vapnik, V. N., "Support-vector networks", Mach. Learn., Vol. 20, No. 3, Sep. 1995, str. 273–297.
- [42] Vapnik, V. N., Lerner, A. Y., "Pattern recognition using generalized portraits", Automation and Remote Control, Vol. 24, 1963, str. 774-780.
- [43] Thomé, A. C. G., "Advances in character recognition", 2012.
- [44] Boser, B. E., Guyon, I. M., Vapnik, V. N., "A training algorithm for optimal margin classifiers", in Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ser. COLT '92. New York, NY, USA: ACM, 1992, str. 144–152, dostupno na: <http://doi.acm.org/10.1145/130385.130401>
- [45] Weston, J., Watkins, C., "Multi-class support vector machines", 1998.
- [46] Kriesel, D., A Brief Introduction to Neural Networks, 2007, dostupno na: available at <http://www.dkriesel.com>

- [47] McCulloch, W. S., Pitts, W., “Neurocomputing: Foundations of research”, 1988, ch. A Logical Calculus of the Ideas Immanent in Nervous Activity, str. 15–27.
- [48] Graves, A., Schmidhuber, J., “Offline handwriting recognition with multidimensional recurrent neural networks”, 2009.