



UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET



Stevica S. Cvetković

**AUTOMATSKA KLASIFIKACIJA SLIKA
ZASNOVANA NA FUZIJI DESKRIPTORA I
NADGLEDANOM MAŠINSKOM UČENJU**

DOKTORSKA DISERTACIJA

Niš, 2018.



UNIVERSITY OF NIŠ
FACULTY OF ELECTRONIC ENGINEERING



Stevica S. Cvetković

**AUTOMATIC IMAGE CLASSIFICATION
BASED ON DESCRIPTOR FUSION AND
SUPERVISED MACHINE LEARNING**

DOCTORAL DISSERTATION

Niš, 2018.

Podaci o doktorskoj disertaciji

Mentor: Dr Saša V. Nikolić, vanredni profesor,
Univerzitet u Nišu, Elektronski fakultet

Naslov: Automatska klasifikacija slika zasnovana na fuziji deskriptora i nadgledanom mašinskom učenju

U ovoj disertaciji su razmatrane mogućnosti za fuziju, tj. kombinovanje različitih vrsta deskriptora sa ciljem povećanja tačnosti i efikasnosti rezultata klasifikacije slika. Analizirane su različite tehnike fuzije deskriptora boje i teksture, uključujući dva pristupa – ranu fuziju i kasnu fuziju. Rana fuzija se obavlja kombinovanjem deskriptora već prilikom njihovog izračunavanja, dok kasna fuzija podrazumeva kombinovanje rezultata klasifikacije obavljene primenom nezavisnih deskriptora. U disertaciji je predložen jedan algoritam za izračunavanje kompaktnog deskriptora slike kroz ranu fuziju robusnih deskriptora teksture i boje. Eksperimentalna evaluacija je pokazala da predložena metoda postiže odličan kompromis između brzine izračunavanja i tačnosti klasifikacije.

Prilikom istraživanja metoda kasne fuzije, akcenat je bio na primeni veštačkih neuronskih mreža i na nedavno uvedenoj metodi za ekstremno brzo treniranje neuronskih mreža, pod nazivom ELM (engl. Extreme Learning Machines). Trenutno glavni ograničavajući faktori za širu primenu ELM su nedovoljna stabilnost i ograničena tačnost ovih klasifikatora. U cilju prevazilaženja ovih problema, predložena je tehnika za kombinovanje rezultata više ELM-a u jedinstveni klasifikator, zasnovana na pravilu sumiranja verovatnoća. Objedinjeni skup ELM klasifikatora je prilikom evaluacije pokazao značajno unapređenje tačnosti i stabilnosti rezultata klasifikacije u odnosu na individualni ELM.

Sa ciljem dodatnog unapređenja tačnosti klasifikacije, predložena je nova hijerarhijska metoda za kombinovanje više komplementarnih deskriptora kroz primenu kasne fuzije ELM klasifikatora. Predložena metoda, u prvom nivou obavlja treniranje zasebnih ELM klasifikatora na osnovu jedne vrste deskriptora slike, dok se klasifikator u drugom nivou trenira da nauči optimalnu kombinaciju deskriptora za svaku od klasa. Ovakav pristup omogućava da se iz skupa većeg broja deskriptora, za svaku od klasa automatski odaberu oni deskriptori koji sadrže najvažniju informaciju za konkretnu sliku, tj. na optimalan način reprezentuju klasu. Komparativna evaluacija je pokazala izuzetno visoku tačnost rezultata klasifikacije slika, koja je u rangu sa vrhunskim rezultatima iz postojeće literature.

Naučna oblast: Elektrotehničko i računarsko inženjerstvo

Naučna disciplina: Digitalna obrada slike

Ključne reči:	Klasifikacija slika, mašinsko učenje, neuronske mreže, ELM, deskriptori slike, fuzija deskriptora.
UDK:	(621.391+004.383.3):004.032.26
CERIF klasifikacija:	T170 Elektronika
Tip licence Kreativne zajednice:	CC BY-NC-ND

Data on Doctoral Dissertation

Doctoral Supervisor:	Dr Saša V. Nikolić, associate professor, University o Nis, Faculty of Electronic Engineering
Title:	Automatic image classification based on descriptor fusion and supervised machine learning
Abstract:	<p>This thesis investigates possibilities for fusion, i.e. combining of different types of image descriptors, in order to improve accuracy and efficiency of image classification. Broad range of techniques for fusion of color and texture descriptors were analyzed, belonging to two approaches – early fusion and late fusion. Early fusion approach combines descriptors during the extraction phase, while late fusion is based on combining of classification results of independent classifiers. An efficient algorithm for extraction of a compact image descriptor based on early fusion of texture and color information, is proposed in the thesis. Experimental evaluation of the algorithm demonstrated a good compromise between efficiency and accuracy of classification results.</p> <p>Research on the late fusion approach was focused on artificial neural networks and a recently introduced algorithm for extremely fast training of neural networks denoted as Extreme Learning Machines - ELM. Main disadvantages of ELM are insufficient stability and limited accuracy of results. To overcome these problems, a technique for combining results of multiple ELM-s into a single classifier is proposed, based on probability sum rules. The created ensemble of ELM-s has demonstrated significant improvement of accuracy and stability of results, compared with an individual ELM.</p> <p>In order to additionally improve classification accuracy, a novel hierarchical method for late fusion of multiple complementary descriptors by using ELM classifiers, is proposed in the thesis. In the first phase of the proposed method, a separate ensemble of ELM classifiers is trained for every single descriptor. In the second phase, an additional ELM-based classifier is introduced to learn the optimal combination of descriptors for every category. This approach enables a system to choose those descriptors which are the most representative for every category. Comparative evaluation over several benchmark datasets, has demonstrated highly accurate classification results, comparable to the state-of-the-art methods.</p>
Scientific Field:	Electronic and computer engineering
Scientific Discipline:	Digital image processing
Key Words:	Image classification, machine learning, neural networks, ELM, image descriptors, descriptor fusion.

UDC: (621.391+004.383.3):004.032.26

CERIF
Classification: T170 Electronics

Creative
Commons
License Type: **CC BY-NC-ND**

Mojoj porodici

Zahvaljujem se kolegama i prijateljima Nemanji Grujiću i dr Milošu Stojanoviću, sa kojima sam aktivno sarađivao tokom istraživanja i realizacije ove doktorske disertacije.

Zahvaljujem se dr Slobodanu Iliću i dr Goranu Stančiću na saradnji i brojnim savetima prilikom istraživanja.

Zahvaljujem se mentoru, prof. dr Saši V. Nikoliću, koji mi je omogućio i pružio podršku za rad na disertaciji.

Najveću zahvalnost dugujem mojoj porodici, bez čije podrške i ljubavi ova disertacija ne bi bila moguća.

SADRŽAJ

1	UVOD.....	10
1.1	Kompleksnost problema automatske klasifikacije slika	13
1.2	Tema i doprinos disertacije	14
1.3	Organizacija disertacije	17
2	AUTOMATSKA KLASIFIKACIJA SLIKA	19
2.1	Predstavljanje digitalne slike.....	19
2.2	Vizuelni deskriptori slike	20
2.3	Postupak izračunavanja deskriptora slike	21
2.4	Postupak automatske klasifikacije primenom nadgledanog mašinskog učenja .	23
2.4.1	Faza treniranja (obučavanja).....	23
2.4.2	Faza testiranja	24
2.5	Metodologija testiranja.....	25
2.6	Kolekcije slika za evaluaciju.....	27
2.7	Pregled literature	31
2.8	Trendovi istraživanja u oblasti automatske klasifikacije slika.....	33
3	DESKRIPTORI SLIKE	34
3.1	Deskriptori teksture	34
3.1.1	Histogram ivica slike – EHD	34
3.1.2	Lokalni Binarni Uzorci – LBP	36
3.1.3	Binarni Gaborovi Uzorci – BGP.....	38
3.1.4	SIFT deskriptor	41
3.2	Deskriptori boje.....	42
3.2.1	Deskriptor rasporeda boje – CLD	42
3.2.2	Histogram boje.....	43
4	NADGLEDANO MAŠINSKO UČENJE ZA KLASIFIKACIJU SLIKA	45
4.1	Teorijske osnove nadgledanog mašinskog učenja	45
4.2	Minimizacija funkcije greške	48
4.2.1	Tehnike treniranja (obučavanja) modela	49

4.2.2	Problem prevelikog ili nedovoljnog podešavanja modela	52
4.3	Klasifikacija	54
4.4	Binarna klasifikacija.....	54
4.4.1	Probabilistička interpretacija logističke regresije	56
4.5	Višeklasna klasifikacija.....	57
4.6	Veštačke neuronske mreže	58
4.6.1	Treniranje (obučavanje) neuronske mreže propagacijom unazad.....	63
4.7	Metoda ekstremno brzog učenja (ELM)	65
5	RANA FUZIJA DESKRIPTORA	70
5.1	Postupak rane fuzije deskriptora	70
5.2	Tehnike normalizacije podataka.....	71
5.2.1	Evaluacija tehnika normalizacije deskriptora	73
5.3	Predloženi postupak rane fuzije deskriptora	74
5.3.1	Eksperimentalna evaluacija	77
6	KASNA FUZIJA KLASIFIKATORA	80
6.1	Postupak kasne fuzije klasifikatora	80
6.2	Agregacija rezultata klasifikatora statističkim merama	84
6.2.1	Ansambli ELM klasifikatora.....	85
6.2.2	Eksperimenti sa ansamblima ELM klasifikatora	88
6.3	Agregacije rezultata klasifikatora primenom mašinskog učenja.....	94
6.3.1	Predložena hijerarhijska metoda za kasnu fuziju ELM klasifikatora	95
6.3.2	Eksperimentalna evaluacija hijerarhijske fuzije	98
7	ZAKLJUČAK	106
7.1	Buduće istraživanje	107
	LITERATURA.....	109
	SPISAK SKRAĆENICA	119
	SPISAK SLIKA.....	120
	SPISAK TABELA.....	122

1 UVOD

Ekspanzija Internet tehnologija i dostupnost digitalnih kamera prouzrokovali su naglo povećanje količine dostupnih vizuelnih informacija, pre svega digitalne slike i videa. Ilustrativno navodimo da je samo u 2015. godini, preko 25 milijardi slika poslato putem Instagram aplikacije¹. Kao posledica ovakve ekspanzije stvorila se potreba za efikasnom obradom, skladištenjem i pretragom vizuelnih informacija. Usled praktičnih zahteva proisteklih iz realnih aplikacija, automatska klasifikacija slika se pojavila kao suštinski važan korak ka efikasnom pretraživanju velikih baza vizuelnih informacija.

Termin klasifikacija slike se odnosi na postupak kojim se ulaznoj slici pridružuje klasa (tj. kategorija) iz unapred definisanog skupa klasa. Cilj kompjuterskog programa za automatsku klasifikaciju slika je da na osnovu analize vizuelnog sadržaja slike izračuna semantičku kategoriju kojoj slika pripada (npr. automobili, životinje, ljudi, itd.). Praktično to znači da za ulaznu sliku treba odrediti oznaku kategorije kojoj ona pripada. Skup klasa u koje se klasifikuje slika zavisi od potreba korisnika klasifikacije. Ipak, klase ne mogu biti proizvoljne, već moraju biti unapred definisane kroz opis i primere slika za svaku od klasa, tako da proces klasifikacije bude ponovljiv.

Automatska klasifikacija slike pripada široj grupi postupaka koji se u literaturi pojavljuju pod nazivom vizuelno prepoznavanje (engl. visual recognition). Osim klasifikacije slike, razmatraju se i dva veoma slična problema: detekcija objekta (engl. object detection) i prepoznavanje objekata (engl. object recognition). Detekcija objekta u slici podrazumeva pretraživanje sadržaja slike (ili više slika) sa ciljem pronalaženja pozicije zadanog objekta, pri čemu nije unapred poznato da li se taj objekat uopšte nalazi u slikama. Prilikom ovog postupka, koriste se samo dve kategorije ishoda, koje označavaju da li je objekat detektovan na određenoj poziciji slike ili nije detektovan. Primer ovakvog postupka je detekcija pešaka u slikama snimljenim kamerama koje nadgledaju saobraćaj. Nešto složeniji postupak je prepoznavanje objekata, čiji je zadatak da na slici pronade i odredi tačnu kategoriju (oznaku) svih objekata iz unapred zadanog skupa. Ovde se koristi veći skup objekata koje treba prepoznati i lokalizovati u slici.

¹ <https://www.pingdom.com/2015>

Automatska klasifikacija slika se vrši upoređivanjem kompaktnih vektora obeležja slike koji se izračunavaju za svaku sliku ponaosob. Prvo se izvršavaju algoritmi za digitalnu obradu slike koji na osnovu sadržaja, slici pridružuju odgovarajući vektor obeležja tj. deskriptor slike. Deskriptor slike predstavlja vektor numeričkih vrednosti koji opisuje karakteristiku teksture, boje, ili oblika objekata u slici. Prilikom izračunavanja deskriptora, slika se transformiše (tj. preslikava) u prostor manje dimenzionalnosti, tzv. prostor deskriptora slike. Deskriptor slike treba da predstavlja kompaktnu i robusnu reprezentaciju koja sublimira što više različitih informacija o slici. Npr. deskriptor slike može biti jednostavan vektor koji sadrži informaciju o nekoliko dominantnih boja u slici. Takode, deskriptor može predstavljati rezultat složenih operacija tako da sadrži informaciju o prostornom rasporedu složenih lokalnih obeležja slike (npr. istaknutih ivica ili uglova).

Osim numeričkih vrednosti izračunatih na osnovu vizuelnog sadržaja slike, deskriptori slike u opštem slučaju mogu sadržati i ključne reči (tekst) koji dodatno opisuje sadržaj slike. Ovakve tekstualne informacije o slici najčešće nisu dostupne, pa se uobičajeno koriste isključivo vizuelni deskriptori slike. U ovom radu ćemo podrazumevati da se koriste isključivo vizuelna obeležja za izračunavanje deskriptora slike prilikom klasifikacije, i u nastavku ćemo pod pojmom deskriptor slike, podrazumevati vizuelni deskriptor slike.

Izbor adekvatnog deskriptora slike predstavlja poseban izazov koji je od suštinskog značaja za uspešnost celokupnog postupka klasifikacije. Različite vrste deskriptora mogu sadržati drugačije vizuelne informacije o slici, kao što su boja, tekstura, ili konture objekata. Otuda postoje deskriptori boje, deskriptori teksture, deskriptori oblika, itd. Standardni pristup klasifikaciji slika uobičajeno koristi samo jednu vrstu deskriptora, što dovodi do ograničene tačnosti ovakvih sistema. Tako npr. deskriptori zasnovani samo na boji su u stanju da uspešno klasifikuju slike u dve kategorije – dnevne slike i slike koje su uslikane noću. Međutim, ako želimo da razlikujemo slike automobila od slika motocikala, onda nam deskriptori boje nisu od značaja, jer na osnovu boje ne možemo razlikovati automobile od motocikala. Ovo je dovelo do zaključka da korišćenje jedne vrste deskriptora nije dovoljno, već je neophodno kombinovanje različitih vrsta deskriptora, tj. fuzija deskriptora, kako bi se unapredila tačnost i robusnost sistema za klasifikaciju slika. Razvoj metoda za izračunavanje robusnih deskriptora slike kroz fuziju više vizuelnih obeležja, izuzetno je kompleksan i aktuelan problem u oblasti digitalne obrade slike.

Generalno postoje dva pristupa za fuziju deskriptora slike: rana fuzija (engl. early fusion) i kasna fuzija (engl. late fusion). Rana fuzija podrazumeva da se deskriptori kombinuju već u

fazi izračunavanja deskriptora, gde se jednostavnim nadovezivanjem ili linearnim kombinovanjem kreira novi višedimenzionalni deskriptor. Kod pristupa sa kasnom fuzijom, klasifikacija slike se inicijalno obavlja sa svakim deskriptorom zasebno, da bi se potom rezultati zasebnih klasifikatora kombinovali u jedinstvenu konačnu odluku. U većini radova u oblasti klasifikacije slika, kombinovanje deskriptora se zasnivalo na ranoj fuziji. Iako ovaj pristup ima prednosti u pogledu jednostavnosti implementacije i brzine izvršenja, novija istraživanja pokazuju da kasna fuzija deskriptora može dovesti do značajnog poboljšanja tačnosti sistema. Zato je izuzetno važno istražiti različite pristupe za fuziju deskriptora, analizirati ih, i uporediti u pogledu tačnosti rezultata klasifikacije, ne zanemarujući kompleksnost izračunavanja kao važan faktor prilikom realne primene. Ovo predstavlja predmet istraživanja i u okviru ove disertacije.

Nakon izračunavanja adekvatnih deskriptora slike, vrši se predviđanje kategorije ulazne slike, na osnovu skupa slika čije su kategorije unapred poznate. Starije metode koje nisu koristile mašinsko učenje, pronalazile su sliku sa najbližijim deskriptorom iz poznatog skupa slika, i njenu kategoriju dodeljivale ulaznoj slici. Problem ovih metoda se ogledao u nemogućnosti da pravilno odrede kategoriju slike u situacijama kada više različitih kategorija sadrži vizuelno slične slike. Takođe, ograničavajući faktor je bila brzina izvršenja u slučaju velikih skupova slika, kakvi se danas sve češće koriste u realnoj primeni. Ovi nedostaci su prevaziđeni primenom nadgledanog mašinskog učenja za predviđanje kategorije slike na osnovu poznatog trening skupa slika.

Osnovu nadgledanog mašinskog učenja predstavlja treniranje (tj. učenje, obučavanje) matematičkog modela koji treba da definiše preslikavanje između slika i njihovih oznaka kategorije tj. klase kojoj slika pripada. Nadgledano učenje matematičkog modela se obavlja na osnovu skupa trening slika čije su kategorije poznate. Nakon definisanja takvog preslikavanja, formirani matematički model se može koristiti za predviđanje kategorije nepoznatih ulaznih slika. Za primenu u klasifikaciji slika, najčešće je korišćena metoda potpornih vektora (engl. Support Vector Machines - SVM) [1–3]. Iako ova metoda postiže relativno dobre rezultate prilikom klasifikacije slika, poslednjih godina se pojavio pristup kojim se paralelno trenira više različitih matematičkih modela nadgledanog mašinskog učenja, i kombinuje u jedinstveni ansambl za rešavanje originalnog problema. Pokazalo se da ovakav pristup ima veću sposobnost generalizacije modela, što povećava tačnost klasifikacije u odnosu na samostalne metode, u različitim oblastima primene. Izbor algoritma za treniranje osnovnog modela, kao i razvoj tehnika za kombinovanje više modela u jedinstveni klasifikator, još uvek predstavlja

izazov među istraživačima u oblasti automatske klasifikacije slika, i stoga je predmet istraživanja i u ovoj disertaciji.

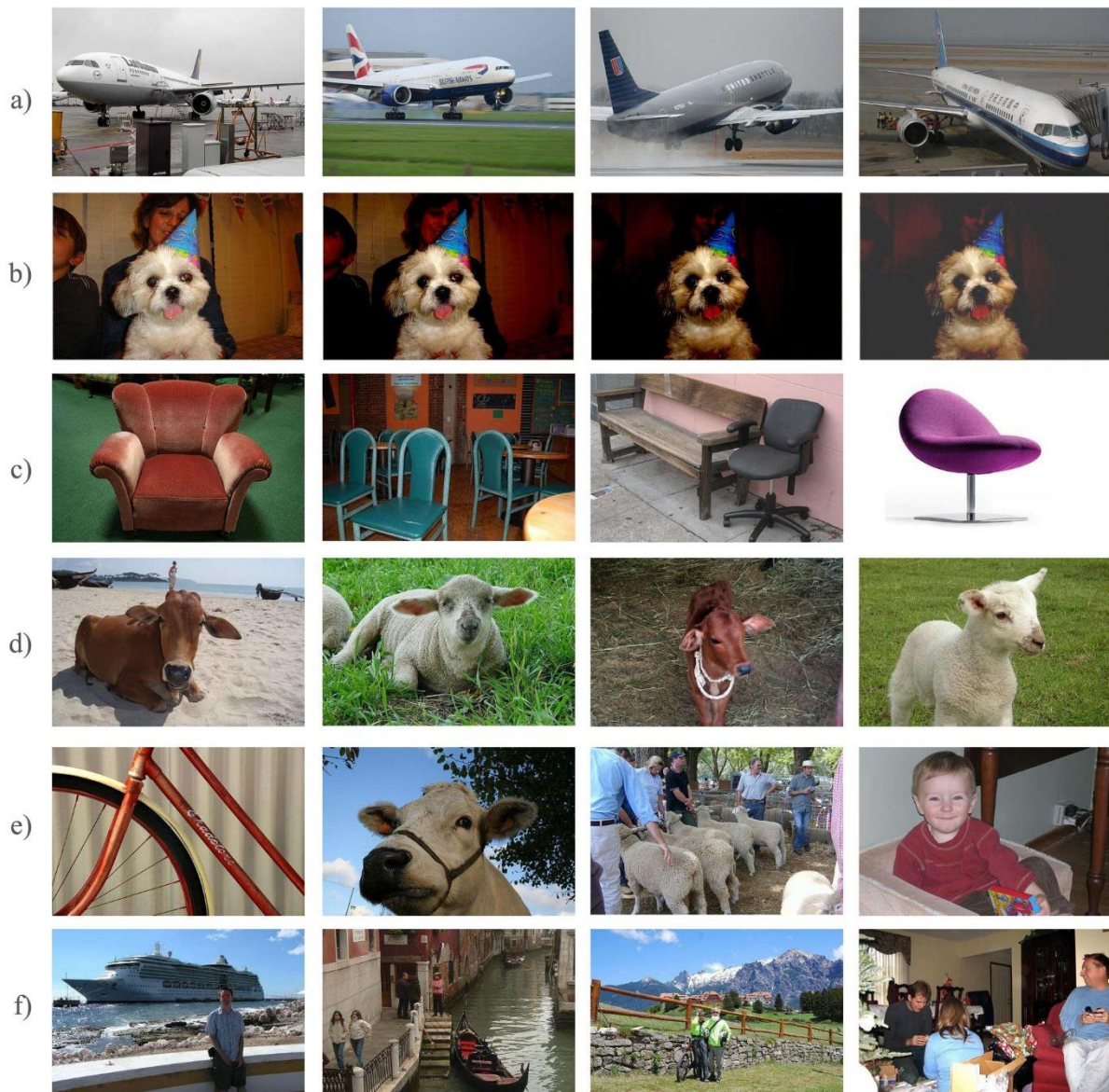
1.1 Kompleksnost problema automatske klasifikacije slika

Automatska klasifikacija slike predstavlja jedan od najkompleksnijih problema u oblasti digitalne obrade slike i kompjuterskog vida. U prilog tome govori činjenica da aktuelni sistemi ne mogu postići čak ni performanse dvogodišnjeg deteta [2]. Bez obzira na svoju složenost, ovo je jedna od najdinamičnijih oblasti istraživanja poslednjih godina, koja postiže ogroman napredak u rezultatima.

Kompleksnost zadatka automatske klasifikacije slika se može sagledati kroz brojne probleme i izazove koje treba rešiti (Slika 1), i koji su opisani u nastavku:

- a) **Različita pozicija kamere** (engl. viewpoint): Objekat ili scena koja se snima podložni su različitim transformacijama usled drugačijeg pozicioniranja kamere koja ih snima (translacija, rotacija, skaliranje, itd.). Ovo može značajno promeniti izgled istog objekta ili scene u različitim slikama (*Slika 1a*).
- b) **Promena osvetljenja scene** (engl. illumination): Promena osvetljenja iste scene može prouzrokovati velike varijacije intenziteta piksela. Ovakve promene piksela mogu biti veoma složene (nelinearne), uz dodatnu kompleksnost koju stvara senka objekta (*Slika 1b*).
- c) **Varijacije unutar iste klase** (engl. intra-class variability): Slike unutar jedne klase mogu biti vizuelno veoma različite, u pogledu boje, teksture, oblika, itd. Npr. slike iz relativno jednostavne klase “stolica”, mogu ispoljiti veliku raznolikost u pogledu izgleda i vrste stolica na slikama, što značajno otežava prepoznavanje te klase objekata (*Slika 1c*).
- d) **Sličnost među različitim klasama** (engl. inter-class similarity): Različite klase mogu sadržati slike koje su veoma slične među sobom. Npr. slike iz klase “krava” i klase “ovca” mogu biti veoma vizuelno slične, što značajno otežava razlikovanje ove dve klase (*Slika 1d*).
- e) **Delimična zaklonjenost objekta** (engl. occlusion): Pojedini delovi objekta mogu biti zaklonjeni (tj. skriveni, nevidljivi) od strane drugih objekata prisutnih u sceni. Osim toga, usled neadekvatnog ugla snimanja, pojedini delovi objekta mogu biti nevidljivi u trenutnoj slici (*Slika 1e*).
- f) **Mnoštvo objekata u sceni** (engl. clutter): U realnim snimcima nije moguće izbeći prisustvo nerelevantnih objekata u sceni, osim glavnog objekta koji je od interesa. Ovo može

predstavljati veliki otežavajući faktor, jer obično nije moguće jasno razgraničiti objekat od interesa, u odnosu na ostatak scene (*Slika 1f*).



Slika 1 Ilustracija glavnih izazova i problema koje treba rešiti prilikom automatske klasifikacije slika: a) različita pozicija kamere u okviru iste klase, b) promena osvetljenja scene, c) varijacije unutar iste klase, d) sličnost među različitim klasama, e) delimična zaklonjenost objekta i f) mnoštvo objekata u sceni. Slike su preuzete iz javno dostupne kolekcije PASCAL VOC 2007 [4].

1.2 Tema i doprinos disertacije

Cilj istraživanja predloženog u ovoj disertaciji je razvoj metoda za efikasnu klasifikaciju slika zasnovanih na fuziji deskriptora i primeni metoda nadgledanog mašinskog učenja, sa ciljem postizanja visoke tačnosti rezultata. Istraživanje je obuhvatilo dve faze postupka za klasifikaciju slika, uključujući izdvajanje deskriptora slike i automatsko predviđanje kategorije.

Za izračunavanje deskriptora slike razmatrane su mogućnosti za fuziju, tj. kombinovanje boje i teksture, kao najvažnijim vizuelnim obeležjima slike. Ove dve vrste deskriptora sadrže komplementarne informacije, a pritom su pokazale visoku uspešnost pri nezavisnom korišćenju za klasifikaciju slika. Istražene su mogućnosti za kombinovanje ova dva komplementarna obeležja u cilju izračunavanja sveobuhvatnog deskriptora slike. Razmatrane su različite tehnike za fuziju deskriptora boje i teksture, uključujući dva pristupa – ranu i kasnu fuziju. Nakon detaljne analize efikasnosti i tačnosti postojećih pristupa, predložen je jedan algoritam za izračunavanje kompaktnog deskriptora slike kroz postupak rane fuzije robusnih deskriptora teksture i boje. Eksperimentalna evaluacija je pokazala da ova metoda postiže odličan kompromis između brzine izračunavanja i tačnosti klasifikacije.

Za predviđanje kategorije slike na osnovu izdvojenih deskriptora razmatrane su mogućnosti za unapređenje postojećih metoda nadgledanog mašinskog učenja. Prvenstveno su istražene metode za kombinovanje više modela mašinskog učenja u jedinstveni klasifikator sa ciljem povećanja tačnosti i efikasnosti klasifikacije. Akcenat istraživanja je bio na veštačkim neuronskim mrežama, zbog njihove sposobnosti modelovanja složenih nelinearnih zavisnosti i zbog jednostavnosti primene za klasifikaciju u više klasa, što su najvažniji zahtevi prilikom klasifikacije slika. Istraživanje u oblasti neuronskih mreža je fokusirano na nedavno uvedenoj metodi za ekstremno brzo treniranje neuronskih mreža, pod nazivom ELM (engl. Extreme Learning Machines). Trenutno glavni ograničavajući faktori za širu primenu ELM metode su nedovoljna stabilnosti i ograničena tačnost ovih klasifikatora. U cilju prevazilaženja ovih problema, predloženo je kreiranje jedinstvenog ansambla koji objedinjuje više ovakvih mreža u jedinstveni klasifikator. Važan problem prilikom kombinovanja klasifikatora je način na koji će se rezultati više klasifikatora kombinovati u jedinstveni rezultat. Sa tim ciljem su detaljno analizirane tehnike kasne fuzije klasifikatora, i predložena je metoda za kombinovanje rezultata više ELM-a zasnovana na pravilu sumiranja verovatnoća. Ova metoda je prilikom eksperimentalne evaluacije pokazala visoku tačnost uz minimalnu računsku složenost.

Prilikom kombinovanja različitih deskriptora primenom više klasifikatora, značajno unapređenje tačnosti je moguće postići uvođenjem jednog dodatnog klasifikatora koji se obučava na osnovu rezultata individualnih klasifikatora. Motivisan ovim relativno novim pristupom, predložen je hijerarhijski metod za kombinovanje više komplementarnih deskriptora kroz primenu kasne fuzije ELM klasifikatora. Predložena arhitektura obezbeđuje da deskriptori koji nose značajniju informaciju za konkretnu sliku, dobiju veći značaj u konačnom rezultatu klasifikacije. Ovakav pristup omogućava da se iz skupa većeg broja deskriptora, za svaku od

klasa automatski odaberu oni deskriptori koji sadrže najvažniju informaciju za konkretnu sliku tj. na optimalan način reprezentuju klasu.

Eksperimentalna evaluacija svih predloženih metoda uz uporednu analizu sa rezultatima iz literature potvrdila je visoku efikasnost, robusnost i tačnost predloženih metoda prilikom automatske klasifikacije slika. Sve predložene metode su softverski implementirane u programskom okruženju Matlab. Prilikom izvođenja eksperimenata, korišćene su standardne javno dostupne kolekcije slika u boji.

Na osnovu teorijskog istraživanja i rezultata eksperimenata koji su detaljno predstavljani u nastavku teksta, najznačajniji doprinosi disertacije se mogu sažeti na sledeći način:

- Pregled i analiza aktuelnih algoritama za fuziju deskriptora slike, u kontekstu primene za klasifikaciju slika.
- Predlog efikasnog algoritma za izračunavanje kompaktnog deskriptora slike baziranog na ranoj fuziji boje i teksture kao najvažnijim vizuelnim karakteristikama slike.
- Sistematska analiza i testiranje uticaja rane fuzije deskriptora na tačnost rezultata klasifikacije nad standardnim kolekcijama slika, i poređenje sa rezultatima nezavisnih deskriptora.
- Pregled i analiza tehnika kasne fuzije primenom nadgledanog mašinskog učenja, sa akcentom na fuziji ELM neuronskih mreža za predviđanje kategorije slike.
- Predlog metode za kasnu fuziju ELM klasifikatora u jedinstveni ansambl, primenom pravila sumiranja verovatnoća.
- Predlog hijerarhijske metode za kombinovanje više komplementarnih deskriptora kroz primenu kasne fuzije ELM klasifikatora, uz značajno unapređenja tačnosti i stabilnosti rezultata klasifikacije.
- Softverska implementacija svih predloženih metoda, uključujući izdvajanje i fuziju deskriptora slike, kao i tehnike kasne fuzije ELM klasifikatora.
- Eksperimentalna evaluacija tačnosti i vremenske kompleksnosti svih razvijenih metoda i poređenje sa aktuelnim metodama za automatsku klasifikaciju slika.

Istraživanje sprovedeno tokom izrade ove disertacije rezultiralo je objavljivanjem značajnog broja naučnih radova čiji su rezultati predstavljeni u daljem tekstu.

U radovima [5], [6] su sistematski analizirane tehnike rane fuzije deskriptora teksture i boje prilikom binarne klasifikacije slika. Zaključeno je da je moguće unaprediti tačnost klasifikacije jednostavnim tehnikama rane fuzije uz adekvatnu normalizaciju raznorodnih

deskriptora. U radu [7] je analiziran uticaj različitih deskriptora slike prilikom postupka generisanja statičkog video pregleda. Predložen je metod koji omogućava brzo generisanje statičkog video pregled visokog stepena vizuelnog kvaliteta. Napredni 3D deskriptori su iskorišćeni u radu [8] za predlog jednog postupka za automatsku rekonstrukciju 3D objekata. Predloženi metod postiže preciznost rekonstrukcije koja je u rangu svetskih rezultata objavljenih u literaturi, uz korišćenje znatno manjeg broja reprezentativnih frejmova u odnosu na konkurenciju.

Primena specifičnog ELM algoritma za treniranje veštačkih neuronskih mreža prilikom višeklasne klasifikacije slika snimljenih iz vazduha, analizirana je u radovima [9], [10]. Pokazan je visok stepen tačnosti klasifikacije uz izuzetno kratko vreme treniranja ELM modela u odnosu na druge metode mašinskog učenja. U radu [11] je predložen jedan algoritam za izračunavanje kompaktnog deskriptora slike kroz postupak rane fuzije robusnih deskriptora teksture i boje. Zatim su detaljno analizirane tehnike fuzije klasifikatora, i predložena je tehnika za kombinovanje rezultata više ELM-a zasnovana na pravilu sumiranja verovatnoća, koja postiže visoku tačnost uz minimalnu računsku složenost. Kombinovanjem predloženog deskriptora sa ansamblima ELM-a, u radu [11] su postignuti unapređeni rezultati u odnosu na konkurentnu literaturu. U radu [12] je predložena tehnika za izračunavanje višekanalnog LBP deskriptora slike koji uključuje i informacije o boji. Obavljena je detaljna analiza uspešnosti klasifikacije ovog deskriptora je u kombinaciji sa ansamblima ELM klasifikatora.

Najznačajniji doprinos disertacije je objavljen u radu [13], gde je predložena hijerarhijska metoda za kombinovanje više komplementarnih deskriptora kroz primenu kasne fuzije ELM klasifikatora. Ovakva metoda omogućava da deskriptori koji nose značajniju informaciju za konkretnu sliku, dobiju veći značaj u konačnom rezultatu predikcije. Na ovaj način će iz skupa različitih deskriptora slike, za svaku od klasa biti odabrani oni deskriptori koji na optimalan način reprezentuju tu klasu. Rezultati eksperimenata su pokazali značajno unapređenje tačnosti klasifikacije u odnosu na korišćenje samostalnih deskriptora ili ranu fuziju deskriptora.

1.3 Organizacija disertacije

Disertacija je u nastavku organizovana na sledeći način:

U Poglavlju 2 su opisani opšti postupci za izračunavanje deskriptora slike i postupci za automatsku klasifikaciju primenom nadgledanog mašinskog učenja. Zatim je opisana metodologija testiranja, uz opis kolekcija slika koje su korišćene prilikom testiranja algoritama predloženih u ovoj disertaciji. Takođe je dat pregled relevantne literature iz oblasti automatske

klasifikacije slika koja je korišćena prilikom istraživanja predstavljenog u ovoj disertaciji. Na kraju se predstavljene trendovi istraživanja u oblasti automatske klasifikacije slika.

Poglavlje 3 detaljno opisuje postupke za izračunavanje deskriptora teksture i boje, čija fuzija će biti razmatrana u postupcima automatske klasifikacije. Ovde su opisani samo reprezentativni postupci za izračunavanje deskriptora slike koji će biti korišćeni u postupcima fuzije predstavljenim u daljem toku disertacije.

U Poglavlju 4 je prvo definisana terminologija i notacija iz oblasti nadgledanog mašinskog učenja. Zatim sledi opis reprezentativnih tehnika nadgledanog mašinskog učenja, uključujući binarnu i višeklasnu klasifikaciju. Poseban akcenat je na opisu veštačkih neuronskih mreža, koje su najviše korišćene prilikom klasifikacije slika u ovoj disertaciji. Na kraju je detaljno opisana efikasna metoda za ekstremno brzo obučavanje neuronskih mreža (engl. Extreme Learning Machines - ELM). Ova metoda je u nastavku disertacija dominantno korišćena u okviru predloženih metoda fuzije.

Poglavlje 5 započinje detaljnim opisom opštih principa rane fuzije deskriptora, uz predstavljanje tehnika normalizacije izračunatih deskriptora. Zatim je opisan algoritam koji je predložen za izračunavanje kompaktnog deskriptora slike kroz ranu fuzije robusnih deskriptora teksture i boje. Na kraju su predstavljeni rezultati eksperimentalne evaluacije predložene metode, uključujući i uporednu analizu sa aktuelnim rezultatima iz literature.

U Poglavlju 6 je prvo opisan opšti postupak kasne fuzije uz razmatranje tehnika za agregaciju rezultata svih klasifikatora, sa akcentom na fuziji ELM klasifikatora. Ovde je predložena efikasna tehnika za kombinovanje rezultata više ELM-a zasnovana na pravilu sumiranja verovatnoća, koja postiže visoku tačnost uz minimalnu računsku složenost. Zatim je predstavljen najznačajniji doprinos ove teze - predlog hijerarhijske metode za kasnu fuziju ELM klasifikatora, koji značajno povećava tačnost klasifikacije prilikom kombinovanja više deskriptora. Na kraju su dati rezultati eksperimentalne evaluacije, uključujući i uporednu analizu sa aktuelnim rezultatima iz literature.

U Poglavlju 7 su predstavljeni najvažniji zaključci disertacije, dobijeni na osnovu teorijske analize i eksperimentalne evaluacije. Na kraju su opisane smernice za buduće istraživanje u oblasti automatske klasifikacije slika.

2 AUTOMATSKA KLASIFIKACIJA SLIKA

2.1 Predstavljanje digitalne slike

Digitalna slika, u najširem smislu predstavlja numeričku reprezentaciju vizuelne stvarnosti, na način koji je pogodan za vizuelni prikaz u ravni. Digitalna slika se sastoji od konačnog broja prostorno-diskretnih elemenata, piksela (engl. pixel - picture element), čija numerička vrednost predstavlja nivo sjajnosti na određenoj lokaciji u slici. Pikel se može definisati kao najmanja komponenta digitalne slike kojoj je moguće dodeliti određenu vrednost. Digitalna slika se matematički predstavlja u obliku matrice, pri čemu jedan piksel ima određene koordinate u matrici i odgovarajući nivo sjajnosti. U daljem tekstu ćemo pod pojmom slika podrazumevati digitalnu sliku.

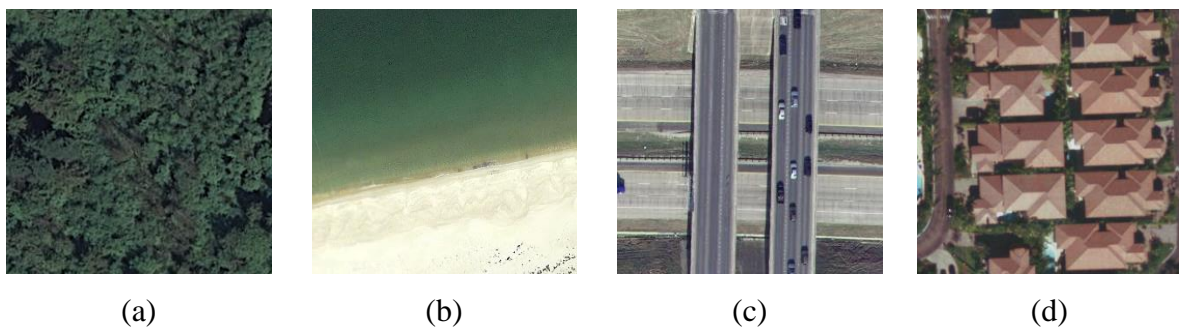
Broj bitova kojima se predstavlja numerička vrednosti piksela, određuje broj mogućih nivoa sjajnosti u slici. Ukoliko se koristi jedan bit po pikselu (1 bpp), njegova numerička vrednost sjajnosti može imati dve različite vrednosti, pa u ovom slučaju kažemo da su slike binarne. U praktičnim aplikacijama, najčešće se koriste tzv. sive slike (monohromatske slike), kod kojih se za predstavljanje jednog piksela koristi 8 bita (8 bpp). Ovakve slike omogućuju da piksel ima jednu od ukupno $256 = 2^8$ dozvoljenih nivoa sjajnosti, iz opsega $[0, 255]$.

Slike u boji se predstavljaju kombinacijom podataka iz različitih opsega vidljivog spektra. U skladu sa percepcijom boje kod čovekovog vizuelnog sistema, najčešće se koriste tri frekvencijska kanala koja odgovaraju crvenoj, zelenoj i plavoj boji. Svaki kanal se predstavlja kao zasebna monohromatska slika, i za ovakve slike kažemo da su predstavljene u RGB prostoru boja (engl. Red, Green, Blue). Kod ovakvih slika u boji, predstavljenih u RGB prostoru, jedan piksel se predstavlja kao uređena trojka, pri čemu svaki element nosi informaciju o intenzitetu piksela u jednom od kanala R, G ili B. Ako pretpostavimo da se svaki kanal boje jednog piksela predstavlja sa 8 bita, broj svih mogućih nijansi boje iznosi $2^8 \times 2^8 \times 2^8 = 2^{24}$. Za ovakve slike se kaže da su 24 bitne i da sadrže približno 16 miliona boja. U zavisnosti od praktičnih potreba i konkretnih aplikacija, mogu se koristiti i drugi prostori boja, kao što su YCbCr ili HSV [14], [15]. U nastavku ćemo za vrednost sjajnosti piksela sive (monohromatske) slike, kao i za intenzitet boje piksela kod slike u boji, koristiti jedinstveni termin - intenzitet piksela.

2.2 Vizuelni deskriptori slike

Da bismo sliku mogli klasifikovati korišćenjem automatskog klasifikatora, potrebno je formirati njenu adekvatnu numeričku reprezentaciju. Intenziteti piksela nisu pogodni kao ulazni podaci klasifikatora zbog velike varijabilnosti na slikama u okviru iste klase, kao i međusobne sličnosti na slikama različitih klasa. Intenziteti piksela su veoma osetljivi na promenu osvetljenja scene, pa će se njihova vrednost razlikovati za isti objekat koji je sniman pod različitim osvetljenjem. Osim toga, vrednosti piksela sadrže visok stepen redundanse i zbog toga nisu efikasni za kompaktno kodovanje sadržaja slike. Kako bi se izbegli pomenuti nedostaci direktnog korišćenja piksela, reprezentacija slike se formira kvantitativnim opisom vizuelnih karakteristika (obeležja) na osnovu kojih je moguće odrediti pripadnost slike nekoj od klasa. Najčešće korišćene vizuelne karakteristike u klasifikaciji slika su boja i tekstura. Dok je za boju kao vizuelnu karakteristika slike intuitivno jasno šta predstavlja, teksturu je znatno teže formalno definisati. U pojedinoj literaturi, tekstura slike se definiše kao informacija o prostornom rasporedu i orijentaciji ivica slike. Pri čemu ivice slike predstavljaju nagle (skokovite) promene intenziteta piksela, uključujući nagle skokove i padove inteziteta.

Uočava se da slike koje pripadaju istoj semantičkoj klasi imaju zajednička vizuelna obeležja. Određene klase objekata imaju zajedničku dominantnu boju, kao npr. snimci šumskog područja imaju dominantno zelenu boju, ili snimci morske obale koji imaju više dominantnih boja - žutu i plavo-zelenu. S druge strane, u određenim klasama prevladavaju pravilne konture (teksture) objekata, npr. slike auto-puteva imaju jasno definisane dugačke prave linije dok slike gusto naseljenih stambenih površina imaju pravougaoni objekte, itd.



Slika 2 Primeri dominantnih vizuelnih obeležja (deskriptora) na slikama snimljenim iz vazduha: (a) šumsko područje - dominantna zelena boja, (b) morska obala – dominantne boje (žuta, zelena, plava), (c) autoput – istaknute paralelne linije (b) gusto naseljeno područje – pravilne konture.

Vizuelne karakteristike (obeležja) se kvantitativno opisuju korišćenjem deskriptora slike. Deskriptori slike imaju za cilj da uproste predstavu slike izvlačeći iz nje najvažnije karakteristike za opis njenog sadržaja. Deskriptori slike su, tipično, višedimenzionalni vektori čiji elementi opisuju raspodelu vizuelnih obeležja slike. Deskriptor slike predstavlja vektor numeričkih vrednosti koji se dobija analizom sadržaja slike na osnovu unapred usvojenog kriterijuma. Tako deskriptor slike može pokazati koliko ima piksela određene boje, ivica određene orijentacije ili nekih drugih obeležja. Deskriptor slike se izračunavaju tehnikama digitalne obrade slike. Najčešće se prvo primenjuje postupak filtriranja slike, nakon čega se za dobijene rezultate filtriranja izračunava globalna statistika ovih vrednosti (npr. histogram). Vizuelna obeležja slike kao što su tekstura ili boja predstavljaju primarne informacije za opis slike, i dominantno se koriste u sistemima za automatsku klasifikaciju slika.

Poželjno je da deskriptor na jedinstven način opisuje sliku, i da se ne menja (ili neznatno menja) prilikom geometrijskih transformacije slike, kao što su translacija ili rotacija. Takođe, deskriptori treba da budu invarijantni u odnosu na promenu osvetljenja i promenu rezolucije slike. U literaturi se pojmovi deskriptor slike i vektor obeležja slike često poistovećuju, pa će i ovoj ovoj disertaciji oba pojma biti ravnopravno korišćena, uz dominantno korišćenje termina deskriptor slike.

2.3 Postupak izračunavanja deskriptora slike

Postupak izračunavanja deskriptora slike se uobičajeno odvija u dve faze koje su opisane u nastavku.

U prvoj fazi se vrši izračunavanje lokalnih deskriptora slike koji opisuju vizuelne karakteristike u okolini jednog piksela (engl. low level features, pixel level features). Ovakvi deskriptori se mogu izračunavati u okolini pravilno raspoređenih tačaka slike dobijenih odmeravanjem sa malim korakom (engl. dense sampling) [16–18]. Korak odmeravanja uobičajeno predstavlja stepen broja 2, tj. iznosi 2, 4, 8 ili 16 piksela po horizontali i vertikali. Druga opcija je da se lokalni deskriptori izračunavaju samo u karakterističnim tačkama slike (engl. keypoints), koje su prethodno određene adekvatnim algoritmom. Karakteristične tačke predstavljaju pozicije u slici koje ostaju uočljive na istoj lokaciji nakon različitih geometrijskih i svetlosnih transformacija. Ovo su uobičajeno uglovi na preseku više istaknutih ivica slike. Razvijen je veliki broj algoritama za određivanje karakterističnih tačaka predstavljenih u literaturi: Harisov detektor uglova [19], SIFT [20], [21], SURF [22], itd. Bez obzira kako su određene pozicije, lokalni deskriptori se izračunavaju nad lokalnim susedstvom piksela na tim

pozicijama (npr. 8×8 , 16×16). Ukoliko je potrebno postići invarijantnost deskriptora u odnosu na rotaciju, prvo se može odrediti dominantna orijentacija ivica u lokalnom susedstvu, a zatim izračunati lokalni deskriptor u odnosu na tu orijentaciju [20]. Druga opcija za postizanje invarijantnosti u odnosu na rotaciju je da se izračuna skup lokalnih deskriptora nad slikama dobijenim nakon što je originalna slika zarotirana više puta za određeni ugao. Ovakav pristup je često primenjen prilikom klasifikacije slika [16], [18].

Druga faza predstavlja računanje globalnog deskriptora cele slike (engl. global image descriptor), koja se obavlja kroz objedinjavanje informacija o prethodno izračunatim lokalnim deskriptorima. Za ovu namenu se najčešće koristi *histogram*, kao statistička mera raspodele verovatnoća pojavljivanja lokalnih deskriptora. Ako su lokalni deskriptori predstavljeni skalarnim vrednostima (npr. LBP [23], BGP [24], itd.), histogram je moguće izračunati direktno. U slučaju višedimenzionalnih lokalnih deskriptora (vektora), celokupan prostor svih mogućih vektora je izuzetno veliki (npr. 128 dimenzija u slučaju SIFT deskriptora [20]). Zato je neophodno obaviti adekvatnu redukciju prostora svih mogućih vektora kroz adekvatnu kvantizaciju. U slučaju klasifikacije slika, dominantno se primenjuje metoda zasnovana na *skupovima vizuelnih reči* (engl. Bag of Words - BoW, Bag of Features - BoF) [25], [26]. Ova metoda je inspirisana tehnikama za prepoznavanje tekstualnih dokumenata u kojima se dokument predstavlja frekvencijama pojavljivanja dominantnih reči, ne uzimajući u obzir njihov prostorni raspored. U slučaju klasifikacije slika, prvo se formiraju tzv. vizuelne reči, koje predstavljaju reprezentativne vektore lokalnih deskriptora dobijene klasterizacijom svih vektora iz trening skupa slika. Na osnovu ovih vizuelnih reči se formira rečnik vizuelnih reči, koje zapravo predstavljaju unapred definisane kodove kojima će biti predstavljen svaki lokalni deskriptor iz slike. U postupku testiranja, za proizvoljne lokalne deskriptore ulazne slike, prvo se obavlja njihovo kodovanje, kroz pronalaženje najslabijih vizuelnih reči u rečniku. Zatim se celokupna slika predstavlja kao histogram pojavljivanja vizuelnih reči. Umesto klasičnog histograma, u novijoj literaturi se koriste naprednije metode objedinjavanja lokalnih deskriptora, koje postižu značajno unapređenje tačnosti klasifikacije. Primeri ovakvih metoda su „sparse“ kodovanje [27], Fišerov vektor [28], Super-vektor [29], itd.

Nedostatak izračunavanja jednog globalnog histograma slike je to što u ovakvom histogramu nije uključena informacija o prostornom rasporedu lokalnih deskriptora, već samo frekvencija (tj. broj) pojavljivanja određenog lokalnog deskriptora u celoj slici. Ovaj nedostatak se može prevazići tako što se histogrami izračunavaju nad regionima (blokovima) slike, nakon čega se svi „blokovski“ histogrami nadovezuju u konačni globalni deskriptor slike [16]. Na

ovaj način je pozicija lokalnog deskriptora određena na nivou bloka kome pripada. Drugi načina za prevazilaženje ovog problema je da se u vektor koji predstavlja lokalni deskriptor uključi i informacija o prostornoj lokaciji tog deskriptora u slici [30].

Pojedine metode vrše dodatno redukovanje dimenzionalnosti globalnih deskriptora slike, kako bi deskriptore učinile efikasnijim za memorisanje i dalju primenu metoda mašinskog učenja. Najčešće primenjivana tehnika za ovu namenu je Analiza glavnih komponenti (engl. Principal Component Analysis – PCA), koja koristi pretpostavku da elementi vektora deskriptora sa većom varijansom nose i veću količinu informacija. Zato se smanjenje dimenzionalnosti vrši odbacivanjem elemenata sa malom varijansom [53], [75]. Na ovaj način je moguće smanjiti dimenzionalnost vektora i po nekoliko puta.

2.4 Postupak automatske klasifikacije primenom nadgledanog mašinskog učenja

Prethodno izračunati deskriptori slike se koriste kao ulazni podaci algoritma za klasifikaciju, koji se uobičajeno skraćeno naziva klasifikator. Automatski klasifikatori slika obavljaju postupak kojim se, na osnovu vizuelnog sadržaja slike (deskriptora), izračunava oznaka klase ulazne slike. Klasifikator se najčešće realizuje primenom tehnika nadgledanog mašinskog učenja. Opšti postupak automatske klasifikacije slika primenom nadgledanog mašinskog učenja uobičajeno sadrži dve faze, koje su detaljno opisane u nastavku.

2.4.1 Faza treniranja (obučavanja)

Ova faza predstavlja pripremu za narednu fazu u kojoj će se obaviti predviđanje klase ulazne slike. U fazi treniranja cilj je da se „nauči“ optimalna matematička funkcija (model) koja na osnovu slike može da predvidi oznaku klase. Ova faza nije kritična u pogledu vremenskih performansi, jer se samo jedanput izvršava prilikom pripreme podataka, a rezultati se kasnije koriste prilikom procesiranja ulaznih test slika. Ova faza uobičajeno obuhvata sledeće korake: a) Kreiranje trening skupa slika, b) Izračunavanje deskriptora za slike iz trening skupa i c) Treniranje klasifikatora nad trening skupom slika.

a) Kreiranje trening skupa slika

Izbor trening slika treba da bude takav da one sadrže što više vizuelnih informacija o klasi kojoj pripadaju. Da bi se obezbedila dovoljna raznolikost trening skupa, treba koristiti što je moguće više relevantnih trening slika koje karakterišu svaku od klasa. Pri tome je dozvoljeno

koristiti različit broj trening slika za svaku od klasa, pri čemu svakoj slici treba pridružiti informaciju (oznaku, labelu) o tome kojoj klasi pripada.

Generalno, tehnike nadgledanog mašinskog učenja zahtevaju velike skupove trening slika, i postižu značajno bolje rezultate korišćenjem velikih skupova koji sadrže stotine hiljada (ili milione) slika. Iako moderni algoritmi nadgledanog mašinskog učenja poseduju visok stepen generalizacije, koji im omogućava da nauče matematičke modele i na osnovu relativno malog broja primera, u praktičnim primenama je preporučljivo koristiti što veći broj i što adekvatnije slike prilikom treniranja. Međutim, u pojedinim praktičnim primenama nije dostupan veliki broj trening slika (npr. termovizijske slike), pa je neophodno analizirati i rezultate kada se za treniranje koristi relativno mali broj primera iz svake od klasa.

b) Izračunavanje deskriptora za slike iz trening skupa

Nakon formiranja trening skupa slika, obavlja se izračunavanje deskriptor vektora za svaku od slika iz tog skupa. Izbor adekvatnog algoritma za izračunavanje deskriptora se vrši na osnovu konkretne aplikacije (npr. medicinske slike, detekcija lica, snimci iz vazduha, itd.). Pre samog izračunavanja deskriptora, opciono je moguće izvršiti pretprocesiranje slike, kao što je konverzija u određeni prostor boja ili normalizacija piksela. Takođe, moguće je obaviti postprocesiranje izračunatih deskriptora kroz postupak redukcije dimenzionalnosti ili kombinovanje (fuziju) više deskriptora jedne slike u novi deskriptor.

c) Treniranje (obučavanje) klasifikatora nad trening skupom slika

Treniranje klasifikatora podrazumeva postupak koji pronalazi optimalnu funkciju (model) koja na osnovu ulazne slike predviđa oznaku njene klase. Ovde se koristi prethodno kreirani trening skup označenih slika, kako bi se odredili parametri funkcije koja preslikava skup deskriptora slika na skup oznaka njihovih klasa. Ova funkcija će se kasnije koristiti za predviđanje klase nove test slike.

2.4.2 Faza testiranja

Ova faza treba da se izvrši u realnom vremenu, i da obavi predviđanje klase ulazne test slike. Uobičajeno se sastoji iz sledećih koraka: a) izračunavanje deskriptora test slike i b) predviđanje klase test slika.

a) Izračunavanje deskriptora test slika

Za novu ulaznu sliku čiju klasu treba odrediti, izračunava se deskriptor slike. Postupak izračunavanja deskriptora mora biti isti kao i za slike iz trening skupa. Npr., ukoliko je izvršena

normalizacija i/ili redukcija dimenzionalnosti deskriptora trening slika, iste transformacije treba primieniti i na deskriptore test slika, kako bi očuvala jedinstvenost reprezentacije.

b) Predviđanje klase test slika

U ovoj fazi se vrši predviđanje oznake klase test slike, na osnovu deskriptora test slike i prethodno treniranog klasifikatora. Trenirani klasifikator predstavlja matematički model tj. funkciju koja deli prostor svih deskriptora na manje podprostore koji pripadaju određenim klasama slika. Predviđanje klase jedne test slike se najčešće svodi na izračunavanje vrednosti funkcije za ulazni deskriptor kao parametar. Nakon što se za svaku test sliku predvidi oznaka klase kojoj pripada, može se proveriti tačnost postupka klasifikacije, tj. izvršiti verifikacija rezultata.

2.5 Metodologija testiranja

U svim narednim eksperimentima, odgovarajuća kolekcija slika se deli se na dva dela: podskup za treniranje i podskup za testiranje. Algoritmi klasifikacije se treniraju korišćenjem primera (slika) iz trening skupa, a zatim se performanse klasifikatora mere nad slikama iz test skupa koje nisu korišćene prilikom treniranja klasifikatora.

Za određivanje optimalnih parametara upotrebljenog algoritma mašinskog učenja, koristi se trening skup podataka, nad kojim se primenjuje tehnika pod nazivom ***unakrsna validacija (engl. cross validation)***. Ova tehnika podrazumeva da se trening skup podataka podeli na k delova približno iste veličine, a zatim se $k - 1$ delova koristi kao skup za učenje parametara a preostali deo kao skup za validaciju. Postupak se ponavlja k puta tako da svaki od delova po jednom učestvuje u ulozi validacionog skupa. Uobičajeno je da se postupak podele na k delova modifikuje tako da se osigura podjednaka zastupljenost klasa u svakom od k delova. Ovakav postupak se naziva stratifikacija i njime bi trebalo obezbediti da zastupljenost klasa bude približno jednaka zastupljenosti u inicijalnom skupu primera. U praksi se najčešće uzima $k = 5$ ili $k = 10$, jer se takva unakrsna validacija pokazala kao dovoljno tačna, pri čemu nije preterano računski zahtevna.

Kao standardna mera uspešnosti klasifikacije, korišćena je ***tačnost klasifikacije (engl. classification accuracy)***. Tačnost klasifikacije se definiše kao broj tačno klasifikovanih slika u odnosu na ukupan broj slika u test skupu, izražen u procentima:

$$ACC = \frac{\text{broj tačno klasifikovanih slika}}{\text{ukupan broj slika test skupa}} \times 100\%$$

Većina javno dostupnih kolekcija slika za evaluaciju klasifikacionih metoda, nije unapred podeljena na trening i test skup, već se podrazumeva slučajna podela u određenom odnosu (npr. 80% slika za treniranje i 20% za testiranje). Kako bi se obezbedilo da izmerena tačnost klasifikacije ne zavisi od konkretne podele dostupnih podataka na trening i test skup, ciklus treniranja i testiranja klasifikatora se ponavlja više puta sa slučajno izabranim podelama. Kao konačni rezultat se navodi srednja vrednost tačnosti klasifikacije, kao i standardna devijacija rezultata klasifikacija dobijenih za različite podele. Ukoliko je standardna devijacija velika, može se zaključiti da testirani algoritam za klasifikaciju značajno zavisi od izbora trening skupa slika. U slučaju male vrednosti devijacije, dobijeni rezultati se smatraju pouzdanim.

Pored tačnosti klasifikacije, kao jedinstvene mere performansi klasifikatora, za klasifikatore koji su pokazali najbolje rezultate moguće je prezentovati i matricu konfuzije (engl. confusion matrix). Matrica konfuzije pokazuje koliko procenata slika iz određene klase je klasifikovano u svaku od postojećih klasa. Na ovaj način se može dobiti uvid u tačnost klasifikacije za svaku od klasa, kao i detektovati klase za koje se najčešće javljaju greške prilikom klasifikacije.

Tabela 1 Primer matrice konfuzije za klasifikaciju u 3 klase.

		Predviđene klase		
		Klasa 1	Klasa 2	Klasa 3
Stvarne klase	Klasa 1	0.75	0.15	0.10
	Klasa 2	0.10	0.85	0.05
	Klasa 3	0.30	0.05	0.65

U prethodnom primeru za klasifikaciju u 3 klase, u matrici konfuzije se uočava da su slike iz Klase 1 ispravno klasifikovane u 75% slučajeva; u 15% slučajeva su pogrešno klasifikovane kao Klasa 2 i u 10% slučajeva pogrešno klasifikovane kao Klasa 3. Uočava se da klasifikacija Klase 3 pokazuje najmanju tačnost, jer je ispravno klasifikovana u 65% slučajeva, dok je čak u 30% slučajeva pogrešno klasifikovana kao Klasa 1. Jasno je da su prilikom analize matrice konfuzije, najvažniji elementi na glavnoj dijagonali, koji zapravo pokazuju procenata tačno klasifikovanih podataka za svaku stvarnu klasu. U slučaju većeg broja klasa, matricu konfuzije je pregledno predstaviti grafički korišćenjem različitih boja. Kada je potrebno izvršiti analizu uticaja određenog parametra na tačnost klasifikacije, uobičajeno se koristi tabelarno ili grafičko predstavljanje tačnosti klasifikacije u odnosu na vrednost parametra.

Da bi se dobijeni rezultati mogli uporediti sa rezultatima iz literature, u eksperimentima su korišćeni isti protokoli testiranja kao i u drugim naučnim radovima koji koriste iste kolekcije slika. Konkretni detalji protokola, veličine trening i test skupova, kao i broj ponavljanja ciklusa, navedeni su u opisima pojedinih kolekcija slika. Pored opisanih protokola testiranja, izvršeni su i eksperimenti u kojima je varirana veličina trening skupa kako bi se dobila zavisnost tačnosti klasifikacije od veličine trening skupa tj. broja trening slika. Svi eksperimenti su ponovljeni po 100 puta korišćenjem različitih podela kolekcija slika na trening i test podskupove, kako bi se postigla robusnost rezultata.

Konkretna implementacija svih predloženih algoritama izvršena je u programskom jeziku MATLAB, uz korišćenje standardnih pomoćnih biblioteka.

2.6 Kolekcije slika za evaluaciju

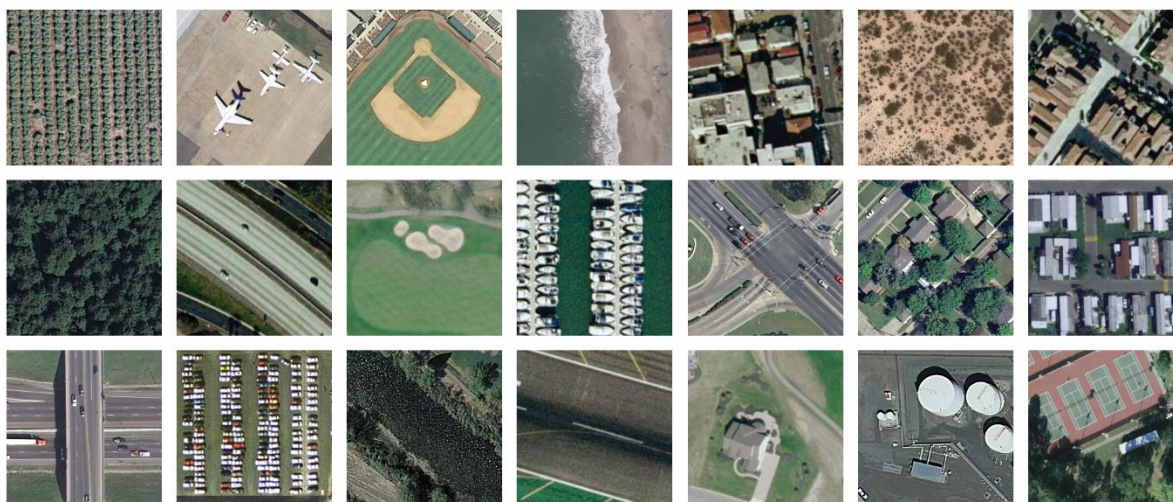
Za eksperimentalnu evaluaciju predstavljenih algoritama fuzije, korišćeno je nekoliko javno dostupnih kolekcija slika. Detaljni opis korišćenih kolekcija slika sledi u nastavku.

1. „Landuse21“ kolekcija slika

Ova kolekcija slika se sastoji od 2100 slika u boji (RGB), rezolucije 256×256 piksela. Landuse21 kolekcija je uvedena u radu [31] za testiranje klasifikatora zasnovanih na SIFT deskriptorima slike [20] i skupovima vizuelnih reči (engl. Bag of Words) [25]. Slike su preuzete iz ortometrijskih aero snimaka prostorne rezolucije 30cm po pikselu, koji su javno dostupni na stranicama USGS². Slike su manuelno grupisane u 21 klasu od po 100 slika i javno su dostupne za preuzimanje³. Korišćene su sledeće klase: poljoprivreda, avioni, bejzbol igralište, plaža, zgrade, prerija, gusto naseljeno područje, šuma, autoput, teren za golf, luka, raskrsnica, naseljeno područje srednje gustine, kamp prikolice, nadvožnjak, parking, reka, aerodromska pista, retko naseljeno područje, cisterne, i teniski tereni. Primere slika iz svake od klasa prikazuje Slika 3.

² United States Geological Survey, <http://www.usgs.gov/>

³ <http://vision.ucmerced.edu/datasets/landuse.html>



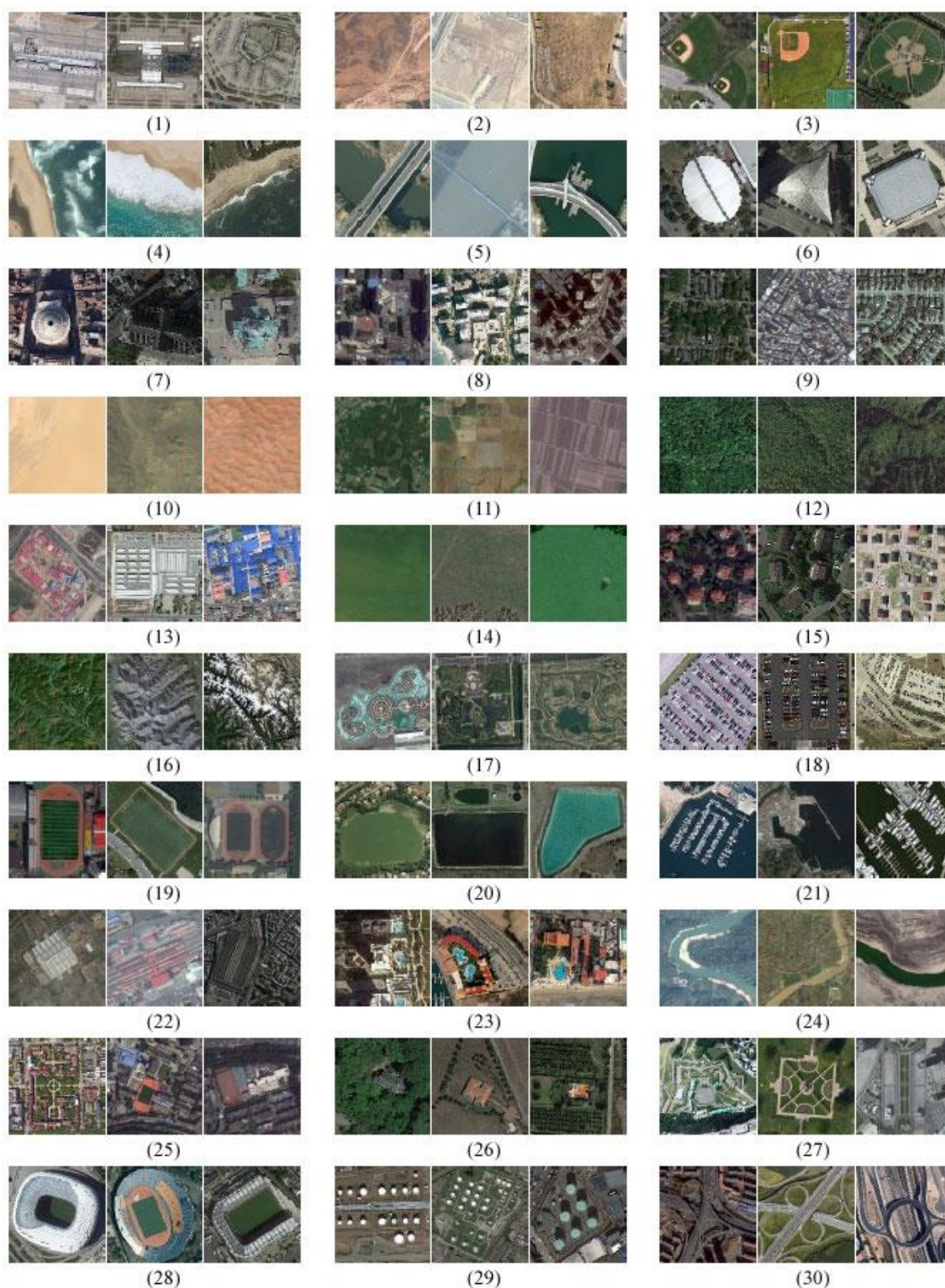
Slika 3 Primeri slika za svaku od 21 klase iz kolekcije „Landuse21” [31]. Prikazane su sledeće klase (u redosledu s leva na desno): poljoprivreda, avioni, bejzbol igralište, morska obala, zgrade, prerija, gusto naseljeno područje, šuma, autoput, teren za golf, luka, raskrsnica, naseljeno područje srednje gustine, kamp prikolice, nadvožnjak, parking, reka, aerodromska pista, retko naseljeno područje, cisterne, i teniski tereni.

Može se uočiti da slike pojedinih klasa imaju veoma sličan vizuelni sadržaj, iako pripadaju različitim klasama. Na primer, sledeće klase sadrže veoma slične slike koje prikazuju stambena naselja sa manjom ili većom gustinom naseljenosti: zgrade, gusto naseljeno zemljište, zemljište srednje gustine naseljenosti i kamp prikolice. Može se zaključiti da Landuse21 predstavlja kompleksnu kolekciju slika koja predstavlja izazov za evaluaciju algoritama automatske klasifikacije slika.

Autori ove kolekcije su koristili po 80 slika iz svake klase za treniranje klasifikatora i vršili testiranje nad preostalim slikama. Ovaj postupak je ponovljen pedeset puta i navedene su srednje vrednosti dobijenih tačnosti klasifikacije. I u ovoj disertaciji je prihvaćena ista metodologija testiranja, s tim što se u eksperimentima dodatno varira broj trening slika, kako bi se analizirao uticaj broja trening slika na tačnost klasifikacije.

2. „AID“ kolekcija slika

Predstavljena je nedavno u radu [27] kao velika kolekcija slika za klasifikaciju scena snimljenih iz vazduha (engl. Aerial Image Dataset - AID).



Slika 4 Primeri po 3 slike za svaku od 30 kategorija iz kolekcije „AID“ [32]: 1) aerodrom, 2) nenaseljeno zemljište, 3) bejzbol igralište, 4) morská obala, 5) most, 6) dvorana, 7) crkva, 8) poslovni objekti, 9) gusto naseljeno područje, 10) pustinja, 11) farma, 12) šuma, 13) industrijski objekti, 14) livada, 15) naseljeno područje srednje gustine, 16) planina, 17) park, 18) parking, 19) sportsko igralište, 20) ribnjak, 21) luka, 22) železnička stanica, 23) odmaralište, 24) reka, 25) škola, 26) gusto naseljeno područje, 27) trg, 28) stadion, 29) silosi, 30) nadvožnjak.

„AID“ je najobimnija i najzahtevnija kolekcija slika korišćena prilikom eksperimenata u ovoj disertaciji. Sadrži 10.000 slika prikupljenih sa Google Earth servisa i klasifikovanih u 30 kategorija. Broj slika po kategoriji se kreće od 220 do 420, pri čemu su sve slike rezolucije 600×600. U odnosu na „Landuse21“ kolekciju, ova kolekcija sadrži znatno veći broj slika i veći broj klasa. Osim toga, varijabilnost slika unutar klasa je znatno izraženija. Sve ovo čini „AID“ kolekciju slika izuzetno zahtevnom za testiranje.

3. Kolekcija slika „8-scenes”

Dodatna kolekcija slika koja je korišćena u eksperimentima je tzv. **8-scenes** kolekcija slika [33]. Sastoji se od 2688 slika u boji, na kojima su fotografije spoljnih scena klasifikovanih u 8 kategorija: obala, planina, šuma, proplanak, kolovoz, gradsko područje, zgrada i autoput. Slika 5 prikazuje primere slika iz svake od klasa ove kolekcije. Za treniranje klasifikatora nad ovom kolekcijom korišćeno je po 100 slika iz svake klase, dok je testiranje obavljeno nad preostalim slikama.

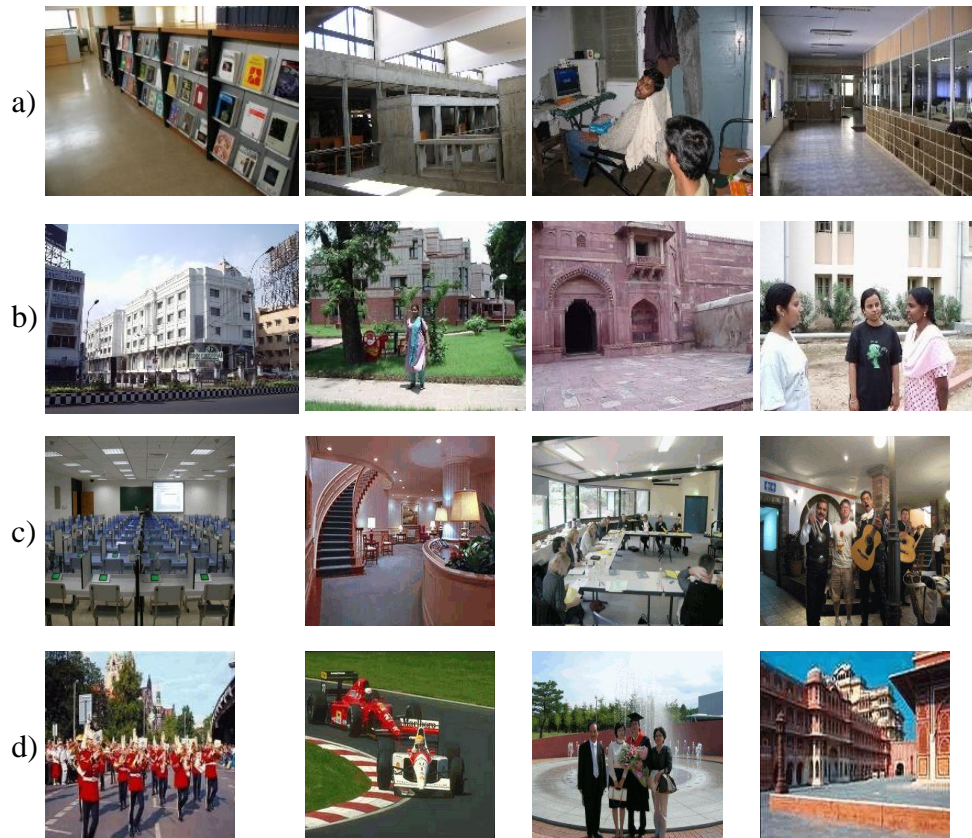


Slika 5 Primeri slika za svaku od 8 klasa iz kolekcije „8-scenes” [33]. Prikazane su klase: obala, šuma, autoput, gradsko područje, planina, proplanak, ulica i zgrada.

4. Kolekcije „IITM-SCID2“ i „COREL-INOUT“

Ove dve kolekcije slika služe za evaluaciju binarnih klasifikatora, što znači da sadrže fotografije koje su klasifikovane u dve klase. U konkretnom slučaju to su klase: a) unutrašnje slike (engl. indoor images) koje su fotografisane u zatvorenim prostorijama i b) spoljašnje slike (engl. outdoor images) koje su slikane u spoljnom prostoru van zatvorenih prostorija. Obe kolekcije pokazuju visok stepen varijacija vizuelnog sadržaja unutar svake od klasa (Slika 6), kao i varijacije u rezoluciji i kvalitetu fotografija. Ovo ih čini veoma zahtevnim u pogledu postizanja visoke tačnosti klasifikacije. Prva kolekcija slika „IITM-SCID2“ [34] sadrži ukupno

902 slike (393+509), sa različitim rezolucijama u opsegu od 80×80 do 2048×1536 . Druga kolekcija „COREL-INOUT“ [35] se sastoji od ukupno 1276 slika iste rezolucije 256×256 . U ovoj kolekciji 630 slika pripada klasi “unutrašnjih slika”, dok je 646 slika iz klase “spoljašnje slike”. Primeri slika iz ovih kolekcija su dati u nastavku.



Slika 6 Primeri slika: a) IITM-SCID2 klasa unutrašnjih slika, b) IITM-SCID2 klasa slika snimljenih spolja, c) COREL-INOUT klasa unutrašnjih slika, d) COREL-INOUT klasa slika snimljenih spolja

2.7 Pregled literature

Tokom poslednjih godina, razvoj računarske tehnologije i povećanje memorijskih resursa omogućili su razvoj složenih algoritama za automatsku klasifikaciju slika.

U prvim značajnim radovima iz automatske klasifikacije slika [33], korišćeni su Gaborovi filtri za izračunavanje prostornog rasporeda globalnih karakteristika teksture, sa ciljem modelovanja sadržaja koji slika prikazuje. U [23] je predstavljen multirezolucioni pristup za izračunavanje lokalnih deskriptora teksture sivih slika koristeći lokalne binarne uzorke (LBP). Ovaj deskriptor je kasnije masovno korišćen u različitim primenama prilikom automatske klasifikacije slika [36]. Kao jedno od uspešnijih unapređenja LBP deskriptora, u radu [37] je

uveden CENTRIST deskriptor, definisan tako da analizira istaknute strukture, kao što su ivice i konture. Definisanje SIFT deskriptora koji je invarijantan u odnosu na promenu rezolucije [20], smatra se radom koji je od ključnog značaja za oblast kompjuterskog vida. SIFT deskriptor je kasnije masovno upotrebljavan prilikom automatske klasifikacije slika [16], [38–40].

Uočljiva je tendencija proširivanja deskriptora definisanih za monohromatske slike i njihova primena na slike u boji i multispektralne slike, sa ciljem unapređenja rezultata klasifikacije. U radu [39] je izvršena relativno jednostavna konkatenacija SIFT deskriptora sivih slika dobijenih iz svih dostupnih kanala, za potrebe klasifikacije scena prirode. U radu [41] je predloženo proširenje CENTRIST deskriptora na slike u boji (mCENTRIST), pri čemu je poseban akcenat stavljen na na tehnikama pretprocesiranja i postprocesiranja deskriptora. Uopredna analiza različitih metoda za klasifikaciju multispektralnih slika je data u [42].

U oblasti automatske klasifikacije slika razmatrane su mogućnosti za efikasnu reprezentaciju prostornog rasporeda deskriptora u slici. U radu [16] je predstavljen je postupak za predstavljanje prostornog rasporeda SIFT deskriptora tako što je analizirana piramidalna reprezentacija slike pri različitim rezolucijama. CENTRIST deskriptor [37], [41] izračunava prostorni raspored lokalnih deskriptora za 31 region slike na različitim nivoima piramidalne dekompozicije, koje kasnije objedinjuje u jedinstveni globalni deskriptor slike.

Dodatna poboljšanja tačnosti klasifikacije su postignuta drugačijom reprezentacijom globalnog deskriptora slike tokom obučavanja klasifikatora. Veoma efikasne metode za reprezentaciju deskriptora slike su predstavljene u radovima [27], [30], [43], uz postizanje značajno unapređenja tačnosti klasifikacionih rezultata.

Najnoviji pristup za automatsku klasifikaciju slika se zasniva na primeni dubokih neuronskih mreža (engl. Deep Learning) tj. konvolucionih neuronskih mreža (engl. Convolutional Neural Networks - CNN) [44], [45]. Za razliku od tradicionalnog pristupa za izračunavanje deskriptora slike, gde su koeficijenti konvolucionih filtara unapred definisani (npr. za izdvajanje ivica slike), konvolucione neuronske mreže vrše učenje (treniranje) koeficijenata većeg broja konvolucionih filtara. Koeficijenti filtara nisu unapred poznati i određuju se u fazi treniranja algoritma, tako da rezultat klasifikacije ima minimalnu grešku na velikom trening skupu slika. S obzirom da su pokazale ogroman potencijal, razmotrane su različite metode korišćenja konvolucionih neuronskih mreža prilikom automatske klasifikacije slika.

2.8 Trendovi istraživanja u oblasti automatske klasifikacije slika

Analizom aktuelne literature može se zaključiti da se trend istraživanja u oblasti automatske klasifikacije kreće u dva smera. Prvi se odnosi na pronalaženje što efikasnijih i efektivnijih deskriptora slike, prilagođenih konkretnim problemima. Drugi se odnosi na primenu konvolucionih neuronskih mreža za automatsku klasifikaciju slika.

Prvi pristup je dominantan u situacijama kada ne postoji dovoljna količina trening podataka i kada nema uslova za dugotrajno treniranje klasifikatora. Istraživanja u oblasti novih deskriptora idu u smeru pronalaženja efikasnih deskriptora što niže dimenzionalnosti [46], deskriptora prilagođenih specifičnoj nameni (npr. udaljena detekcija [40], medicina [47], itd.), kao i u smeru kombinovanja (fuzije) deskriptora [48].

Konvolucione neuronske mreže pokazuju superiorne rezultate u slučaju kada postoji dovoljno velika količina trening podataka [44], [49], [50], uz osobinu da je njihovo treniranje izuzetno zahtevno u računskom i vremenskom pogledu. Da bi postigle vrhunske rezultate, konvolucione neuronske mreže zahtevaju treniranje nad velikim kolekcijama slika, kao što je ImageNet LSVRC-2010 [51] koja sadrži 1.2 miliona slika klasifikovanih u 1000 klasa. Međutim, za potrebe klasifikacije slika iz specifičnog domena primene (npr. snimci terena iz vazduha, medicinski snimci, itd.), ne postoje dovoljno velike kolekcije označenih slika, pri čemu njihovo formiranje može biti veoma zahtevno i skupo. Zato se istražuju načini prilagođavanja postojećih modela koji se trenirani na velikim kolekcijama opštih slika, kroz dodatno treniranje nad slikama iz specifičnog domena primene [52], [53].

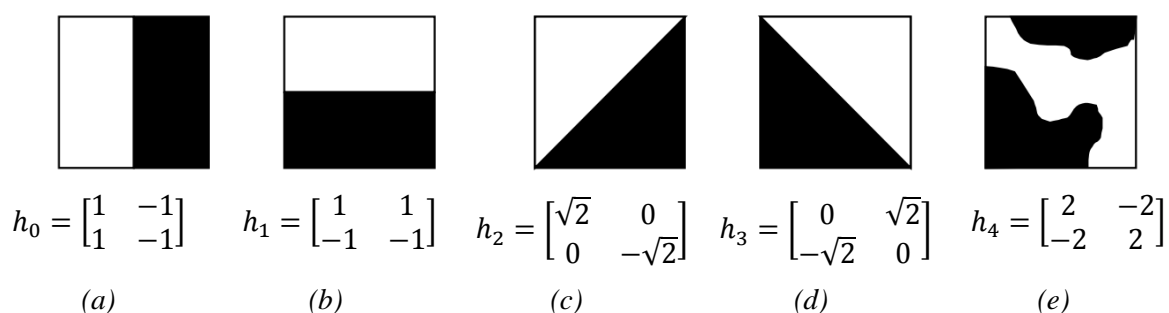
3 DESKRIPTORI SLIKE

Vizuelna obeležja slike kao što su tekstura ili boja, uz opis dominantnih oblika, predstavljaju primarne informacije za opis slike. Tekstura i boja se dominantno koriste za izračunavanje deskriptora slike u sistemima za automatsku klasifikaciju slika. U ovom poglavlju će biti opisani deskriptori teksture i boje čija će fuzija biti razmatrana u postupcima automatske klasifikacije. U novijoj literaturi je opisan veliki broj deskriptora slike i postupaka za njihovo izračunavanje. U nastavku će biti opisani samo reprezentativni postupci za izračunavanje deskriptora koji će biti korišćeni u postupcima fuzije predstavljenim u ovoj disertaciji. Za dodatne informacije o deskriptorima slike, čitalac se upućuje na [2], [54].

3.1 Deskriptori teksture

3.1.1 Histogram ivica slike – EHD

Histogram ivica slike (engl. Edge Histogram Descriptor - EHD) predstavlja prostornu raspodelu 5 vrsta ivica različitih orijentacija u slici [54], [55]. Postupak započinje podelom slike na nepreklapajuće blokove (tzv. “pod slike”), a zatim se izračunava po jedan histogram od 5 elemenata (5 orijentacija ivica) za svaku podsliku. EHD podrazumeva podelu na $4 \times 4 = 16$ nepreklapajućih blokova (“pod slike”), pa konačni histogram ivica cele slike sadrži ukupno $5 \times 16 = 80$ elemenata.



Slika 7 Prikaz pet tipova ivica slike za izračunavanje EHD uz odgovarajuće maske filtara za detekciju [54]: a) h_0 - vertikalna 90° , b) h_1 - horizontalna 0° , c) h_2 - dijagonalna 45° , d) h_3 - dijagonalna 135° , e) h_4 - neorijentisana.

Izračunavanje dominantne orijentacije svakog piksela u slici obavlja se primenom 5 filtara koje prikazuje Slika 7. Da bi se odredila dominantna orijentacija, vrši se konvolucija

slike $f(x, y)$ sa svakom od 5 maski filtara (h_0, \dots, h_4), i određuje se filtar sa maksimalnim odzivom. Ovo se može zapisati kao:

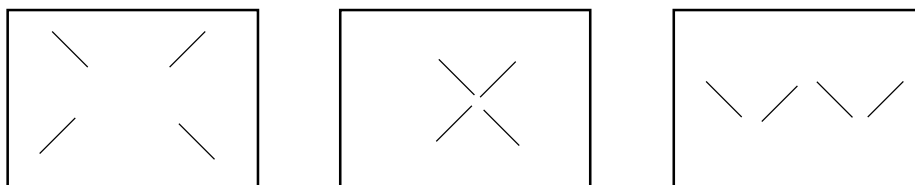
$$g_i(x, y) = f(x, y) * h_i(x, y); i = 0, 1, 2, 3, 4$$

$$g_{max} = \max_{0 \leq i \leq 4} (|g_i(x, y)|) \quad 3.1$$

$$i_{max} = \operatorname{argmax}_{0 \leq i \leq 4} (|g_i(x, y)|)$$

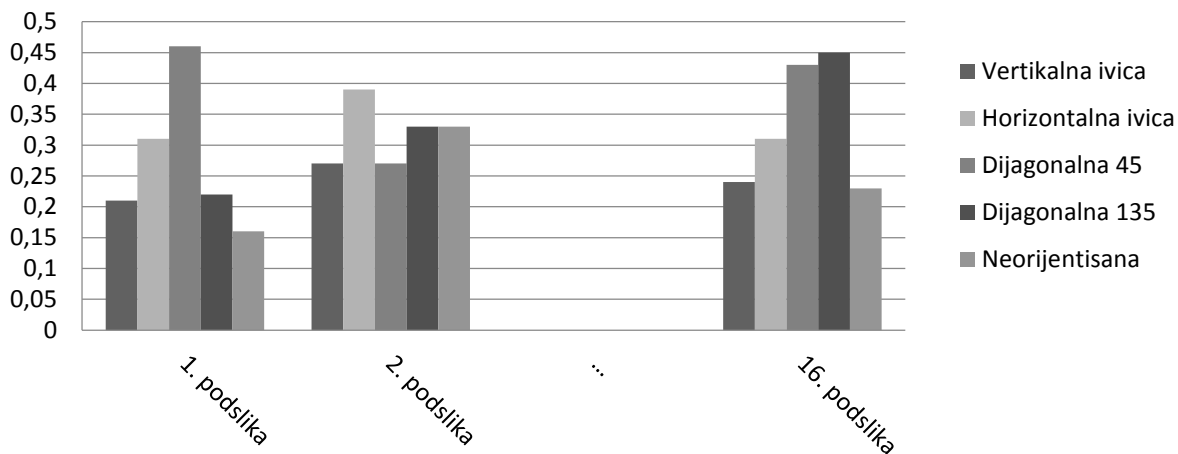
Ukoliko je maksimalni odziv veći od unapred definisanog praga $g_{max} \geq Th$, onda se za piksel smatra da ima dominantu orijentaciju i_{max} , što se upisuje na odgovarajućoj poziciji u histogramu. U suprotnom, ukoliko je maksimalni odziv manji od praga $g_{max} < Th$, smatra se da piksel nema nijednu dominantnu orijentaciju, i neće biti upisan u histogram. Uočava se da pikseli bez dominantne orijentacije nisu uračunati u histogramu, pa je histogram neophodno normalizovati na kraju izračunavanja.

Da bi obezbedili robusnost deskriptora u odnosu na promenu rezolucije slike, detekcija ivica se ne obavlja direktno nad originalnom slikom. Umesto toga, slika se deli na unapred fiksiran broj nepreklapajućih manjih blokova (prema MPEG-7 standardu [54], broj blokova je ukupno 1100). Za svaki blok se izračunavaju 2×2 odmerka kao srednje vrednosti piksela. Zatim se nad ovako izračunatim odmercima slike (2×2) primenjuje svih 5 filtara koje prikazuje Slika 7. Dominantna orijentacija se određuje na osnovu maksimalnog odziva, kao što je prethodno bilo objašnjeno.



Slika 8 Primer tri slike sa različitim rasporedom ivica, koje imaju isti histogram ivica slike - EHD.

Dakle, za svaki od 1100 blokova slike biće izračunata dominantna orijentacija, kao jedna od 5 vrednosti. Na osnovu ovih vrednosti treba izračunati histogram, tj. raspodelu verovatnoća za 5 dominantnih orijentacija. Ukoliko bi cela slika bila predstavljena jedinstvenim histogramom ivica, postoji mogućnost da bi dve slike sa potpuno različitim rasporedom ivica, imale isti histogram orijentacija ivica (Slika 8). Da bi prevazišli ovaj problem, MPEG-7 standardom je predviđeno da se originalna slika prvo podeli na $4 \times 4 = 16$ podslika, i da se za svaku podsliku izračuna po jedan histogram od 5 elemenata. Nadovezivanjem histograma svih 16 podslika biće formiran konačni deskriptor cele slike od $5 \times 16 = 80$ elemenata.



Slika 9 Primer izračunatog histograma ivica slike – EHD.

Napominjemo da je MPEG-7 standardom dozvoljeno izračunavanje i dodatnih poluglobalnih i globalnih histograma slike, sa grubljom podelom na podslike. Ovi histogrami se mogu nadovezati na osnovni histogram od 80 elemenata, kako bi se dobila diskriminativnija reprezentacija slike.

Prethodno opisani postupak izračunavanja EHD podrazumeva monohromatsku (sivu) ulaznu sliku. Ukoliko se na ulaz dovede slika u boji (RGB), treba prvo izračunati njen lumentni signal, i nad njim izvršiti izdvajanje EHD deskriptora.

3.1.2 Lokalni Binarni Uzorci – LBP

Lokalni binarni uzorak (engl. Local Binary Pattern - LBP) je široko korišćeni deskriptor slike baziran na operatoru koji izračunava izgled teksture oko centralnog piksela. Ovaj deskriptor je uveden u radu [56] za problem klasifikacije tekstura, a zatim je unapređen da postiže invarijantnost u odnosu na rotaciju kao i da obuhvati šire susedstvo oko piksela [23]. Nakon toga, LBP je unapređivan i premenjivan na mnoge probleme iz oblasti digitalne obrade slike [36].

Za zadatu sliku f , lokalni LBP deskriptor za centralni piksel f_0 predstavlja niz od 8 bitova, gde svaki bit predstavlja kodovanje jednog od suseda iz lokalnog susedstva (Slika 10). Prilikom računanja binarnog koda, prvo se za svaki od susednih piksela vrši upoređivanje intenziteta tog piksela sa intenzitetom centralnog piksela. Ako je intenzitet susednog piksela veći od centralnog – ovo će biti kodovano kao '1', dok će u suprotnom kod biti '0'. Neka je centralni piksel označen sa f_0 , a njegovi susedni pikseli sa f_i , $i = 1, \dots, 8$ počevši od suseda na poziciji desno od centralnog piksela, obilazeći u smeru suprotnom od časovnika:

$$\begin{bmatrix} f_4 & f_3 & f_2 \\ f_5 & f_0 & f_1 \\ f_6 & f_7 & f_8 \end{bmatrix}$$

Onda se binarni kod razlike suseda i centralnog piksela $B(f_i, f_0)$ može izraziti kao:

$$B(f_i, f_0) = \begin{cases} 1 & , f_i > f_0 \\ 0 & , f_i \leq f_0 \end{cases} \quad 3.2$$

Formiranje niza na osnovu prethodno izračunatih binarnih kodova se vrši obilaskom suseda u smeru suprotnom od časovnika, počevši od suseda na poziciji desno od centralnog piksela f_0 . Ovako formirani binarni niz se zatim konvertuje u decimalni broj, koji predstavlja LBP vrednost centralnog piksela f_0 . Matematički se ovo može predstaviti kao:

$$\begin{aligned} LBP(f_0) &= B(f_1, f_0)2^0 + B(f_2, f_0)2^1 \dots B(f_8, f_0)2^7 = \\ &= \sum_{i=1}^8 B(f_i, f_0)2^{i-1} \end{aligned} \quad 3.3$$

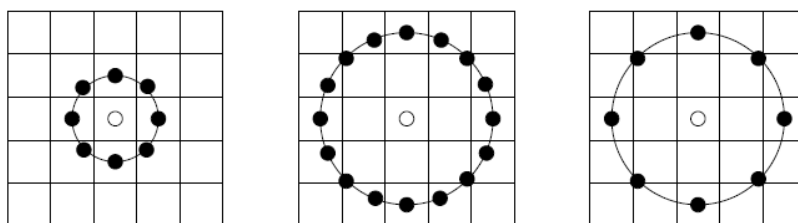
Na narednoj slici je dat primer izračunavanja lokalnog LBP deskriptora:

$$f = \begin{bmatrix} 167 & 221 & 221 \\ 147 & \mathbf{217} & 198 \\ 132 & 230 & 212 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad LBP = (01100010)_2 = 98$$

Slika 10 Primer izračunavanja lokalnog LBP deskriptora za centralni piksel sa intenzitetom $f_0 = 217$ i lokalnim susedstvom dimenzija 3×3 .

Ukoliko se koristi lokalno susedstvo dimenzija 3×3 , postoji ukupno $256 = 2^8$ mogućih LBP uzoraka. Primenom unapredene LBP metode iz [23], ovaj broj može biti redukovano na 58 uzoraka koji su invarijantni u odnosu na rotaciju. Ovakvo unapređenje bilo je inspirisano činjenicom da se pojedini uzorci pojavljuju znatno čeće od ostalih, pa ostali uzorci mogu biti eliminisani ili grupisani.

Iako je originalni LBP deskriptor bio definisan samo za 3×3 okolinu piksela, unapređenja LBP metode iz [23], [57] proširila su definiciju deskriptora tako da se okolina piksela može posmatrati na proizvoljnoj udaljenosti i sa proizvoljnim brojem susednih piksela (Slika 11).



Slika 11 Primeri različito definisanih okolina za izračunavanje LBP [57]. Ukoliko se sused nalazi na granici više piksela, onda se vrši bilinearna interpolacija graničnih piksela.

Za reprezentaciju celokupne slike jedinstvenim deskriptorom, uobičajeno se koristi histogram prethodno izračunatih LBP vrednosti svih piksela. Da bi se postigla dodatna

robusnost deskriptora, pojedini algoritmi prvo dele sliku na blokove, izračunavaju po jedan histogram za svaki blok, i formiraju globalni deskriptor slike konkatencijom prethodno izračunatih „blokovskih“ histograma. Ovakav deskriptor, međutim pokazuje slabosti u slučaju rotacije slike. Zato je u ovom radu, za problem klasifikacije slika, korišćen jedan globalni LBP histogram izračunat na celokupnoj slici

U slučaju slike u boji, konačni višekanalni LBP deskriptor koji uključuje i informacije o boji, može biti izračunat na način koji smo predložili u radu [12]. U radu je korišćen $YCbCr$ prostor boja, s obzirom da standardni RGB prostor boja pokazuje veliku korelaciju među komponentama boje. U radu je predloženo da se ulazna slika prvo konvertuje u $YCbCr$ prostor boja, a zatim je nad svakom od komponenti Y, Cb i Cr izračunat po jedan nezavisni LBP histogram. Da bi u konačnom deskriptoru bile objedinjene informacije o slici sa različitim rezolucijama, i na taj način se postigla robusnost deskriptora na promenu rezolucije, izdvajanje LBP histograma je obavljeno nad originalnom slikom kao i slikama smanjenje rezolucije. Slike manje rezolucije su dobijene tako što je originalna slika rezolucije $M \times N$ umanjena na rezoluciju $\frac{M}{2} \times \frac{N}{2}$, pri čemu je korišćena bikubična interpolacija piksela. Korišćeno je ukupno tri rezolucije slike tj. originalna slika i dve umanjene slike. Konačni višerezolucijski kolor descriptor slike je formiran konkatencijom LBP histograma izdvojenih nad sve 3 komponente boje i 3 različite rezolucije slike. Dakle, konačni LBP histogram slike će imati ukupno $3 \times 3 \times 58 = 522$ elementa.

3.1.3 Binarni Gaborovi Uzorci – BGP

Binarni Gaborovi Uzorci (engl. Binary Gabor Pattern - BGP) je deskriptor slike koji je nedavno uveden za potrebe klasifikacije tekstura [24]. Osnovu ovog deskriptora predstavlja korišćenje banke Gaborovih filtara sa različitim parametrima. Banka Gaborovih filtara se primenjuje na celokupnu sliku, a zatim se odzivi u svakom pikselu koduju u jedinstven decimalni broj, uz očuvanje rotacione invarijantnosti.

Dvodimenzionalnu Gaborovu funkciju uveo je Daugman [58] kako bi modelirao dvodimenzionalna receptivna polja prostih ćelija u vizuelnom korteksu ljudskog mozga. Osnovni 2D Gaborov filter je dobijen modulacijom sinusoidnog signala Gausovom funkcijom, i može se predstaviti parom parnog i neparnog filtra:

$$g_1(x, y, \theta, \sigma, \gamma, \lambda) = \exp \left[-\frac{1}{2} \left(\frac{x'^2}{\sigma^2} + \frac{y'^2}{(\gamma\sigma)^2} \right) \right] \cos \left(\frac{2\pi}{\lambda} x' \right) \quad 3.4$$

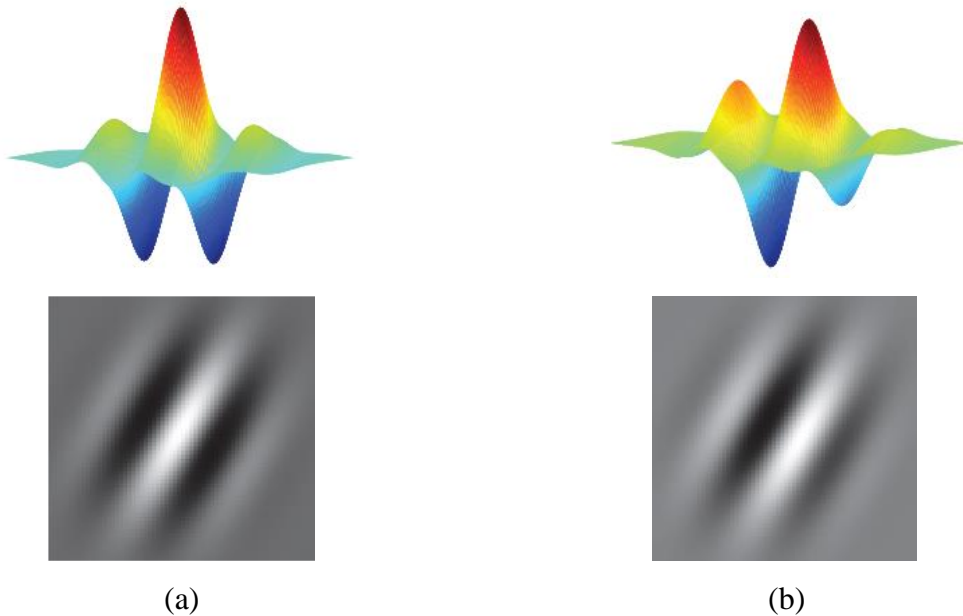
$$g_2(x, y, \theta, \sigma, \gamma, \lambda) = \exp\left[-\frac{1}{2}\left(\frac{x'^2}{\sigma^2} + \frac{y'^2}{(\gamma\sigma)^2}\right)\right] \sin\left(\frac{2\pi}{\lambda} x'\right)$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -y\sin\theta + y\cos\theta$$

gde je:

- θ – orijentacija Gaborovog filtra tj. ugao koji gradi normala na dominantnu orijentaciju filtra,
- σ - standardna devijacija Gausove krive koja predstavlja prostornu širinu filtra (u pikselima),
- γ – prostorni odnos širine i visine koji određuje eliptični oblik filtra (uobičajeno fiksiran na vrednost 2),
- λ – talasna dužina filtra u prostoru (u pikselima).



Slika 12 Primer 2D Gaborovog filtra: (a) parni tj. realni deo, (b) neparni tj. imaginarni deo.

Ako fiksiramo parametre σ, γ i λ , nadalje možemo koristiti notaciju $g(x, y, \theta)$ za označavanje Gaborovog filtra orijentacije θ (obe parnosti). Za zadatu digitalnu sliku $f(x, y)$, odziv Gaborovog filtra se može predstaviti konvolucijom:

$$G(x, y, \theta) = f(x, y) * g(x, y, \theta) \quad 3.5$$

Da bi se formirao robusni lokalni deskriptor za svaki piksel u slici, prvo se vrši konvolucija slike sa bankom Gaborovih filtara od n različitih diskretnih orijentacija $\theta_i = i \frac{\pi}{n}$; $i = 0, \dots, n - 1$.

Zatim se za svaku lokaciju piksela (x, y) , svaki od n odziva filtara $G(x, y, \theta_i)$ konvertuje u jednu binarnu cifru. Korišćeno je jednostavno upoređivanje sa pragom da bi se odziv u svakom pikselu predstavio kompaktnim binarnim kodom (označenim sa $BG(x, y, \theta)$):

$$BG(x, y, \theta) = \begin{cases} 1 & , G(x, y, \theta) \geq 0 \\ 0 & , G(x, y, \theta) < 0 \end{cases} \quad 3.6$$

Konkatenacijom prethodno izračunatih n binarnih vrednosti, dobija se n -bitna reprezentacija tj. kodovanje piksela na lokaciji (x, y) . Ovakav n -bitni niz se može konvertovati u jedinstven decimalni broj, koji predstavlja rotaciono zavistan binarni Gaborov uzorak $BGP'(x, y, n)$:

$$\begin{aligned} BGP'(x, y, n) &= [BG(x, y, \theta_1), BG(x, y, \theta_2), \dots, BG(x, y, \theta_{n-1})]_2 \\ &= \sum_{i=0}^{n-1} BG(x, y, \theta_i) \cdot 2^i \end{aligned} \quad 3.7$$

gde $[\cdot]_2$ označava binarni broj. Na osnovu zaključaka iz [23], uočeno je da bitski pomerene vrednosti $BGP'(x, y, n)$ predstavljaju isti uzorak teksture, koji je samo zarotiran za određeni ugao. Dakle, da bi formirali BGP deskriptor koji je invarijantan u odnosu na rotaciju, dovoljno je da grupišemo zajedno pomerene BGP' vrednosti. Formalno, ovo se može zapisati kao:

$$BGP(x, y, n) = \max_{0 \leq i \leq n-1} ROR(BGP'(x, y, n), i) \quad 3.8$$

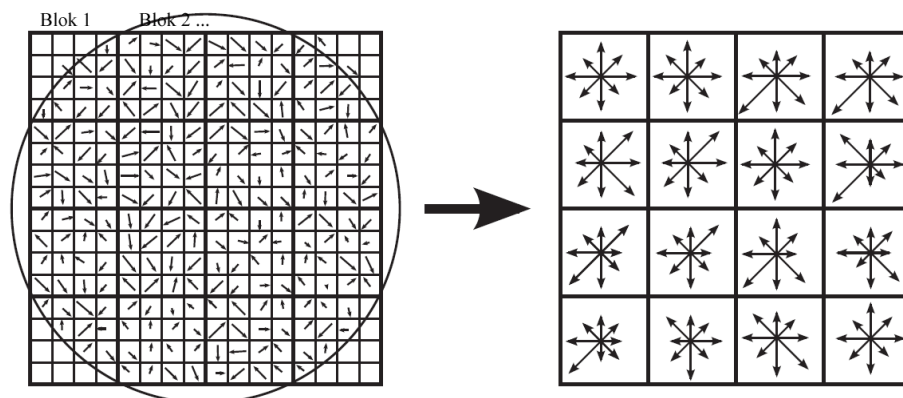
gde $ROR(BGP'(x, y, n), i)$ vrši cirkularno pomeranje u desno nad n -bitnim brojem $BGP'(x, y, n)$. Osim što postiže invarijantnost deskriptora u odnosu na rotaciju, što je od suštinske važnosti za automatsku klasifikaciju slika, prethodna operacija značajno redukuje dimenzionalnost globalnog deskriptora slike. Npr, ako je $BGP'(x, y, 6) = (001100)_2$, onda će sledećih 6 uzoraka biti grupisano u jedinstveni rotaciono invarijantni uzorak $BGP(x, y, 6) = \max\{(001100)_2, (000110)_2, (000011)_2, (100001)_2, (110000)_2, (011000)_2\} = (110000)_2 = 48$. Na ovaj način, inicijalni skup od $2^6 = 64$ mogućih vrednosti $BGP'(x, y, 6)$ biće grupisan u 14 različitih vrednosti rotaciono invarijantnih binarnih Gaborovih uzoraka $BGP(x, y, 6)$. U slučaju kada je broj orijentacija Gaborovog filtra $n = 8$, inicijalni skup od $2^8 = 256$ kodova biće redukovan na 36 vrednosti $BGP(x, y, 8)$.

Nakon izračunavanja $BGP(x, y, n)$ na svakoj lokaciji piksela slike, globalni deskriptor slike se izračunava kao normalizovani histogram $BGP(x, y, n)$ vrednosti. Treba uočiti da se za jednu sliku izračunavaju dva deskriptora na osnovu parnog i neparnog Gaborovog filtra, i nadovezuju se u jedinstveni deskriptor slike označen kao $BGP(n)$.

3.1.4 SIFT deskriptor

Najviše primenjivan lokalni deskriptor slike je SIFT (engl. Scale Invariant Feature Transform) predstavljen u radu [20]. Ovaj deskriptor pokazuje veoma dobru stabilnost prilikom različitih geometrijskih transformacije slike. SIFT deskriptor je prvobitno predložen za detekciju objekata u slikama, gde je izračunavan samo u okolini karakterističnih tačaka slike (engl. keypoints), koje su prethodno određene adekvatnim algoritmom. Karakteristične tačke predstavljaju lokacije u slici koje ostaju uočljive na istoj lokaciji nakon različitih geometrijskih i svetlosnih transformacija. SIFT je kasnije masovno primenjivan u klasifikaciji slika, gde je izračunavan u pravilno raspoređenim tačkama slike dobijenim odmeravanjem sa malim korakom (engl. dense sampling) [15–17].

Bez obzira na to kako su određene karakteristične tačke u čijoj okolini će biti izračunati deskriptori, SIFT se izračunava u lokalnim regionima kvadratnog oblika oko detektovanih karakterističnih tačaka. Za dati kvadratni region, SIFT deskriptor se izračunava tako što se izračunavaju vrednosti gradijenta slike unutar tog regiona. Region se prvo pravilnom mrežom dimenzija 4×4 deli na blokove. Svaki blok sadrži 4×4 lokacije u kojima se izračunava gradijent, a zatim se za svaki blok se formira histogram orijentacija gradijenta (Slika 13). Prilikom formiranja histograma orijentacija, koristi se 8 kvantizovanih vrednosti uglova, tako da je ukupna dimenzionalnost lokalnog SIFT deskriptora $4 \times 4 \times 8 = 128$.



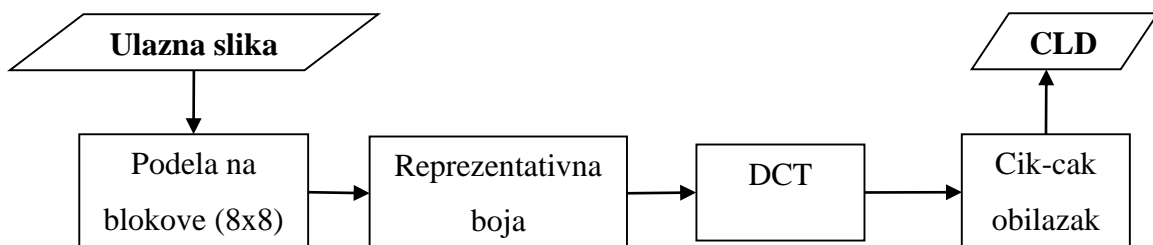
Slika 13 Postupak izračunavanja SIFT deskriptora na jednom lokalnom regionu.

Da bi se formirala kompaktna reprezentacija celokupne slike na osnovu izračunatih lokalnih deskriptora, formira se histogram broja pojavljivanja kodnih reči koje su dobijene vektorskom kvantizacijm 128-dimenzionalnih lokalnih deskriptora. Ovakva reprezentacija je zasnovana na skupovima vizuelnih reči (engl. Bag of Words – BoW) o kojoj je bilo više reči u Poglavlju 2.3.

3.2 Deskriptori boje

3.2.1 Deskriptor rasporeda boje – CLD

Deskriptor rasporeda boje (engl. Color Layout Descriptor – CLD), je formulisan sa ciljem da obezbedi kompaktnu reprezentaciju prostornog rasporeda boja u slici. Ovaj deskriptor je deo međunarodnog standarda MPEG-7 [54], [59]. Izdvajanje ovog deskriptora započinje izračunavanjem reprezentativnih boja za svaki od nepreklapajućih blokova slike, nakon čega se ove vrednosti kompresuju primenom 2D Diskretne Kosinusne Transformacije (DCT). Deskriptor se karakteriše visokom efikasnošću postupka izračunavanja, kompaktnom reprezentacijom i invarijantnošću u odnosu na promenu rezolucije slike. U brojnim publikacijama je pokazana visoka uspešnost ovog deskriptora u postupcima pretraživanja i klasifikacije slika [60].



Slika 14 Postupak izračunavanja CLD

Izračunavanje CLD je u ovom radu obavljeno u $YCbCr$ prostoru boja, bez dodatnog kvantizacionog koraka koji je deo zvaničnog MPEG-7 standarda. Istraživanja su pokazala da ovaj korak, koji unosi dodatnu kompleksnost izračunavanja, ne utiče na unapređenje rezultata. Izračunavanje se obavlja nezavisno za svaki kolor kanal $c \in \{Y, Cb, Cr\}$. Postupak započinje podelom slike na $8 \times 8 = 64$ nepreklapajućih blokova, što obezbeđuje invarijantnost izračunatog deskriptora u odnosu na eventualnu promenu rezolucije slike. Zatim se za svaki blok izračunava po jedna reprezentativna vrednost $b_c(i, j)$; $i = 0, 1, \dots, 7$; $j = 0, 1, \dots, 7$, kao srednja vrednost piksela unutar tog bloka. Pokazalo se da ovako jednostavan način izračunavanja reprezentativne vrednosti piksela unutar bloka obezbeđuje dovoljnu tačnost uz

minimalnu kompleksnost izračunavanja. Nakon izračunavanja 64 vrednosti $b_c(i, j)$ za svaki kolor kanal, obavlja se njihovo kompresovanje korišćenjem 2D-DCT. Ako DCT koeficijente kolor kanala c označimo sa $F_c(u, v)$, onda se za izračunavanje koriste formule:

$$F_c(u, v) = C(u)C(v) \sum_{i=0}^7 \sum_{j=0}^7 b_c(i, j) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16}$$

$$C(u) = \begin{cases} 1/\sqrt{8} & , u = 0 \\ 1/2 & , u \neq 0 \end{cases} \quad 3.9$$

$$C(v) = \begin{cases} 1/\sqrt{8} & , v = 0 \\ 1/2 & , v \neq 0 \end{cases}$$

Izračunati DCT koeficijenti $F_c(u, v)$ se zatim preuređuju prema „cik-cak” redosledu obilaženja, kako bi se najznačajniji DCT koeficijenti grupisali na početku. Inače, ovakva tehnika se koristi i u JPEG standardu [61]. Konačno, najznačajniji DCT koeficijenti će biti izdvojeni za svaki kanal boje c : $CLD_c = [F_c(0,0), F_c(0,1), F_c(1,0), F_c(2,0), F_c(1,1), F_c(0,2), F_c(0,3), F_c(1,2), F_c(2,1), F_c(3,0), \dots]$

Konačni CLD deskriptor slike u boji, formira se konkatencijom CLD-a izračunatih zasebno za svaki kanal boje:

$$CLD = [CLD_Y, CLD_{Cb}, CLD_{Cr}] \quad 3.10$$

Empirijski je utvrđeno da je za potrebe klasifikacije slike dovoljno koristiti do 10 najznačajnijih DCT koeficijenata za svaki kanal boje. U ovom radu je korišćen deskriptor dužine 22, pri čemu je 10 koeficijenata za Y kanal, 6 koeficijenata za Cb , i 6 za Cr kanal boje.

3.2.2 Histogram boje

Ovo je bazični deskriptor boje, koji opisuje globalnu raspodelu boja u slici predstavljenoj u određenom prostoru boja [62]. Standardna tehnika za izračunavanje ovog deskriptora u RGB prostoru boja podrazumeva da se prvo izračunaju tri nezavisna histograma za svaku komponentu boje (R, G i B), a zatim se izvrši nadovezivanje ovako izračunatih histograma u jedinstveni deskriptor. Ako pretpostavimo da se za svaki kanal boje $f_R, f_G, i f_B$, izračunava po jedan histogram $h_R, h_G, i h_B$ respektivno, onda se ovo matematički može predstaviti kao:

$$h_R(i) = \text{card}\{(x, y) \mid f_R(x, y) = i\}; \quad i = 1, \dots, 256$$

$$h_G(i) = \text{card}\{(x, y) \mid f_G(x, y) = i\}; \quad i = 1, \dots, 256$$

$$h_B(i) = \text{card}\{(x, y) \mid f_B(x, y) = i\}; \quad i = 1, \dots, 256$$

$$h = [h_R, h_G, h_B] \quad 3.11$$

Gde $card\{.\}$ označava kardinalnost skupa, tj. broj elemenata skupa. Podrazumevano se koristi histogram od 256 elemenata za svaki kanal boje. Međutim, kako bi redukovali dimenzionalnost deskriptora, moguće je koristiti “grublju” kvantizaciju, tj. umesto 256 vrednosti po kanalu koristiti npr. 64 ili 16. Mi smo u radu [13] koristili histograme sa 16 vrednosti po kanalu boje, što rezultira globalnim deskriptorom slike od $3 \times 16 = 48$ elemenata. Eksperimenti sa preciznijom kvantizacijom piksela u histogramima, nisu pokazali povećanje performansi klasifikacije.

Treba uočiti da deskriptori boje, uključujući CLD i histogram boje, vrše redukovanje veoma velikog prostora boja (npr. 2^{24}) na relativno mali prostor (npr. 22 ili 48), što predstavlja nedovoljno preciznu („grubu“) aproksimaciju celokupnog prostora boja (npr. 2^{24} elementa u RGB prostoru), koja nije dovoljno diskriminativna da bi se koristila kao samostalni deskriptor slike. Ovo pokazuju rezultati brojnih eksperimenata [63], [64]. Zato je preporučljivo kombinovati deskriptore boje sa deskriptorima teksture, kako bi se dobila reprezentacija slike koja je adekvatna za postizanje visoke tačnosti prilikom klasifikacije slika.

4 NADGLEDANO MAŠINSKO UČENJE ZA KLASIFIKACIJU SLIKA

Nadgledano mašinsko učenje predstavlja granu veštačke inteligencije koja proučava sisteme koji su u stanju da “nauče“ matematičke modele iz velikih skupova podataka. U cilju lakšeg razumevanja i praćenja daljeg sadržaja disertacije, u ovom poglavlju će prvo biti definisana osnovna terminologija i notacije iz oblasti nadgledanog mašinskog učenja. Zatim sledi opis pojmova i najvažnijih tehnika nadgledanog mašinskog učenja. Poseban akcenat biće stavljen na opisu veštačkih neuronskih mreža, koje su dominantno korišćene za klasifikaciju slika u ovoj disertaciji. Konačno će biti predstavljena efikasna metoda za ekstremno brzo obučavanje neuronskih mreža (engl. Extreme Learning Machines).

4.1 Teorijske osnove nadgledanog mašinskog učenja

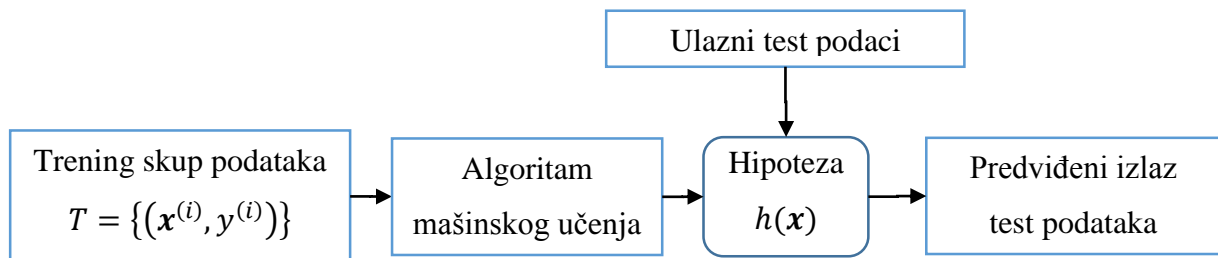
Neka je sa $\mathbf{x}^{(i)} \in R^n$ označen jedan vektor ulaznih podataka (engl. feature vector), a sa $y^{(i)} \in R$ odgovarajuća izlazna tj. ciljna vrednost (engl. output, target). Uređeni par $(\mathbf{x}^{(i)}, y^{(i)})$ se naziva i -ti trening primer (eng. training example), a konačan skup parova podataka koji ćemo koristiti za treniranje $T = \{(\mathbf{x}^{(i)}, y^{(i)})\}; i = 1, \dots, M$ se naziva **trening skup**. U ovako definisanom trening skupu T , postoji M trening vektora $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}]^T$, i svakom od njih je pridružena poznata skalarna vrednost izlazne promenljive $y^{(i)}$. Napominjemo da oznaka " i " u ovakvoj notaciji predstavlja indeks objekta u trening skupu, u našem slučaju indeks slike tj. njenog deskriptora, i nema nikakve povezanosti sa eskponencijalnom funkcijom.

	Ulazni podatak	Izlaz
Trening skup	$\mathbf{x}^{(1)}$	$y^{(1)}$
	$\mathbf{x}^{(2)}$	$y^{(2)}$

	$\mathbf{x}^{(M)}$	$y^{(M)}$
Test skup	$\mathbf{x}^{(M+1)}$?

	$\mathbf{x}^{(N)}$?

Cilj nadgledanog mašinskog učenja je da na osnovu zadatog trening skupa T “nauči” funkciju $h(\mathbf{x})$ koja preslikava skup ulaznih vektora $\mathbf{x}^{(i)} \in R^n$ na skup izlaznih promenljivih $y^{(i)} \in R$, tako da $h(\mathbf{x})$ na najbolji način aproksimira odgovarajuće vrednosti izlaznih promenljivih $y^{(i)}$. Matematički posmatrano, problem određivanja funkcije $h(\mathbf{x})$ predstavlja problem **aproksimacije funkcije** $h(\mathbf{x})$. Funkcija $h(\mathbf{x}): R^n \rightarrow R$ se naziva **funkcija predikcije**, a u literaturi se često sreće pod nazivom **hipoteza**. U praksi se umesto pojma **naučiti funkciju** predikcije $h(\mathbf{x})$, uobičajeno koristi termin **trenirati model**. Funkcija predikcije (hipoteza) se formira primenom odgovarajućeg algoritma nadgledanog mašinskog učenja, na osnovu poznatog trening skupa podataka (Slika 15). Ovo praktično znači da se parametri hipoteze $h(\mathbf{x})$ izračunavaju kroz postupak treniranja modela.



Slika 15 Postupak nadgledanog mašinskog učenja.

U slučaju kada su izlazne vrednosti kontinualne tj. $y^{(i)} \in R$, tada se problem naziva **regresija**. Ukoliko izlazne vrednosti mogu da imaju samo konačan skup diskretnih vrednosti, problem se naziva **klasifikacioni problem**. U kontekstu klasifikacije slika, ulazni podatak $\mathbf{x}^{(i)} \in R^n$ predstavlja vektor deskriptora slike, dok izlazni podatak $y^{(i)} \in N$ predstavlja oznaku klase kojoj slika (i) pripada. Dakle, cilj algoritama nadgledanog mašinskog učenja je da “nauči” parametre funkcije hipoteze $h(\mathbf{x})$, tako da je $h(\mathbf{x}^{(i)}) \approx y^{(i)}$ za svako i iz trening skupa podataka T . Ova prva faza algoritma se naziva **faza treniranja**. Nakon određivanja optimalne funkcije $h(\mathbf{x})$, ova funkcija se dalje koristi za izračunavanje (tj. predviđanje) izlaza na osnovu novih test ulaznih podataka koji nisu sadržani u trening skupu. Ova faza algoritma se naziva **faza testiranja**.

Da bismo izračunali optimalne parametre hipoteze $h(\mathbf{x})$, prvo je neophodno da funkciju $h(\mathbf{x})$ predstavimo u određenom parametarskom obliku. U najjednostavnijem slučaju, možemo se opredeliti za prostor linearnih funkcija:

$$h(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + \dots + w_nx_n \quad 4.1$$

gde $\mathbf{w} = [w_0, w_1, \dots, w_j, \dots, w_n]^T$ predstavlja parametre (tj. težine) koji određuju prostor linearnih funkcija koje preslikavaju skup ulaznih vektora $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ na skup izlaza. Ovaj prostor funkcija se uobičajeno naziva **prostor hipoteza**. Da bi se prethodna formula za $h(\mathbf{x}, \mathbf{w})$ predstavila u kompaktnom obliku, kao proizvod dva vektora \mathbf{w} i \mathbf{x} , neophodno je svaki vektor ulaznih promenljivih $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ proširiti elementom $x_0 = 1$, tako da postane $\mathbf{x} = [1, x_1, x_2, \dots, x_n]^T$. Prostor hipoteza se onda može predstaviti kao proizvod dva vektora:

$$h(\mathbf{x}, \mathbf{w}) = [w_0, w_1, \dots, w_n] \begin{bmatrix} 1 \\ x_1 \\ \dots \\ x_n \end{bmatrix} = \mathbf{w}^T \mathbf{x} \quad 4.2$$

Za ovako odabran linearni prostor hipoteza, cilj je pronaći optimalne parametre \mathbf{w} , tako da vrednost $h(\mathbf{x}^{(i)})$ bude što sličnija odgovarajućem izlazu $y^{(i)}$, za svaki primer (i) iz trening skupa, tj. $h(\mathbf{x}^{(i)}, \mathbf{w}) \approx y^{(i)}$. Da bismo ovo obavili, definišaćemo funkciju koja izračunava razliku vrednosti $h(\mathbf{x}^{(i)}, \mathbf{w})$ i odgovarajućeg $y^{(i)}$, u zavisnosti od izbora parametara \mathbf{w} . Ovakva funkcija se naziva **funkcija greške** (engl. *loss function, cost function, objective function*). Funkcija greške treba da objedini razlike između $h(\mathbf{x}^{(i)}, \mathbf{w})$ i $y^{(i)}$ za sve primere (i) iz trening skupa i uobičajeno se izračunava kao srednja kvadratna greška (engl. Mean Squared Error - MSE):

$$L(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^M (h(\mathbf{x}^{(i)}, \mathbf{w}) - y^{(i)})^2 \quad 4.3$$

Za ovako definisanu funkciju greške, cilj je odrediti parametre \mathbf{w} za koje je greška minimalna, tj. odrediti parametre \mathbf{w} koji minimizuju funkciju greške $L(\mathbf{w})$. Matematički posmatrano, treba rešiti optimizacioni problem, što se formalno može predstaviti kao pronalaženje optimalnih parametara \mathbf{w}^* :

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} L(\mathbf{w}) \quad 4.4$$

Najčešće korišćena metoda za rešavanje ovakvog optimizacionog problema je **metoda gradijentnog spusta** (eng. gradient descent) [65], [66]. Nakon izračunavanja parametara \mathbf{w} na osnovu trening skupa podataka, u potpunosti će biti definisana funkcija hipoteze $h(\mathbf{x})$. Ova funkcija zatim može biti iskorišćena za predviđanje novih izlaznih vrednosti za ulazne podatke iz test skupa. Dakle, nakon izračunavanja parametara funkcije hipoteze $h(\mathbf{x})$, prestaje potreba za čuvanjem celokupnog trening skupa u memoriji, već je dovoljno samo poznavanje funkcije hipoteze, tj. njenih parametara.

4.2 Minimizacija funkcije greške

Za pronalaženje optimalnih parametara \mathbf{w} koji minimizuju funkciju greške $L(\mathbf{w})$, u nastavku ćemo razmotriti najčešće korišćen algoritam pod nazivom *metoda gradijentnog spusta*. Naime, iz matematike je poznato da prvi izvod funkcije u proizvoljnoj tački određuje smer rasta funkcije u toj tački. Kada su u pitanju funkcije više promenljivih, kao što je $L(\mathbf{w}) \equiv L(w_0, w_1, \dots, w_j, \dots, w_n)$, onda je smer maksimalnog rasta funkcije određen gradijentom funkcije $\nabla L(\mathbf{w}) = \left[\frac{\partial L(\mathbf{w})}{\partial w_0}, \frac{\partial L(\mathbf{w})}{\partial w_1}, \dots, \frac{\partial L(\mathbf{w})}{\partial w_n} \right]^T$. Slično, negativni gradijent određuje smer opadanja funkcije u nekoj tački. Dakle, da bismo pronašli minimum funkcije $L(\mathbf{w})$, treba vršiti pretraživanje u smeru maksimalnog opadanja funkcije, tj. u smeru negativnog gradijenta.

Metoda gradijentnog spusta predstavlja iterativni metod, koji koristi prethodno opisane osobine gradijenta funkcije. Metoda započinje sa proizvoljnom vrednošću parametara \mathbf{w} , nakon čega iterativno ažurira vrednost \mathbf{w} kako bi smanjila vrednost greške $L(\mathbf{w})$. Ažuriranje vektora parametara \mathbf{w} se obavlja na sledeći način:

$$w_j := w_j - \alpha \frac{\partial L(\mathbf{w})}{\partial w_j}; j = 0, 1, \dots, n. \quad 4.5$$

gde je sa w_j označen jedan element vektora \mathbf{w} na poziciji j , dok se α naziva *korak učenja*. Uočava se da se ažuriranje vrednosti w_j obavlja u smeru najstrmijeg pada vrednosti $L(\mathbf{w})$, tj. u smeru koji je određen parcijalnim izvodom funkcije $L(\mathbf{w})$. Još jednom napominjemo da se ažuriranje vrednosti w_j obavlja u smeru negativnog gradijenta tj. smeru opadanja funkcije, s obzirom da je cilj minimizirati funkciju greške $L(\mathbf{w})$.

U cilju implementacije prethodnog algoritam, razmotrićemo prvo postupak izračunavanja parcijalnog izvoda funkcije greške u slučaju linearne funkcije hipoteze. Radi jednostavnosti označavanja, pretpostavićemo da postoji samo jedan trening primer, tj. $(\mathbf{x}^{(i)}, y^{(i)}) \equiv (\mathbf{x}, y)$.

$$\begin{aligned} \frac{\partial L(\mathbf{w})}{\partial w_j} &= \frac{\partial}{\partial w_j} \left(\frac{1}{M} (h(\mathbf{x}^{(i)}, \mathbf{w}) - y^{(i)})^2 \right) = \frac{1}{M} \frac{\partial}{\partial w_j} (\mathbf{w}^T \mathbf{x} - y)^2 \\ &= \frac{2}{M} (\mathbf{w}^T \mathbf{x} - y) x_j \end{aligned} \quad 4.6$$

Za jedan primer iz trening skupa podataka, može se izvesti jednačina za ažuriranje parametara:

$$w_j := w_j - \alpha \frac{\partial L(\mathbf{w})}{\partial w_j}; j = 0, 1, \dots, n$$

$$w_j := w_j + \frac{2\alpha}{M} (y - \mathbf{w}^T \mathbf{x}) x_j; j = 0, 1, \dots, n \quad 4.7$$

$$w_j := w_j + \hat{\alpha} (y - \mathbf{w}^T \mathbf{x}) x_j; j = 0, 1, \dots, n$$

gde je sa $\hat{\alpha} = \frac{2\alpha}{M}$ označen novi parametar učenja, bez gubljenja opštosti postupka. Prethodno pravilo se naziva **LMS pravilo ažuriranja** (engl. *Least Mean Squares - LMS*). Uočava se da ovo pravilo ima osobinu da je amplituda promene w_j proporcionalna grešci $(y - \mathbf{w}^T \mathbf{x})$, što je i logično očekivati. Naime, ukoliko je greška predikcije mala, tj. predviđena vrednost izlaza za $\mathbf{x}^{(i)}$ je bliska stvarnoj vrednosti $y^{(i)}$, onda treba izvršiti neznatno ažuriranje parametara w_j . Nasuprot tome, ukoliko je greška predikcije velika, onda treba značajno korigovati vrednosti w_j .

Treba primetiti da metoda gradijentnog spusta može ispoljiti osetljivost u slučaju lokalnog minimuma, što znači da može pronaći “optimalne” parametre koji odgovaraju lokalnom minimumu funkcije, a ne globalnom minimumu kako se očekuje. Takođe, postoji mogućnost da ukoliko korak postavimo na preveliku vrednost, da se prilikom iteracija “preskoči” minimum funkcije $L(\mathbf{w})$ i pređe na suprotnu stranu funkcije u odnosu na minimum, gde je vrednost funkcije veća nego u trenutnoj tački. Međutim, u slučaju linearne regresije, gde je funkcija greške $L(\mathbf{w})$ konveksna kvadratna funkcija koja ima samo jedan globalni minimum, ukoliko je korak učenja dovoljno mali, metoda gradijentnog spusta će uvek konvergirati ka globalno optimalnom rešenju.

4.2.1 Tehnike treniranja (obučavanja) modela

Prilikom izvođenja prethodnih pravila ažuriranja, pretpostavili smo da imamo samo jedan trening primer $(\mathbf{x}^{(i)}, y^{(i)}) \equiv (\mathbf{x}, y)$. Da bismo postupak ažuriranja težinskih parametara \mathbf{w} primenili nad većim brojem primera iz celokupnog trening skupa podataka, postoji nekoliko pristupa.

Prvi pristup, pod nazivom **stohastički gradijentni spust** (engl. *stochastic gradient descent*, *incremental gradient descent*, *online gradient descent*), podrazumeva da se ažuriranje parametara \mathbf{w} vrši iterativno na osnovu svakog pojedinačnog trening primera. U ovoj metodi, algoritam prolazi kroz trening skup, i za jedan trening primer vrši ažuriranje svih parametara \mathbf{w} na osnovu vrednosti gradijenta samo tog jednog primera. Pseudo-algoritam ovakvog postupka se može predstaviti na sledeći način:

```

foreach (i) //za svaki trening par  $(\mathbf{x}^{(i)}, y^{(i)})$ 
{
  foreach (j) //za svaki parametar  $w_j$ 
  {
     $w_j := w_j + \hat{\alpha}(y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})x_j^{(i)}$ 
  }
}

```

Prethodni postupak ažuriranja parametara treba ponoviti veći broj puta, kako bi se došlo do konačne konvergencije.

U realnim aplikacijama za klasifikaciju slika, trening skup podataka može sadržati milione trening slika [51]. U ovakvim aplikacijama, korišćenje prethodnog metoda se pokazalo neefikasnim, jer je za svaki trening podatak neophodno obaviti ažuriranje svih parametara \mathbf{w} (čiji je broj je takođe veoma veliki - reda veličine milion). Zato je kao alternativa metodi stohastičkog gradijentnog spusta, predložen drugačiji pristup, pod nazivom **grupni gradijentni spust** (engl. **batch gradient descent**). Ovakav pristup podrazumeva da se ažuriranje parametara $w_j; j = 0, 1, \dots, n$ vrši korišćenjem svih primera iz trening skupa “odjednom”. Ovde ćemo prvo izvesti opštu formulu parcijalnog izvoda funkcije greške koja obuhvata sve primere iz trening skupa, i koja predstavlja generalizaciju formule 4.6:

$$\begin{aligned}
\frac{\partial L(\mathbf{w})}{\partial w_j} &= \frac{\partial}{\partial w_j} \left(\frac{1}{M} \sum_{i=1}^M (h(\mathbf{x}^{(i)}, \mathbf{w}) - y^{(i)})^2 \right) = \frac{1}{M} \frac{\partial}{\partial w_j} \sum_{i=1}^M (\mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)})^2 \\
&= \frac{2}{M} \sum_{i=1}^M (\mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)}) x_j
\end{aligned} \tag{4.8}$$

Na osnovu toga se može izvesti formula za ažuriranje parametara modela:

$$\begin{aligned}
w_j &:= w_j - \alpha \frac{\partial L(\mathbf{w})}{\partial w_j}; j = 0, 1, \dots, n \\
w_j &:= w_j + \frac{2\alpha}{M} \sum_{i=1}^M (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)}) x_j; j = 0, 1, \dots, n \\
w_j &:= w_j + \hat{\alpha} \sum_{i=1}^M (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)}) x_j; j = 0, 1, \dots, n
\end{aligned} \tag{4.9}$$

Dakle, grupno ažuriranje parametara, u slučaju metode grupnog gradijentnog spusta, se može predstaviti seldećim pseudo-algoritmom:

```

foreach (j) // za svaki parametar  $w_j$ 
{

$$w_j := w_j + \hat{\alpha} \sum_{i=1}^M (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)}) x_j$$

}

```

Treba uočiti da se u prethodnoj formuli, prilikom ažuriranja w_j pojavljuje suma svih primera iz trening skupa $\sum_{i=1}^M(\dots)$. To znači da se prilikom ažuriranja jedog parametara w_j , računaju gradijenti svih primera iz celokupnog trening skupa podataka. Prethodni postupak ažuriranja parametara treba ponoviti više puta nad celokupnim trening skupom, da bi se došlo do konačne konvergencije. Grupno ažuriranje svih parametara je efikasnije u odnosu na stohastički gradijentni spust, ali je veoma zahtevno u pogledu memorijskih resursa. Osim toga, u pojedinim primenama celokupan trening skup nije unapred dostupan, već se novi trening primeri generišu tokom vremena. Npr. Google inteligentni sistem za prevenciju nepoželjne pošte se neprekidno adaptira tj. ažurira težine modela na osnovu novih tipova nepoželjnih poruka. U ovakvim situacijama, neophodno je dodatno ažurirati parametre modela na osnovu novih trening primera.

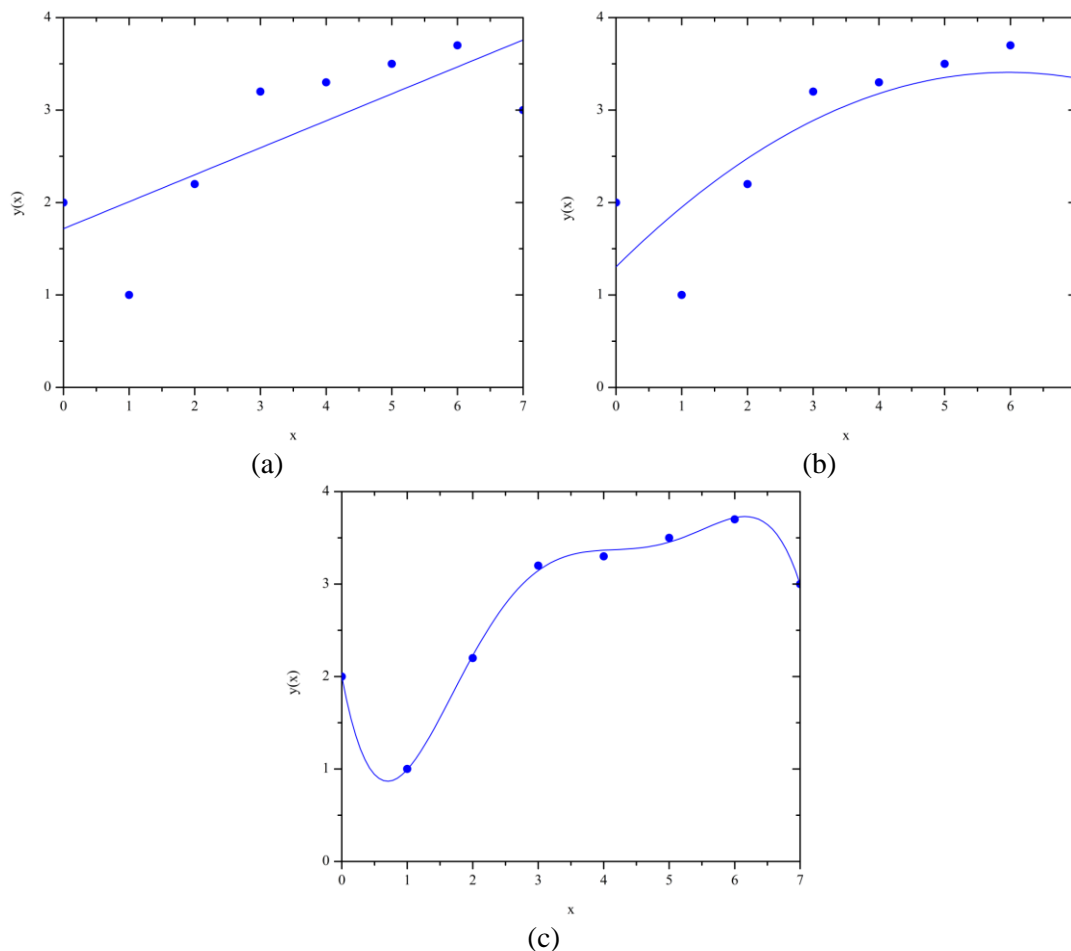
Da bi se prevazišli prethodni nedostaci, uvedena je metoda koja obavlja učenje modela korišćenjem malog podskupa trening podataka, tzv. **metoda gradijentnog spusta na mini-skupu** (engl. *mini-batch gradient descent*) [20]. Ova metoda se pokazala kao optimalan izbor u većini praktičnih aplikacija, jer kombinuje prednosti prethodne dve metode. S obzirom da je ažuriranje parametara \mathbf{w} računski zahtevna operacija, korišćenje mini-skupova je računski brže od stohastičkog gradijentnog spusta jer nema ažuriranja parametara nakon svakog trening primera. Do ubrzanja izračunavanja dolazi i zato što se za računanje gradijenata nad mini-skupovima koriste matična množenja koja se u modernim računarski sistemima jako efikasno implementiraju, pogotovo ako se koriste moderne grafičke kartice (GPU). S druge strane, metoda stohastičkog gradijentnog spusta, gde se težine ažuriraju nakon svakog primera iz trening skupa, obično brže konvergira ka optimalnom rešenju, jer nije neophodno računati gradijent svih primera da bi izvršili ažuriranje. Međutim, kod ove metode postoji veći rizik od pronalaženja lokalnog minimuma koji ne predstavlja optimalno rešenje.

Važno je uočiti da metode koje računaju gradijent nad podskupom trening podataka (mini-skup i grupni gradijentni spust), praktično unose šum u izračunavanje gradijenata, i svode se na aproksimaciju gradijenta skupa trening podataka. Taj šum je koristan u postupku optimizacije jer pomaže da postupak prevaziđe eventualni lokalni minimum koji se može pojaviti. Zato metoda gradijentnog spusta na mini-skupu predstavlja dobar kompromis između

memorijskih zahteva i brzine konvergencije, i preporučljivo je primenjivati je u situacijama kada je moguće.

4.2.2 Problem prevelikog ili nedovoljnog podešavanja modela

Razmotrićemo sada problem izbora prostora hipoteza, tj. definisanja parametarskog oblika funkcije $h(\mathbf{x}, \mathbf{w})$. Radi jednostavnosti pretpostavićemo da se vektor ulaznih podataka sastoji samo od jednog elementa, tj. $\mathbf{x} = [x_1] \equiv [1, x_1]^T$. Razmotrimo prvo slučaj ukoliko bismo odabrali jednostavan linearni oblik funkcije hipoteze $h(\mathbf{x}) = w_0 + w_1x_1$, za koju smo izračunali parametre $\mathbf{w} = [w_0, w_1]^T$ na osnovu trening skupa podataka. Grafički prikaz rezultata aproksimacije linearne funkcije na jednom trening skupu prikazuje Slika 16a. Sa slike se može uočiti da se većina podataka (tačaka) iz trening skupa ne nalazi na pretpostavljenoj pravoj liniji, što implicira da linearna funkcija ne predstavlja optimalan izbor za $h(\mathbf{x}, \mathbf{w})$. Ovo je primer **nedovoljnog podešavanja** modela (engl. underfitting), kod koga je hipoteza previše jednostavna tj. nije dovoljno dobro naučena na osnovu trening skupa.



Slika 16 Aproksimacija trening skupa podataka različitim funkcijama: a) linearna funkcija, b) kvadratna funkcija, c) polinom 5-og stepena.

Ukoliko bismo za aproksimaciju koristili kvadratnu funkciju hipoteze $h(\mathbf{x}) = w_0 + w_1x_1 + w_2x_1^2$ (Slika 16b), uočava se da ovakva funkcija znatno bolje aproksimira elemente trening skupa. Može se steći utisak da se daljim povećanjem stepena polinoma može doći do optimalne aproksimacione funkcije tj. funkcije hipoteze. Međutim, ako posmatrao primer polinoma petog stepena (Slika 16c), uočavamo da bez obzira što aproksimirana kriva prolazi kroz sve elemente trening skupa, ona neće rezultovati dobrim predviđanjem vrednosti za npr. $x_1 = 10$. Ovo je tipičan primer *preteranog podešavanja modela* (engl. *overfitting*), kod koga je model previše prilagođen podacima trening skupa, odnosno previše je kompleksan. Bez obzira što bi polinom petog stepena davao dobre rezultate na primerima koji se nalaze u trening skupu, ne treba očekivati da bismo dobili dobre rezultate na novim nepoznatim podacima prilikom testiranja. Drugim rečima, matematički model treniran na osnovu trening skupa, neće generalizovati dovoljno dobro primere izvan njega. Jedan od razloga za nastanak ovog problema je usled toga što vektor parametara \mathbf{w} nije jedinstven, tj. može postojati veći broj sličnih vektora parametara \mathbf{w} koji uz minimalnu grešku aproksimiraju funkciju hipoteze na trening skupu podataka.

Na osnovu prethodnih primera može se zaključiti da linearna funkcija, kao i polinomi visokog stepena, ne predstavljaju optimalan izbor funkcije hipoteze. Postavlja se pitanje, kako iskoristiti polinome nižeg stepena, a pri tome eliminisati probleme prevelikog podešavanja modela. Ovo se rešava tehnikom koja se naziva *regularizacijom*. Regularizacija je često korišćena tehnika u mašinskom učenju, a odnosi se na postupak pružanja dodatnih informacija funkciji greške kako bi se rešio problem šuma u trening podacima ili problem prevelikog podešavanja modela. Regularizacijom se sprečava da se model previše prilagodi trening skupu dozvoljavajući da greška na trening skupu ne bude minimalna, sa ciljem da greška na nepoznatim podacima bude što manja. Ovo se uobičajeno realizuje uvođenjem dodatnog *regularizacionog člana* u funkciju greške $L(\mathbf{w})$, na sledeći način:

$$L(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^M (h(\mathbf{x}^{(i)}, \mathbf{w}) - y^{(i)})^2 + \lambda \sum_j w_j^2 \quad 4.10$$

gde $\lambda > 0$ predstavlja *regularizacioni parametar* (engl. regularization term). Vrednost λ se može odrediti na osnovu trening skupa, primenom tehnika za unakrsnu validaciju modela. Prethodni oblik regularizacionog člana $\lambda \sum_j w_j^2$ se naziva kvadratna L^2 norma, koja ima za cilj da značajno poveća vrednost funkcije greške $L(\mathbf{w})$ za velike vrednosti težina, i na taj način „potisne“ ovakve vrednosti iz konačnog izbora. Kao rezultat, težine koje su izračunate nakon L^2 regularizacije će imati male vrednosti koje su gusto raspoređene. Druga uobičajena forma

regularizacionog člana je L1 norma, koja se zapisuje u obliku $\lambda \sum_j |w_j|$. Ovakav oblik regularizacije rezultira težinama koje su “rasplinute” i koje množenjem implicitno selektuju samo najreprezentativnije primere iz trening skupa. U praktičnim aplikacijama, ukoliko ne postoji jasna potreba za selekcijom samo pojedinih primera iz trening skupa, L2 regularizacija uobičajeno daje bolje rezultate u odnosu na L1 regularizaciju.

4.3 Klasifikacija

Problem klasifikacije se može svesti na linearnu regresiju pri čemu se vrši predviđanje diskretne izlazne vrednosti. Npr. u sistemima za vizuelnu kontrolu saobraćaja, za zadatu ulaznu sliku sa kamere (tj. jedan blok slike) moguće je automatski odrediti da li sadrži pešaka ili ne [17]. Ukoliko slika sadrži pešaka, izlaz automatskog algoritma za klasifikaciju treba da bude “1”, a ukoliko ne sadrži, izlaz je “-1”. Ovakav problem, gde predviđeni izlaz može imati vrednost samo iz diskretnog skupa vrednosti, naziva se *klasifikacioni problem*. U konkretnom slučaju, gde vršimo klasifikaciju na pešake i “ne-pešake”, diskretan skup izlaznih vrednosti sadrži samo dve vrednosti $y \in \{+1, -1\}$. Ovakav problem se naziva *binarna klasifikacija*. Pri tome, vrednost “1” se u pojedinoj literaturi javlja pod nazivom pozitivna klasa, a vrednost “-1” pod nazivom negativna klasa. U pojedinim slučajevima se za označavanje negativne klase koristi vrednost 0. Ukoliko skup mogućih izlaznih vrednosti sadrži više od dva elementa, radi se o *više klasnoj klasifikaciji* (engl. *multiclass classification*).

4.4 Binarna klasifikacija

Postupak binarne klasifikacije se na najjednostavniji način može realizovati na osnovu algebarskog znaka predviđene izlazne vrednosti linearne regresije. To znači da se oznaka rezultujuće klase može dobiti nakon primene sledeće funkcije g nad izračunatom vrednošću hipoteze:

$$g(h(\mathbf{x}, \mathbf{w})) = \begin{cases} +1 & , h(\mathbf{x}, \mathbf{w}) \geq 0 \\ -1 & , h(\mathbf{x}, \mathbf{w}) < 0 \end{cases} \quad 4.11$$

Ukoliko se za binarnu klasifikaciju koristi linearna hipoteza oblika $h(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$, onda će ulazni podatak \mathbf{x} biti ispravno klasifikovan ukoliko je

$$y = g(\mathbf{w}^T \mathbf{x}) \quad 4.12$$

Pretpostavljajući da $y \in \{+1, -1\}$, ovo praktično znači da obe vrednosti y i $\mathbf{w}^T \mathbf{x}$ treba da budu istog znaka, što je ekvivalentno sa:

$$y\mathbf{w}^T \mathbf{x} > 0 \quad 4.13$$

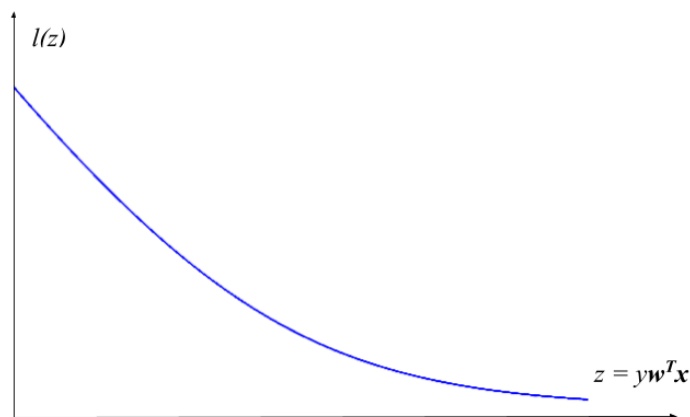
Veličina $y\mathbf{w}^T \mathbf{x}$ u prethodnom izrazu je veoma značajna za postupak binarne klasifikacije, i uobičajeno se naziva *margin*a primera (\mathbf{x}, y) . Prethodni izraz za marginu se može tumačiti na sledeći način: veća pozitivna vrednost $\mathbf{w}^T \mathbf{x}$ prouzrokuje veću verovatnoću da je na izlazu pozitivna klasa (“+1”), i slično za negativne vrednosti. Dakle, *veća vrednost margine* ($y\mathbf{w}^T \mathbf{x}$) *prouzrokuje veću verovatnoću tačne klasifikacije*.

Da bismo definisali funkciju greške hipoteze, počecemo od definisanja greške i -tog trening primera $(\mathbf{x}^{(i)}, y^{(i)})$, čija je margin a označena kao $z^{(i)} = y^{(i)}\mathbf{w}^T \mathbf{x}^{(i)}$. Ako ovu “delimičnu” grešku označimo sa $l(z^{(i)}) = l(y^{(i)}\mathbf{w}^T \mathbf{x}^{(i)})$, onda je kompletna funkcija greške koju treba minimizovati predstavljena kao:

$$L(\mathbf{w}) = \frac{1}{M} \sum_{i=1}^M l(z^{(i)}) = \frac{1}{M} \sum_{i=1}^M l(y^{(i)}\mathbf{w}^T \mathbf{x}^{(i)}) \quad 4.14$$

Prilikom izbora funkcije “parcijalne” greške $l(z^{(i)})$, logika kojom se rukovodimo bazirana je na pretpostavci da u idealnom slučaju margin a treba da ima vrednost $z^{(i)} = y^{(i)}\mathbf{w}^T \mathbf{x}^{(i)} > 0$ za svaki trening par $(\mathbf{x}^{(i)}, y^{(i)})$. Otuda, greška $l(z^{(i)})$ treba da ima malu vrednost ukoliko je margin a velika, i obratno, greška treba da je velika ukoliko je margin a mala (Slika 17). Matematički se ovo se može zapisati kao:

$$\begin{aligned} l(z) &\rightarrow 0, & \text{ako } z &\rightarrow \infty \\ l(z) &\rightarrow \infty, & \text{ako } z &\rightarrow -\infty \end{aligned} \quad 4.15$$



Slika 17 Oblik funkcije greške

U nastavku su dati primeri funkcija greške koje zadovoljavaju ove uslove, a koje se često koriste za binarnu klasifikaciju u oblasti mašinskog učenja.

- Logaritamska funkcija greške:

$$l(z) = \log(1 + e^{-z})$$

- Zglobna funkcija greške (engl. hinge loss):

$$l(z) = \max\{1 - z, 0\}$$

- Eksponencijalna funkcija greške:

$$l(z) = e^{-z}$$

Izbor konkretne funkcije greške zavisi od problema koji se rešava, kao i od tehnike mašinskog učenja koja se primenjuje. Tako npr. metoda logističke regresije koristi logaritamsku funkciju greške, metoda potpornih vektora (SVM) koristi zglobnu funkciju greške [1], dok se eksponencijalna funkcija greške koristi kod metode Boosting [67]. Treba primetiti da sve prethodne funkcije greške zadovoljavaju osobine da su neprekidne i konveksne, što je neophodno za izračunavanje gradijenta u postupku minimizacije ukupne funkcije greške.

4.4.1 Probabilistička interpretacija logističke regresije

Problem binarne klasifikacije se može definisati i na osnovu teorije verovatnoće, polazeći od prethodno definisane linearne regresije. Ukoliko želimo da postupak linearne regresije prilagodimo za rešavanje problema binarne klasifikacije, počecemo od definisanja adekvantnog prostora hipoteza. Pretpostavimo da izlaz može imati jednu diskretnu vrednost iz skupa $y \in \{0, 1\}$. Jasno je da prethodno korišćena linearna hipoteza ne daje izlaz u ovom opsegu, pa je hipotezu neophodno prilagoditi problemu klasifikacije. Ovo se može obaviti tako što se izlaz linearne hipoteze “propusti” kroz nelinearnu *sigmoidnu (logističku) funkciju*. Ova funkcija će izlazne vrednosti ograničiti na realne brojeve iz opsega $[0, 1]$. Nakon toga treba obaviti kvantizaciju rezultata na jednu od dve vrednosti iz skupa $\{0, 1\}$, jednostavnim upoređivanjem sa unapred definisanim pragom. Prethodni postupak klasifikacije se naziva *logistička regresija*.

Sigmoidna (logistička) funkcija $g(z)$ ima oblik

$$g(z) = \frac{1}{1 + e^{-z}} \quad 4.16$$

U slučaju da koristimo linearnu početnu hipotezu, klasifikaciona hipoteza postaje

$$h(\mathbf{x}, \mathbf{w}) = g(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}} \quad 4.17$$

Treba uočiti da će izlaz prethodne funkcije hipoteze biti u opsegu $[0, 1]$, pa je ovu funkciju moguće matematički interpretirati kao funkciju verovatnoće, na sledeći način:

$$\begin{aligned} P(y = 1 | \mathbf{x}, \mathbf{w}) &= h(\mathbf{x}, \mathbf{w}) \\ P(y = 0 | \mathbf{x}, \mathbf{w}) &= 1 - h(\mathbf{x}, \mathbf{w}) \end{aligned} \quad 4.18$$

gde je sa $P(y = 1 | \mathbf{x}, \mathbf{w})$ označena verovatnoća pripadnosti klasi $y = 1$, dok je sa $P(y = 0 | \mathbf{x}, \mathbf{w})$ označena verovatnoća pripadnosti klasi $y = 0$, za zadati ulazni vektor \mathbf{x} i težine \mathbf{w} .

Cilj postupka logističke regresije je odrediti vrednosti težina \mathbf{w} , tako da verovatnoća $P(y = 1 | \mathbf{x}, \mathbf{w}) = h(\mathbf{x}, \mathbf{w})$ bude što veća za $\mathbf{x}^{(i)}$ koji pripada klasi $\mathbf{1}$ (tj. $y^{(i)} = 1$). Nasuprot tome, vrednost $h(\mathbf{x}, \mathbf{w})$ treba bude što manja kada $\mathbf{x}^{(i)}$ pripada klasi $\mathbf{0}$. Nakon ovoga će biti jednostavno odrediti prag za poređenje, koji će celokupan prostor rezultata podeliti na dve oblasti. Funkcija greške koja zadovoljava ove osobine i koja se uobičajeno koristi za ovu namenu, bazirana je na *unakrsnoj entropiji* i može se predstaviti kao:

$$L(\mathbf{w}) = - \sum_i \left(y^{(i)} \log \left(h(\mathbf{x}^{(i)}, \mathbf{w}) \right) + (1 - y^{(i)}) \log \left(1 - h(\mathbf{x}^{(i)}, \mathbf{w}) \right) \right) \quad 4.19$$

Treba uočiti da će u prethodno definisanoj funkciji greške, jedan od dva člana uvek biti anuliran za svaki od trening primera (jer je $y^{(i)} = 0$ ili $y^{(i)} = 1$). Za minimizaciju prethodne funkcije i određivanje optimalnih parametara \mathbf{w} , mogu se koristiti prethodno opisane metode gradijentnog spusta, kao i u slučaju linearne regresije. Nakon određivanja parametara \mathbf{w} , imamo u potpunosti definisanu hipotezu $h(\mathbf{x}, \mathbf{w})$, i moguće je klasifikovati bilo koji ulazni podatak $\mathbf{x}^{(i)}$ na osnovu izračunate vrednosti $h(\mathbf{x}^{(i)}, \mathbf{w})$. Konačna klasa se može odrediti poređenjem sa pragom, tj. ukoliko je $h(\mathbf{x}^{(i)}, \mathbf{w}) \geq 0.5$ onda se ulazu $\mathbf{x}^{(i)}$ dodeljuje klasa $\mathbf{1}$, dok se u suprotnom dodeljuje klasa $\mathbf{0}$. Ovo je ekvivalentno uslovu da je $P(y = 1 | \mathbf{x}, \mathbf{w}) \geq P(y = 0 | \mathbf{x}, \mathbf{w})$.

4.5 Višeklasna klasifikacija

Višeklasna klasifikacija predstavlja generalizaciju postupka binarne klasifikacije. Za razliku od binarne klasifikacije gde je izlazna labela mogla imati samo jednu od dve vrednosti $y \in \{0, 1\}$, u slučaju višeklasne klasifikacije izlazna labela može imati jednu od C vrednosti $y \in \{1, 2, \dots, C\}$. Dakle, u trening skupu $T = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(M)}, y^{(M)})\}$, vrednost $y^{(i)} \in \{1, 2, \dots, C\}$. Primer višeklasne klasifikacije u $C = 10$ klasa je postupak automatske klasifikacije slika cifara.

Da bismo formulisali metodu **višestruke logističke regresije** koja se primenjuje u slučaju klasifikacije u više klasa, neophodno je prvo definisati adekvatnu funkciju hipoteze. Za ulazni test vektor \mathbf{x} , cilj funkcije hipoteze je da predvidi verovatnoće pripadnosti svakoj od klasa $P(y = j | \mathbf{x}, \mathbf{w})$, za svako $j = 1, \dots, C$. Otuda će nova hipoteza biti C -dimenzioni vektor, koji daje C verovatnoća pripadnosti svakoj od klasa:

$$h(\mathbf{w}, \mathbf{x}) = \begin{bmatrix} P(y = 1 | \mathbf{x}, \mathbf{w}) \\ P(y = 2 | \mathbf{x}, \mathbf{w}) \\ \vdots \\ P(y = C | \mathbf{x}, \mathbf{w}) \end{bmatrix} \quad 4.20$$

U slučaju višeklasne klasifikacije, za izračunavanja verovatnoće pripadnosti svakoj od klasa $P(y = j | \mathbf{x}, \mathbf{w})$, uobičajeno se koristi tzv. *softmax funkcija (tj. normalizovana eksponencijalna funkcija)* [68]. Ova funkcija predstavlja generalizaciju sigmoidne funkcije i osim osobine da proizvoljne realne elemente vektora mapira na opseg $[0, 1]$, ima dodatnu osobinu da će suma elemenata izlaznog vektora biti 1. Na ovaj način, elementi vektora će u matematički smislu predstavljati raspodelu verovatnoća. Za elemente ulaznog vektora $\mathbf{z} = [z_1, \dots, z_j, \dots, z_C]^T$, softmax funkcija je zadata kao:

$$g(z_j) = \frac{e^{z_j}}{\sum_{k=1}^C e^{z_k}}; \quad j = 1, \dots, C \quad 4.21$$

Ovako definisana softmax funkcija se koristi u različitim metodama višeklasne klasifikacije. Konkretno, u slučaju višestruke (višeklasne) logističke regresije, softmax funkcija se primenjuje nad izlazom linearne kombinacije ulaznog vektora \mathbf{x} i težinskog vektora \mathbf{w} :

$$P(y = j | \mathbf{x}, \mathbf{w}) = \frac{e^{(\mathbf{w}^{(j)T} \mathbf{x})}}{\sum_{k=1}^C e^{(\mathbf{w}^{(k)T} \mathbf{x})}}; \quad j = 1, \dots, C \quad 4.22$$

U prethodnoj formuli treba uočiti da su težinski parametri zadati kao skup vektora, gde $\mathbf{w}^{(j)}$ predstavlja j -ti vektor težina. Na osnovu ovoga, hipoteza dobija oblik:

$$g(z_j) = \frac{e^{z_j}}{\sum_{k=1}^C e^{z_k}}; \quad j = 1, \dots, C \quad 4.23$$

Za minimizaciju prethodne funkcije i određivanje optimalnih parametara $\mathbf{w}^{(j)}$, mogu se koristiti ranije opisane metode gradijentnog spusta, koje sada treba prilagoditi višedimenzionalnim podacima.

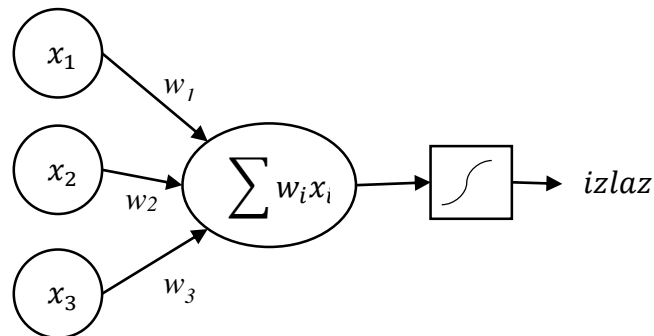
4.6 Veštačke neuronske mreže

Veštačke neuronske mreže je moguće veoma efikasno iskoristiti za definisanje složenih nelinearnih funkcija hipoteze. Na ovaj način, primena neuronskih mreža postaje od suštinske važnosti za predikciju i/ili klasifikaciju podataka.

Veštačka neuronska mreža (engl. artificial neural network), ili skraćeno neuronska mreža, predstavlja matematički model za procesiranje informacija inspiriran načinom na koji ljudski mozak obrađuje informacije tj. električne impulse. Neuronske mreže se sastoje od jednostavnih

elementa (neurona), međusobno povezanih vezama koje se u ljudskom mozgu nazivaju sinapse. U ljudskom mozgu ima preko 80 milijardi neurona povezanih sa preko 10^{14} sinapsi. Proces učenja u ljudskom mozgu podrazumeva podešavanje sinapsi, a slično tome se uči i podešavanje parametara (tj. težina) u veštačkoj neuronskoj mreži.

Osnovni gradivni element neuronske mreže je veštački perceptron, odnosno veštački neuron. Matematički model perceptrona je razvio Frank Rosenblatt 1958. godine [69], na osnovu analogije sa biološkim neuronom ljudskog mozga.



Slika 18 Model veštačkog neurona sa tri ulaza: x_1, x_2, x_3 . Težinski faktori su označeni sa w_1, w_2, w_3 . U opštem slučaju, broj ulaza može biti proizvoljan.

Rosenblatt je definisao pravilo po kojem se računa izlaz veštačkog neurona tako što se težinska suma $w_1x_1 + w_2x_2 + w_3x_3 = \sum_i w_ix_i$ dovodi na ulaz nelinearne **aktivacione funkcije** $g(\sum_i w_ix_i)$. Najjednostavnija aktivaciona funkcija se definiše tako da daje izlaz +1 ukoliko je argument veći od zadatog praga, a izlaz -1 ako je argument manji od praga. Izlaz neurona onda postaje:

$$g\left(\sum_i w_ix_i\right) = \begin{cases} +1 & , \sum_i w_ix_i \geq Th \\ -1 & , \sum_i w_ix_i < Th \end{cases} \quad 4.24$$

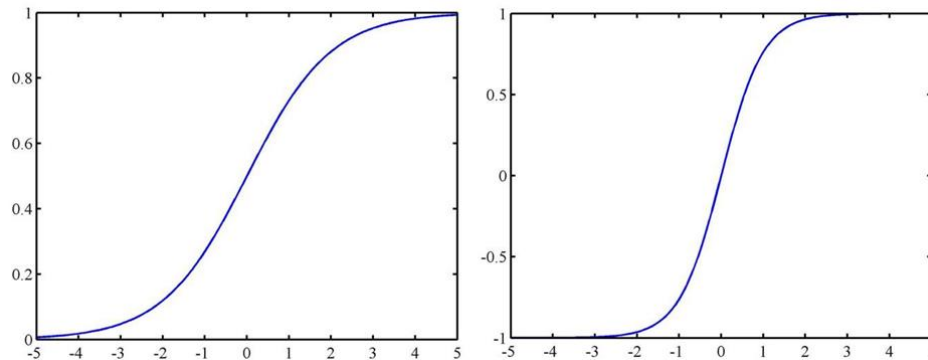
Učenje tj. treniranje prethodnog modela neurona se zapravo svodi na podešavanje težina w_i i praga Th , čime se postižu željene vrednosti izlaza. Ovakav model neurona zapravo predstavlja primer **linearnog binarnog klasifikatora** (opisanog u Poglavlju 4.4), gde izlaz može imati jednu od ukupno dve diskretne vrednosti.

Najčešći oblik aktivacione funkcije neurona je **sigmoidna (logistička) funkcija**:

$$g(z) = \frac{1}{1 + e^{-z}} \quad 4.25$$

Osim sigmoidne funkcije čiji je izlaz u opsegu $[0, 1]$, često se kao aktivaciona funkcija koristi i tangens hiperbolički čiji je opseg $[-1, 1]$:

$$g(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad 4.26$$



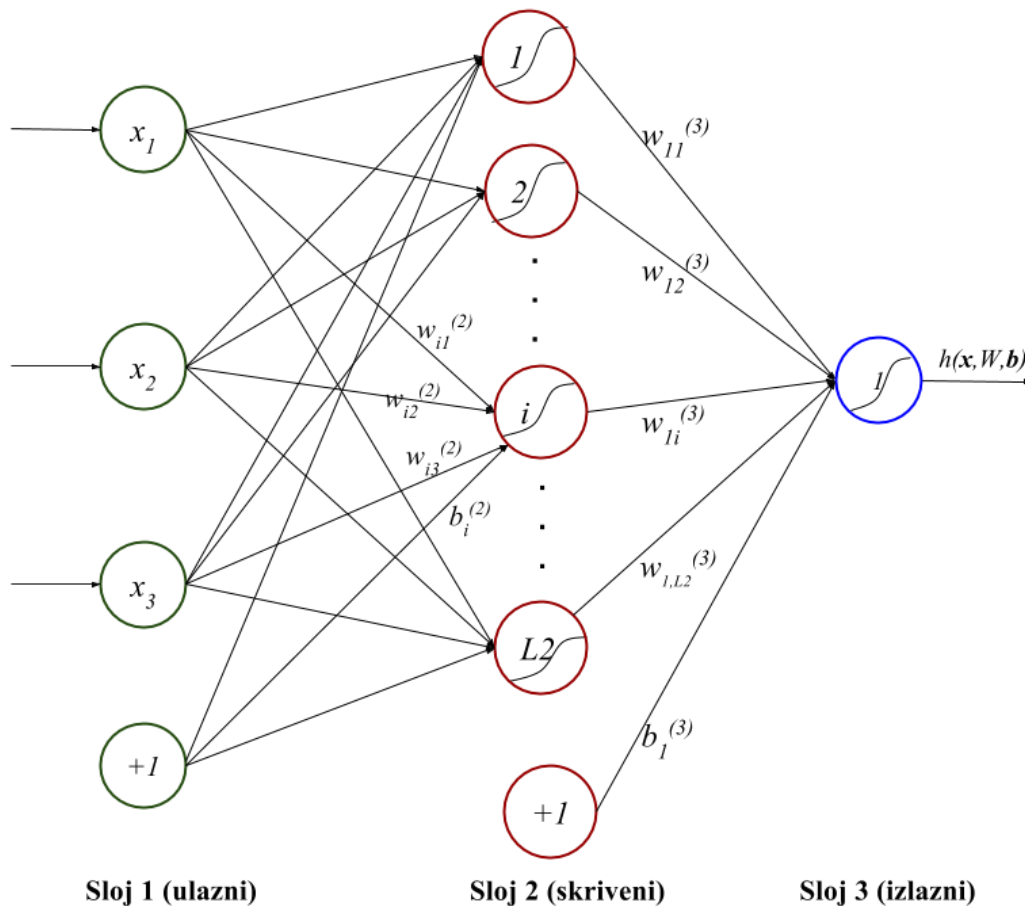
Slika 19 Aktivacione funkcije veštačkog neurona a) Sigmoidna (logistička) funkcija, b) Tangens hiperbolički.

Novija istraživanja u oblasti višeslojnih neuronskih mreža [44], uobičajeno koriste efikasnu aktivacionu funkciju, pod nazivom *ispravljena linearna funkcija* (engl. *rectified linear function*):

$$g(z) = \max(0, x) = \begin{cases} x & , x \geq 0 \\ 0 & , x < 0 \end{cases} \quad 4.27$$

Ova aktivaciona funkcija je pokazala veliku uspešnost u kompleksnim dubokim neuronskim mrežama (engl. Deep Neural Networks), uz veliku efikasnost izračunavanja [44]. Treba uočiti da ova funkcija, za razliku od sigmoidne i tanh funkcije, nije ograničena i nije diferencijabilna.

Neuronske mreže se formiraju međusobnim povezivanjem većeg broja neurona, tako što se izlaz jednog neurona poveže na ulaz narednog. Na ovaj način se mogu dobiti složene neuronske mreže različitih struktura. Razmotrimo primer jednostavne neuronske mreže koja je prikazana dijagramom na narednoj slici.



Slika 20 Primer veštačke neuronske mreže sa jednim ulaznim slojem, jednim skrivenim slojem i jednim izlazom.

Neuroni su grupisani u vertikalnim slojevima, pri čemu neuroni unutar jednog sloja nisu međusobno povezani. Sloj neurona koji se nalazi najdalje levo (Sloj 1), naziva se ulazni sloj neurona i u konkretnom slučaju sadrži 3 ulazna čvora. Čvorovi označeni sa „+1“ predstavljaju pomeraj (engl. bias), i oni se ne broje prilikom opisivanja mreže. Srednji sloj neurona (Sloj 2) se naziva skriveni sloj (engl. hidden layer). U konkretnom primeru postoji samo jedan skriveni sloj od 3 neurona, dok u opštem slučaju može biti veći broj skrivenih slojeva sa različitim brojem neurona. Napominjemo da se u skrivenom sloju podrazumeva nelinearna aktivaciona funkcija $g(z)$ koja se ne crta u posebnom bloku, već se podrazumeva u okviru kruga koji prikazuje jedan skriveni neuron. Najdalji desni sloj (Sloj 3), je izlazni sloj neurona. Aktivaciona funkcija se podrazumeva i za neurone izlaznog sloja. Za ovu neuronsku mrežu možemo reći da ima ukupno 3 sloja. U pojedinoj literaturi je uobičajeno da se navodi samo broj skrivenih slojeva, pa bi se za prethodnu neuronsku mrežu reklo da ima 1 (skriveni) sloj.

Neka sa $w_{ij}^{(l)}$ označimo parametar (težinu) veze između čvora i iz sloja (l) , i čvora j iz sloja $(l - 1)$. Treba obratiti pažnju na redosled indeksa ij , kao i na to da se oznaka sloja (l)

koristi za sve ulazne težine ovog sloja. Dakle, težine $w_{i1}^{(l)}$, $w_{i2}^{(l)}$, $w_{i3}^{(l)}$, predstavljaju ulazne težine neurona i u sloju (l) . Napominjemo da u ovakvoj notaciji, oznaka sloja (l) nema nikakvih sličnosti sa indeksom trening primera iz objašnjenja linearne regresije. Sa $b_j^{(l)}$ je označena težina pomeraja (engl. bias) na ulazu čvora j u sloju (l) . Skup svih težina iz svih slojeva predstavlja ujedno i skup parametara neuronske mreže, koji potpuno definišu izlaz mreže, tj. funkciju hipoteze. Ovo praktično znači da je kod neuronskih mreža, funkcija hipoteze $h(\mathbf{x}, W, \mathbf{b})$ određena izlazom mreže.

Ako sa $z_j^{(l)}$ označimo izlaz j -tog neurona sloja (l) , onda izlaze iz sloja (2) u pojednostavljenoj neuronskoj mreži koju prikazuje Slika 20, možemo predstaviti kao:

$$\begin{aligned} z_1^{(2)} &= g(w_{11}^{(2)} x_1 + w_{12}^{(2)} x_2 + w_{13}^{(2)} x_3 + b_1^{(2)}), \\ z_2^{(2)} &= g(w_{21}^{(2)} x_1 + w_{22}^{(2)} x_2 + w_{23}^{(2)} x_3 + b_2^{(2)}), \\ z_3^{(2)} &= g(w_{31}^{(2)} x_1 + w_{32}^{(2)} x_2 + w_{33}^{(2)} x_3 + b_3^{(2)}). \end{aligned} \quad 4.28$$

U opštem slučaju, neuronske mreže mogu imati veći broj skrivenih slojeva i više izlaza, i ovakve mreže se nazivaju **višeslojne neuronske mreže**.

U opštem slučaju, sloj (k) može imati proizvoljan broj neurona označenih kao L_k . Pogodno je sve izlaze iz sloja 2 zapisati u vektorskom obliku $\mathbf{z}^{(2)} = [z_1^{(2)}, z_2^{(2)}, \dots, z_{L_2}^{(2)}]^T$, a težine $w_{ij}^{(2)}$ na ulazu drugog sloja smestiti u matricu $W^{(2)}$. Onda se prethodne jednačine 4.28 mogu napisati u matričnoj formi:

$$\mathbf{z}^{(2)} = g(W^{(2)} \mathbf{x} + \mathbf{b}^{(2)}) \quad 4.29$$

Na sličan način se može predstaviti $\mathbf{z}^{(3)}$ vektor izlaza iz sloja 3 (izlazni sloj), pri čemu ulazi ovog sloja predstavljaju zapravo izlaze prethodnog sloja $\mathbf{z}^{(2)}$:

$$\mathbf{z}^{(3)} = g(W^{(3)} \mathbf{z}^{(2)} + \mathbf{b}^{(3)}) \quad 4.30$$

U opštem slučaju, zaključuje se da vektor izlaza $\mathbf{z}^{(l)}$ iz sloja (l) možemo izračunati na osnovu vektora izlaza prethodnog sloja $\mathbf{z}^{(l-1)}$ i matrice težina $W^{(k)}$ za veze koja ulaze u sloj l :

$$\mathbf{z}^{(l)} = g(W^{(l)} \mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}) \quad 4.31$$

Izlaz poslednjeg L -tog sloja neurona predstavlja zapravo funkciju hipoteze celokupne neuronske mreže $h(\mathbf{x}, W, \mathbf{b})$:

$$h(\mathbf{x}, W, \mathbf{b}) = \mathbf{z}^{(L)} = g(W^{(L)} \mathbf{z}^{(L-1)} + \mathbf{b}^{(L)}) \quad 4.32$$

Neuronske mreže kod kojih se izlaz dobija sukcesivnim proračunom izlaza svakog od skrivenih slojeva redom $\mathbf{z}^{(2)}$, $\mathbf{z}^{(3)}$, $\mathbf{z}^{(4)}$, itd. korišćenjem jednačine 4.31, nazivaju se **višeslojne neuronske mreže sa prostiranjem unapred** (engl. *feedforward multilayer neural networks*). U ovakvim mrežama ne postoje nikakve zatvorene petlje niti povratne sprege.

Osim za rešavanje regresionih problema, višeslojne neuronske mreže se mogu ravnopravno koristiti i za rešavanje klasifikacionih problema. U slučaju binarne klasifikacije, moguće je upotrebiti mrežu sa jednim izlaznim neuronom, čije izlazne vrednosti (nakon primene npr. sigmoidalne aktivacione funkcije) mogu biti $y = 0$ za prvu klasu, ili $y = 1$ za drugu klasu. U slučaju klasifikacije u više klasa ($C > 2$), pogodno je koristiti mreže sa onoliko izlaznih neurona, koliko klasa postoji. Da bi bilo moguće treniranje, izlazni vektor u ovakvim slučajevima treba predstaviti kao binarni vektor, gde postoji samo jedna vrednost različita od nule koja određuje indeks klase kojoj ulazni podatak pripada. Npr. ukoliko vršimo klasifikaciju u $C = 5$ klasa, izlazni vektor $[0, 1, 0, 0, 0]$ će označiti pripadnost klasi na poziciji 2. Izračunate izlaze neuronske mreže, koji su realni brojevi, treba preslikati na binarni vektor, tako što će vrednost 1 biti dodeljena samo maksimalnom izlazu, dok će svi ostali dobiti vrednost 0. Npr. ukoliko je izlaz neuronske mreže predstavljen kao vektor $[0.2; 0.4; 0.1; 0.3; 0.1]$, onda će maksimalna vrednost 0.4 odrediti preslikavanje na vektor $[0, 1, 0, 0, 0]$, što označava pripadnost klasi 2.

4.6.1 Treniranje (obučavanje) neuronske mreže propagacijom unazad

Postupak treniranja (obučavanja) mreže omogućuje da se odrede parametri W i \mathbf{b} , na osnovu skupa trening parova $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$. Treba uočiti da izlaz neuronske mreže u opštem slučaju može biti vektor koji se sastoji od većeg broja elemenata, označen sa $\mathbf{y}^{(i)}$.

Funkcija greške, ukoliko se koristi srednje kvadratna greška, može se predstaviti kao:

$$L(W, \mathbf{b}) = \frac{1}{M} \sum_{i=1}^M (h(\mathbf{x}^{(i)}, W, \mathbf{b}) - \mathbf{y}^{(i)})^2 \quad 4.33$$

Ako uključimo i ranije opisani regularizacioni član, onda funkcija greške postaje:

$$L(W, \mathbf{b}) = \frac{1}{M} \sum_{i=1}^M (h(\mathbf{x}^{(i)}, W, \mathbf{b}) - \mathbf{y}^{(i)})^2 + \lambda \sum_l \sum_i \sum_j (w_{ij}^{(l)})^2 \quad 4.34$$

Podsetimo da regularizacioni član ima za cilj da spreči preveliko prilagođavanje modela trening podacima kroz ograničavanje amplituda težinskih parametara. Regularizacioni parametar λ kontroliše relativni značaj celokupnog regularizacionog člana. Kao i u slučaju

linearne regresije/klasifikacije, postupak treniranja ima za cilj da odredi optimalne parametre W i \mathbf{b} , za koje $L(W, \mathbf{b})$ ima minimalnu vrednost. Treniranje se vrši primenom adekvatnog optimizacionog algoritma, kao što je metoda gradijentnog spusta. Treba obratiti pažnju da inicijalizaciju početnih vrednosti parametara W i \mathbf{b} , treba uraditi dodeljivanjem nasumičnih malih vrednosti bliskih nuli, a ne postavljanjem svih vrednosti parametara na nulu ili drugu konstantnu vrednost. Ukoliko bi svi parametri imali istu početnu vrednost, onda bi izlazi svih neurona jednog skrivenog sloja predstavljali (tj. optimizovali) istu funkciju za ulaz \mathbf{x} . Napominjemo da funkcija $L(W, \mathbf{b})$ u opštem slučaju ne mora biti konveksna, pa teorijski postoji mogućnost da se metoda gradijentnog spusta zaustavi u lokalnom minimumu. Međutim, ovaj problem se uspešno prevazići odgovarajućim izborom početnih parametara i adekvatnim algoritmom optimizacije.

Metoda gradijentnog spusta za određivanje parametara W, \mathbf{b} , na osnovu $L(W, \mathbf{b})$ svodi se na ažuriranje sledećih vrednosti u jednoj iteraciji:

$$\begin{aligned} W_{ij}^{(l)} &:= W_{ij}^{(l)} - \alpha \frac{\partial L(W, \mathbf{b})}{\partial W_{ij}^{(l)}} \\ b_i^{(l)} &:= b_i^{(l)} - \alpha \frac{\partial L(W, \mathbf{b})}{\partial b_i^{(l)}}. \end{aligned} \tag{4.35}$$

U prethodnim formulama ključni deo predstavlja izračunavanje parcijalnih izvoda, koji se može efikasno realizovati primenom algoritma *propagacije unazad* (*engl. backpropagation algorithm*). Za trening skup od M primera, parcijalni izvodi se mogu predstaviti kao:

$$\begin{aligned} \frac{\partial L(W, \mathbf{b})}{\partial W_{ij}^{(l)}} &= \frac{\partial}{\partial W_{ij}^{(l)}} \frac{1}{M} \sum_{i=1}^M (h(\mathbf{x}^{(i)}, W, \mathbf{b}) - \mathbf{y}^{(i)})^2 \\ &= \left[\frac{2}{M} \sum_{i=1}^M \frac{\partial h(\mathbf{x}^{(i)}, W, \mathbf{b})}{\partial W_{ij}^{(l)}} \right] + \lambda W_{ij}^{(l)} \end{aligned} \tag{4.36}$$

$$\frac{\partial L(W, \mathbf{b})}{\partial b_i^{(l)}} = \frac{\partial}{\partial b_i^{(l)}} \frac{1}{M} \sum_{i=1}^M (h(\mathbf{x}^{(i)}, W, \mathbf{b}) - \mathbf{y}^{(i)})^2 = \frac{2}{M} \sum_{i=1}^M \frac{\partial h(\mathbf{x}^{(i)}, W, \mathbf{b})}{\partial b_i^{(l)}}$$

Algoritmom propagacije unazad, za jedan primer iz trening skupa $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, prvo obavlja “propagaciju unapred“ kako bi se izračunali izlazi svih neuron u mreži, uključujući i izlazni sloj L koji predstavlja vrednost funkcije hipoteze $h(\mathbf{x}^{(i)}, W, \mathbf{b})$. Zatim se za svaki neuron i u sloju (l) , računa delimična greška $\delta_i^{(l)}$ koja određuje koliko taj neuron ima uticaja na ukupnu grešku izlaza. Za neurone u izlaznom sloju (L) , moguće je izračunati delimičnu grešku $\delta_i^{(L)}$ na

osnovu razlike izlaznih vrednosti neurona i stvarnih vrednosti iz trening skupa. Za neurone u skrivenom sloju, delimična greška $\delta_i^{(l)}$ se može izračunati kao težinska suma čvorova čiji su ulazi $z_i^{(l)}$.

Konkretno, za jedan trening primer označen kao (\mathbf{x}, \mathbf{y}) , algoritam propagacije unazad se može predstaviti kroz sledeće korake:

- 1) Kroz propagaciju unapred izračunati izlaze $\mathbf{z}^{(l)}$ za sve slojeve neuronske mreže (jednačina 4.31), uključujući i izlazni sloj L .
- 2) Za izlazni sloj L , izračunati delimičnu grešku za svaki čvor i u ovom sloju:

$$\delta_i^{(L)} = \frac{\partial}{\partial z_i^{(L)}} (h(\mathbf{x}, \mathbf{W}, \mathbf{b}) - \mathbf{y})^2 = 2(z_i^{(L)} - \mathbf{y}_i) \frac{\partial \mathbf{z}^{(L)}}{\partial z_i^{(L)}} \quad 4.37$$

- 3) Za svaki sloj $l = L - 1, L - 2, L - 3, \dots, 2$, izračunati za svaki čvor i :

$$\delta_i^{(l)} = \left(\sum_{j=1}^{n_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) \frac{\partial \mathbf{z}^{(l)}}{\partial z_i^{(l)}} \quad 4.38$$

gde je n_l broj neurona u l -tom sloju.

- 4) Konačno je moguće izračunati parcijalne izvode:

$$\begin{aligned} \frac{\partial L(\mathbf{W}, \mathbf{b})}{\partial W_{ij}^{(l)}} &= z_j^{(l)} \delta_i^{(l+1)} \\ \frac{\partial L(\mathbf{W}, \mathbf{b})}{\partial b_i^{(l)}} &= \delta_i^{(l+1)} \end{aligned} \quad 4.39$$

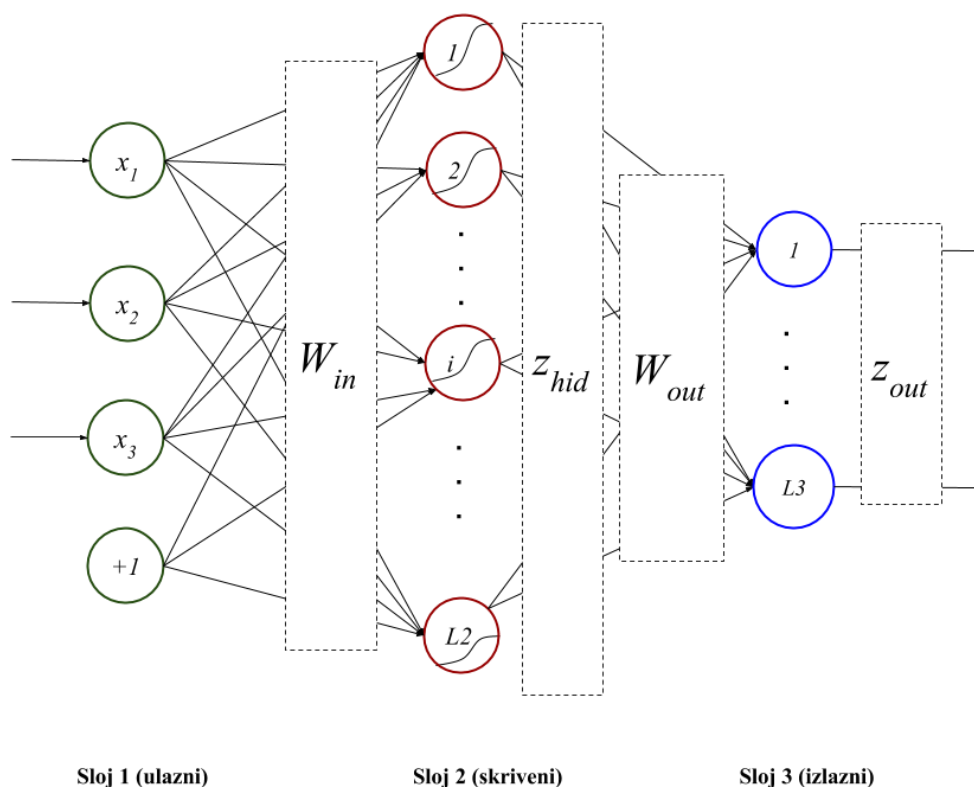
4.7 Metoda ekstremno brzog učenja (ELM)

Primena prethodno opisanog algoritma propagacije unazad za treniranje velikih neuronskih mreža, pokazala je relativno veliku računsku složenost postupka. Ovo se pokazalo kao važan ograničavajući faktor za širu primenu neuronskih mreža. Zato su razmatrane različite tehnike za ubrzanje postupka treniranja neuronskih mreža. Jedna od novih metoda koja uvodi ekstremno brz postupak treniranja neuronskih mreža, jeste metoda ekstremno brzog učenja (*engl. Extreme Learning Machines – ELM*) [70–72]. ELM uvodi novi postupak treniranja jednoslojne neuronske mreže koji se zasniva na dodeljivanju nasumičnih malih vrednosti težinama na ulazu skrivenog sloja, nakon čega se težine na izlazu skrivenog sloja analitički izračunavaju metodom najmanjih kvadrata. Na ovaj način se izbegava iterativna metoda gradijentnog spusta, koja pokazuje veliku računski složenost u slučaju velikih neuronskih

mreža. Ovako je prevaziđen nedostatak sporog treniranja klasičnih neuronskih mreža, koji predstavlja veliku ograničavajući faktor u široj primeni neuronskih mreža. ELM metoda je u prethodnom periodu veoma uspešno primenjivana na mnoštvu klasifikacionih problema, kao što su: klasifikacija proteina [73], klasifikacija cifara [74], prepoznavanje saobraćajnih znakova [75], klasifikacija slika tekstura [76], klasifikacija slika daljinskog nadgledanja [11], [77], [78], prepoznavanje 3D modela [79], prepoznavanje spomenika [80], [81], itd.

Konfiguracija ELM-a podrazumeva model neuronske mreže sa jednim ulaznim slojem, jednim skrivenim slojem i jednim izlaznim slojem. ELM dodatno podrazumeva da izlazni sloj ima linearnu aktivacionu funkciju i da nema vektora pomeraja. U radovima [82], [83] je dokazano da ELM predstavlja tzv. univerzalni aproksimator, tj. da je u stanju da reši proizvoljan regresioni problem uz zadatu tačnost, ukoliko ima dovoljan broj skrivenih neurona i dovoljan broj trening podataka za učenje parametara mreže. Podsećamo da se neuronske mreže, uključujući i ELM, mogu jednostavno prilagoditi za rešavanje klasifikacionih problema, tako što se indeks izlaznog neurona sa maksimalnom vrednošću može smatrati oznakom klase. U kontekstu ovoga, u [72] je pokazano da, ukoliko naučeni parametri ELM mreže mogu da aproksimiraju zadatu kontinualnu funkciju hipoteze, onda ELM može da klasifikuje podatke proizvoljno raspoređene u nepreklapajućim oblastima prostora.

U nastavku ćemo predstaviti matematički model koji ELM koristi prilikom klasifikacije podataka. Zbog razumljivosti ćemo umesto brojnog indeksa koji označava sloj mreže, koristiti oznake: *in*, *hid*, *out*. Otuda ćemo težine na ulazu skrivenog sloja označiti sa $W_{in} = W^{(2)}$, a na izlazu skrivenog sloja sa $W_{out} = W^{(3)}$. Vektore izlaze skrivenog sloja ćemo označiti sa $\mathbf{z}_{hid} = \mathbf{z}^{(2)}$, a vektor izlaza cele mreže sa $\mathbf{z}_{out} = \mathbf{z}^{(2)}$. Prilikom korišćenja ELM za klasifikaciju, kao što je slučaj u ovoj disertaciji, podrazumevaćemo da je za svaki trening primer $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ izlazni vektor $\mathbf{y}^{(i)}$ predstavljen kao binarni vektor, gde postoji samo jedna vrednost različita od nule koja određuje indeks klase kojoj $\mathbf{x}^{(i)}$ pripada. Npr. ukoliko vršimo klasifikaciju u 5 klasa, izlazni vektor $\mathbf{y}^{(i)} = [0, 1, 0, 0, 0]$ bi označio pripadnost klasi 2. Slika 21 ilustruje notaciju koju ćemo koristiti u nastavku.



Slika 21 Označavanje težinskih matrica W_{in} , W_{out} i izlaznih vektora \mathbf{z}_{in} , \mathbf{z}_{out} u ELM neuronskoj mreži.

Prema formuli 4.31, vektor izlaza $\mathbf{z}^{(k)}$, iz sloja k možemo izračunati na osnovu vektora izlaza prethodnog sloja $\mathbf{z}^{(k-1)}$, i matrice težina $W^{(k)}$ ulaznih veza sloja k . Onda se izlazi skrivenog sloja \mathbf{z}_{hid} , i izlaz cele mreže \mathbf{z}_{out} , mogu predstaviti kao

$$\begin{aligned} \mathbf{z}_{hid} &= g(W_{in}\mathbf{x} + \mathbf{b}_{in}) \\ \mathbf{z}_{out} &= g(W_{out}\mathbf{z}_{hid} + \mathbf{b}_{out}) \end{aligned} \tag{4.40}$$

Uz prethodno uvedene pretpostavke za ELM, da izlazni sloj ne sadrži pomeraj $\mathbf{b}_{out} = 0$ i da koristi linearnu aktivacionu funkciju $g(\mathbf{z}) = \mathbf{z}$, izlaz cele ELM mreže $\mathbf{z}^{(3)}$ postaje

$$\mathbf{z}_{out} = W_{out}\mathbf{z}_{hid} = W_{out} \cdot g(W_{in}\mathbf{x} + \mathbf{b}_{in}) \tag{4.41}$$

Na osnovu prethodne formule, može se obaviti treniranje celokupne ELM mreže, tj. odrediti svi težinski parametri mreže. Prema ELM teoriji, ulazne težine skrivenog sloja W_{in} i pomeraji \mathbf{b}_{in} mogu biti unapred postavljeni na slučajno generisane male vrednosti, nezavisno od ulaznih podataka. Na ovaj način se na osnovu ulaznih trening podataka $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, i poznatih parametara W_{in} i \mathbf{b}_{in} može direktno izračunati $\mathbf{z}_{hid} = g(W_{in}\mathbf{x}^{(i)} + \mathbf{b}_{in})$. Zatim se na osnovu izračunatog \mathbf{z}_{hid} , izlaz poslednjeg sloja mreže može predstaviti kao:

$$\mathbf{z}_{out} = W_{out}\mathbf{z}_{hid} = \mathbf{y}^{(i)} \tag{4.42}$$

Prethodnu formulu možemo proširiti na sve ulazne trening parove $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ za $i = 1, \dots, M$, na sledeći način. Prvo ćemo na osnovu svih ulaznih trening vektora formirati matricu izlaza skrivenog sloja:

$$Z = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \mathbf{z}_{hid}^{(1)} & \mathbf{z}_{hid}^{(2)} & \dots & \mathbf{z}_{hid}^{(M)} \\ \vdots & \vdots & \dots & \vdots \end{bmatrix}_{L_{hid} \times M} = \begin{bmatrix} g(W_{in}\mathbf{x}^{(1)} + \mathbf{b}_{in}) & g(W_{in}\mathbf{x}^{(2)} + \mathbf{b}_{in}) & \dots & g(W_{in}\mathbf{x}^{(M)} + \mathbf{b}_{in}) \\ \vdots & \vdots & \dots & \vdots \end{bmatrix}_{L_{hid} \times M} \quad 4.43$$

Na osnovu svih izlaznih vektora iz trening skupa možemo formirati matricu

$$Y = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \mathbf{y}^{(1)} & \mathbf{y}^{(2)} & \dots & \mathbf{y}^{(M)} \\ \vdots & \vdots & \dots & \vdots \end{bmatrix}_{L_{out} \times M} \quad 4.44$$

Dakle, matrica izlaza skrivenog sloja Z se može unapred izračunati za sve trening primere $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, pa ostaje jedino treniranje (tj. izračunavanje) matrice izlaznih težina W_{out} , na osnovu formule za izlaz celokupne mreže. Izlaz celokupne ELM mreže se na osnovu (4.42) može kompaktno zapisati u matricnoj formi

$$W_{out}Z = Y \quad 4.45$$

U praktičnim aplikacijama, broj trening primera M je gotovo uvek znatno veći od broja skrivenih neurona L_{hid} , pa je prethodni sistem jednačina predeterminisan. Zato se koristi metoda Moore-Penrose generalizovanog inverza (Z^\dagger) tzv. pseudo-inverz matrice [84], za izračunavanje matrice težina W_{out} na osnovu poznatog Y i prethodno izračunatog Z . Matematički se postupak izvodi na sledeći način:

$$\begin{aligned} W_{out}Z &= Y \\ W_{out}ZZ^T &= YZ^T \\ W_{out}(ZZ^T)(ZZ^T)^{-1} &= YZ^T(ZZ^T)^{-1} \\ W_{out} &= Y \underbrace{Z^T(ZZ^T)^{-1}}_{Z^\dagger} \\ W_{out} &= YZ^\dagger. \end{aligned} \quad 4.46$$

Dakle, izračunavanje pseudo-inverza se svodi na $Z^\dagger = Z^T(ZZ^T)^{-1}$. Postupak izračunavanja pseudo-inverza može pokazati numeričku nestabilnost, ukoliko je matrica ZZ^T približno sinuglarna. Zato se u praktičnim impementacijama (npr. u Matlabu), uvodi mali

regularizacioni član C prilikom izračunavanja $Z^\dagger = Z^T (ZZ^T + CI)^{-1}$ [72]. Na ovaj način se izbegava eventualna pojava singularne matrice ZZ^T .

Na osnovu prethodnog razmatranja, algoritam ELM klasifikacije se može predstaviti na sledeći način:

Algoritam ELM klasifikacije

1. Trening faza algoritma

Ulaz: Trening skup podataka $T = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}; i = 1, \dots, M$.

- 1.1. Dodeliti nasumične male vrednosti težinama W_{in} i pomerajima \mathbf{b}_{in} na ulazu skrivenog sloja.
- 1.2. Izračunati matricu izlaza skrivenog sloja Z , na osnovu $\mathbf{x}^{(i)}, W_{in}, \mathbf{b}_{in}$, prema (4.43).
- 1.3. Izračunati težine na izlazu skrivenog sloja $W_{out} = YZ^\dagger$, na osnovu pseudo-inverza $Z^\dagger = Z^T (ZZ^T)^{-1}$.

2. Test faza algoritma

Ulaz: Test skup podataka $S = \{\mathbf{x}^{(j)}\}; j = M + 1, \dots, N$.

- 2.1. Izračunati matricu izlaza skrivenog sloja Z za nove instance iz test skupa S , korišćenjem težina W_{in} i pomeraja \mathbf{b}_{in} iz trening faze.
 - 2.2. Izračunati izlaz mreže $Y_{test} = W_{out}Z$, korišćenjem težina W_{out} izračunatih u 3. koraku trening faze.
 - 2.3. Za svaki test primer (tj. svaku vrstu matrice Y_{test}), odrediti oznaku klase kao indeks maksimalnog elementa u toj vrsti.
-

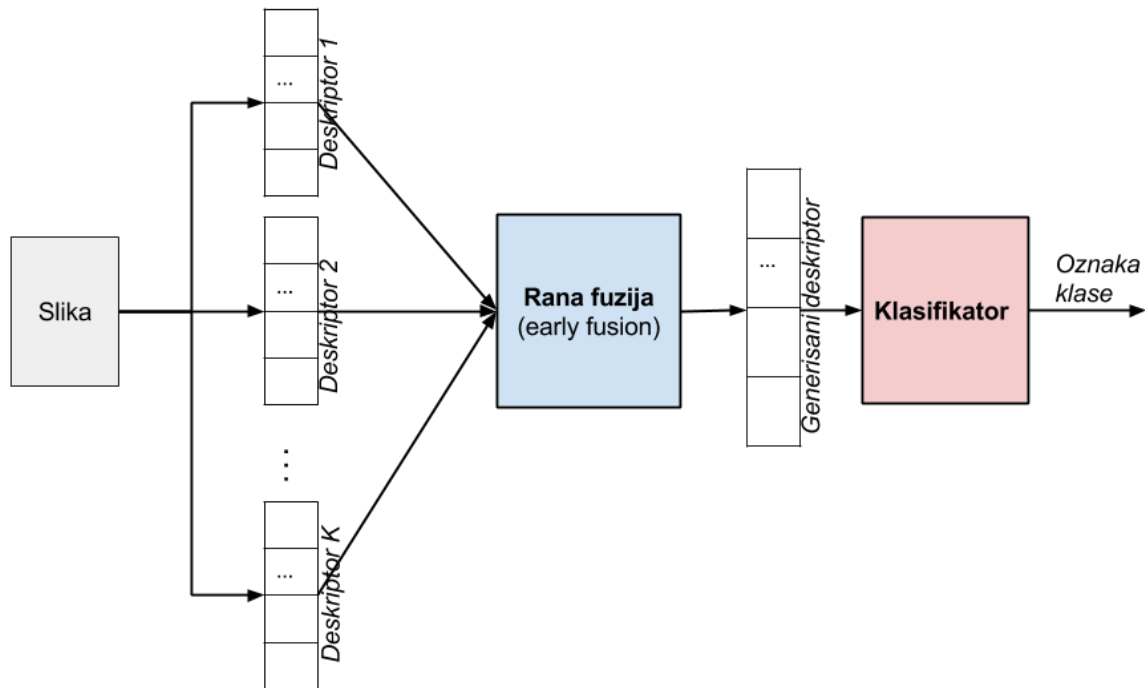
5 RANA FUZIJA DESKRIPTORA

Novija istraživanja u oblasti automatske klasifikacije slika, pokazala su da korišćenje samo jedne vrste deskriptora slike nije dovoljno za postizanje visoke tačnosti i robusnosti rezultata [5], [40], [48], [63], [64], [85], [86]. Ovo je dovelo do zaključka da je neophodno razviti metode za kombinovanje većeg broja deskriptora. Prilikom korišćenja više različitih deskriptora za klasifikaciju, ključni problem predstavlja izbor metode za njihovo kombinovanje (fuziju). Generalno, postoje dva pristupa za fuziju, u zavisnosti od toga da li se fuzija vrši na nivou deskriptora ili na nivou rezultata klasifikatora [48], [86–90]. Prvi pristup, pod nazivom “rana fuzija”, vrši kombinovanje deskriptora slike u novi jedinstveni deskriptor, pre primene automatskih klasifikatora (tj. tehnika mašinskog učenja). S obzirom da se fuzija vrši direktno nad deskriptorima, ova tehnika se često naziva “fuzija deskriptora” (engl. descriptor fusion). Drugi pristup - “kasna fuzija”, prvo primenjuje zasebne klasifikatore nad nezavisno izračunatim deskriptorima, a zatim integriše rezultate pojedinačnih klasifikatora u konačni jedinstveni rezultat klasifikacije. Ova tehnika se sreće i pod nazivom “fuzija klasifikatora” (engl. classifier fusion).

U nastavku će prvo biti detaljno opisan opšti princip rane fuzije, uz detaljan opis tehnika normalizacije izračunatih deskriptora. Zatim će biti predstavljen algoritam za ranu fuziju deskriptora boje i teksture, koji je predložen od strane autora. Na kraju će biti predstavljeni rezultati eksperimentalne evaluacije opisanih metoda, uključujući i uporednu analizu sa aktuelnim rezultatima iz literature.

5.1 Postupak rane fuzije deskriptora

Rana fuzija tj. fuzija na nivou deskriptora, predstavlja tehniku kombinovanja deskriptora koja integriše nezavisne deskriptore pre primene postupka mašinskog učenja (klasifikatora) za klasifikaciju [87]. Postupak rane fuzije ilustruje Slika 22.



Slika 22 Postupak rane fuzije deskriptora

Uočava se da se u postupku rane fuzije formira novi deskriptor tzv. „generisani deskriptor“ (meta-deskriptor), nad kojim se zatim primenjuju tehnike mašinskog učenja (klasifikatori). Generisanje novog deskriptora se može obaviti jednostavnim nadovezivanjem više vektora deskriptora, pri čemu se pojedinim deskriptorima (svim ili pojedinim elementima vektora) može dodeliti određeni težinski faktor [5], [11], [87]. Ovakva tehnika se karakteriše jednostavnošću implementacije i malom računskom kompleksnošću. Nedostatak ove tehnike se ogleda u povećanoj dimenzionalnosti generisanih meta-deskriptora nakon rane fuzije, kao i relativno manjom tačnošću rezultata ovih tehnika u odnosu na kasnu fuziju.

5.2 Tehnike normalizacije podataka

U opštem slučaju, različiti ulazni vektori podataka (deskriptori) prilikom kombinovanja mogu imati različit opseg vrednosti kao i različitu raspodelu verovatnoća. Kao posledica toga, vektori čije vrednosti imaju veću varijansu, biće dominantni prilikom računanja rastojanja među vektorima. Ovo može dovesti do neopravdanog smanjenja značaja vektora sa manjom varijansom vrednosti elemenata. Ovaj problem je podjednako zastupljen prilikom rane fuzije deskriptora kao i u postupku kasne fuzije rezultata klasifikacije, o čemu će kasnije biti više reči. Problem je posebno uočljiv prilikom rane fuzije kada se kombinuju različite vrste deskriptora (boje, teksture, oblika) ili različite vrste signala (npr. audio, vizuelni i tekstualni podaci).

Da bi se prilikom kombinovanja ovakvih vektora deskriptora eliminisao uticaj vektora sa većom varijansom, neophodno je izvršiti adekvatnu normalizaciju podataka, tj. preslikavanje na standardni opseg vrednosti (npr. [0, 1]). Normalizacija se obavlja nad originalnim vektorima deskriptora, pre bilo kakve fuzije. U literaturi se mogu sresti različite metode koje se koriste za normalizaciju heterogenih vektora [91]. Ove metode se na identičan način mogu primeniti kako za normalizaciju različitih deskriptora prilikom rane fuzije, tako i za normalizaciju rezultata klasifikatora (verovatnoća) prilikom kasne fuzije. Najčešće korišćene tehnike normalizacije obuhvataju: “min-max”, “z-score” i *tanh* [91]. Svaka od ovih metoda ima određene prednosti i nedostatke. “Min-max” metoda je veoma osetljiva na pojavu vrednosti koje značajno odstupaju od proseka (engl. outliers). Metoda “z-score” pokazuje znatno veću stabilnost i daje optimalne rezultate u situacijama kada je moguće izračunati srednju vrednost i varijansu svakog od elemenata u vektoru, na osnovu većeg skupa podataka. Metoda zasnovana na funkciji *tanh* je veoma robusna i efikasna, ali zahteva računski zahtevno predviđanje parametara kroz iterativne postupke primenjene nad trening skupom podataka. U nastavku je dat opis postupaka normalizacije, prethodno pomenutih metoda.

Ako sa $\mathbf{x} = [x_1, x_2, \dots, x_i, \dots, x_n]$ označimo vektor ulaznog deskriptora čiji je i -ti element označen sa x_i , a sa \mathbf{x}^* vektor deskriptora nakon normalizacije, onda se linearna “min-max” normalizacija matematički može predstaviti kao:

$$x_i^* = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}; \quad i = 1, \dots, n \quad 5.1$$

gde $\min(x_i)$ i $\max(x_i)$ predstavljaju minimalnu i maksimalnu vrednost elementa na i -toj poziciji u svim vektorima iz trening skupa. Vrednosti $\min(x_i)$ i $\max(x_i)$ se mogu postaviti na fiksnu vrednost, ukoliko je unapred poznat opseg vrednosti elemenata vektora deskriptora. U suprotnom, ove vrednosti je moguće izračunati na osnovu trening skupa podataka. Elementi normalizovanog vektora će imati vrednosti u opsegu [0, 1].

Najčešće korišćena tehnika normalizacije je tzv. “z-score” metoda koja vrši normalizaciju tako da elementi vektora na svakoj poziciji imaju srednju vrednost nula i standardnu devijaciju jednaku jedinici. Matematički se ovaj metod može predstaviti kao:

$$x_i^* = \frac{x_i - \mu(x_i)}{\sigma(x_i)}; \quad i = 1, \dots, n \quad 5.2$$

gde $\mu(x_i)$ i $\sigma(x_i)$ predstavljaju srednju vrednost i standardnu devijaciju elementa na i -toj poziciji u svim vektorima iz trening skupa, dok x_i^* predstavlja i -ti element vektora nakon normalizacije. Srednju vrednost i standardnu devijaciju je moguće izračunati na osnovu celokupnog trening skupa vektora.

U slučaju normalizacije metodom zasnovanom na \tanh funkciji, elementi vektora deskriptora nakon normalizacije x_i^* , se izračunavaju primenom formule

$$x_i^* = 0.5 \left\{ \tanh \left(0.01 \left(\frac{x_i - \mu(x_i)}{\sigma(x_i)} \right) \right) + 1 \right\}; i = 1, \dots, n. \quad 5.3$$

Eksperimentalni rezultati nekoliko studija [5], [91], pokazali su da linearna “min-max“ tehnika pokazuje znatno slabije rezultate u odnosu na “z-score” i \tanh normalizaciju, pri čemu je metoda \tanh značajno kompleksnija u računskom pogledu. Uzimajući u obzir kompromis između efikasnosti izračunavanja i tačnosti rezultata, može se zaključiti da “z-score” predstavlja optimalan izbor, zbog čega je ova tehnika preporučena za normalizaciju podataka u postupcima fuzije.

Treba odmah naglasiti da se potreba za normalizacijom heterogenih podataka javlja i prilikom kasne fuzije rezultata klasifikacije, kao što će biti opisano u narednom poglavlju. Normalizaciju je prilikom kasne fuzije neophodno primeniti kada se kombinuju rezultati potpuno različitih (heterogenih) klasifikatora, npr. SVM, Neuronske mreže, itd. U slučaju kombinovanja više istih vrsta klasifikatora, uz korišćenje različitih parametara (npr. ansambli ELM), nije neophodno normalizovati rezultate.

5.2.1 Evaluacija tehnika normalizacije deskriptora

Uticaj različitih tehnika normalizacije deskriptora na tačnost klasifikacije slika prilikom rane fuzije, analizirali smo u radovima [5], [6]. U ovim radovima je razmatran problem binarne klasifikacije slika, tj. klasifikacije u dve klase. Klase su definisane tako da jedna klasa sadrži slike scena snimljenih u unutrašnjosti različitih objekata (engl. indoor), dok druga klasa sadrži slike snimljene izvan objekata (engl. outdoor). Kao bazični deskriptori slike, korišćeni su MPEG-7 deskriptori. Konkretno je korišćen CLD deskriptor boje i EHD deskriptor teksture, opisani u Poglavlju 2. Dodatno je analiziran deskriptor dobijen kombinovanjem ovih deskriptora kao jednostavno nadovezivanje ova dva deskriptora (CLD+EHD). Deskriptor dobijen ovakvom tehnikom fuzije predstavlja jednostavan primer rane fuzije deskriptora. Kao klasifikator je korišćen nelinearni SVM metod [1]. Analizirane su dve najčešće korišćene tehnike normalizacije podataka – „min-max“ i „z-score“. Tabela 2 prikazuje rezultate klasifikacije u zavisnosti od primenjene tehnike normalizacije deskriptora, nad test kolekcijama IITM-SCID2 [35] i COREL-INOUT [34].

Tabela 2 Tačnost klasifikacije pod uticajem različitih tehnika normalizacije deskriptora [5].

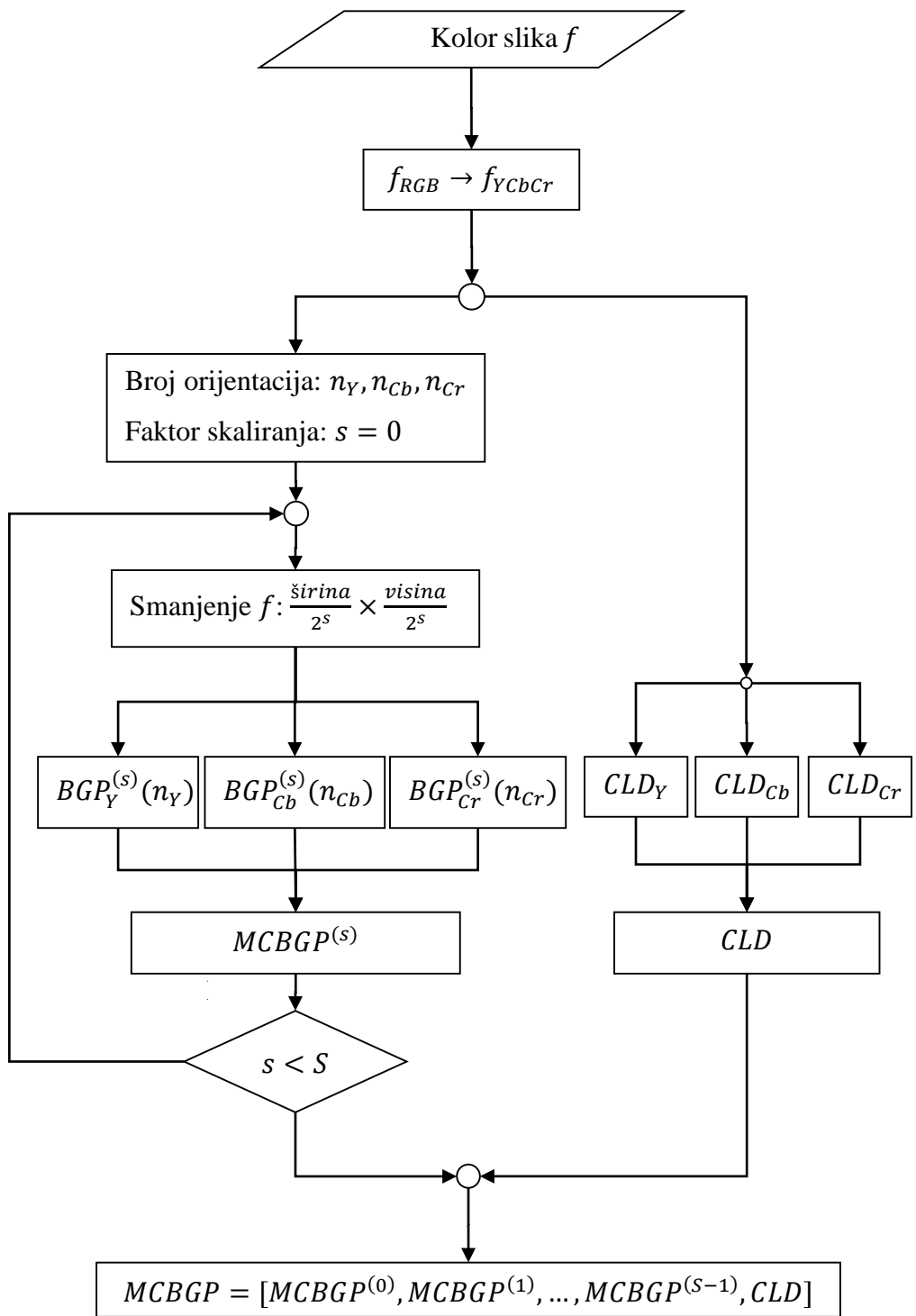
Deskriptor	Dimenzije	IITM-SCID2 kolekcija		COREL-INOUT kolekcija	
		“min-max”	“z-score”	“min-max”	“z-score”
CLD	22	78.98%	82.71%	80.83%	82.59%
EHD	80	83.10%	87.03%	83.70%	83.87%
CLD+EHD	102	88.79%	91.55%	89.97%	91.53%

Na osnovu prethodnih rezultata, eksperimentalno je potvrđeno da “z-score” tehnika normalizacije konzistentno postiže bolje rezultate u odnosu na „min-max“ normalizaciju. Ovde se odmah može uočiti značaj fuzije različitih deskriptora u postupku klasifikacije slike. Čak i najjednostavniji oblik rane fuzije deskriptora boje i teksture donosi značajno poboljšanje rezultata klasifikacije. Tako se u prethodnim eksperimentima, najveća tačnost klasifikacije od preko 91.5%, dobija primenom deskriptora CLD+EHD koji je formiran tehnikom rane fuzije, uz primenu “z-score” normalizacije.

5.3 Predloženi postupak rane fuzije deskriptora

U radu [11] je predložen novi deskriptor slike kroz postupak rane fuzije robusnog BGP deskriptora [24] i kompaktnog CLD deskriptora [54]. Da bi formirali novi višekanalni deskriptor slike koji integriše informacije iz više kanala boje, proširili smo BGP deskriptor [24] za izračunavanje nad kolor slikama i integrisali ga sa kompaktnim CLD deskriptorom [54]. Novi deskriptor je nazvan Višekanalni Binarni Gaborovi Uzorci (engl. Multichannel Binary Gabor Patterns – MCBGP). Detaljni opis postupka izračunavanja MCBGP deskriptora je dat u nastavku, i prikazuje ga Slika 23.

Postupak započinje konverzijom slike u $YCbCr$ prostor boja. S obzirom da standardni RGB prostor boja pokazuje veliku korelaciju među komponentama boje, konvencionalni kolor prostori kao što je $YCbCr$ su pokazali veću efikasnost prilikom klasifikacije slika. Ovaj kolor prostor ima važnu osobinu da samo Y komponenta sadrži informaciju o sjajnosti kao najznačajnijem vizuelnom sadržaju. Hrominentne komponente Cb i Cr nose informaciju o boji i mogu biti odmeravane sa manjom frekvencijom u cilju redukovanja dužine (dimenzionalnosti) deskriptora.



Slika 23 Predloženi postupak izračunavanja višekanalnog BGP deskriptora – MCBGP, kroz fuziju deskriptora teksture i boje [11].

Neka sa $BGP_c^{(s)}(n_c)$ označimo BGP deskriptor izračunat nad kolor kanalom $c \in \{Y, Cb, Cr\}$, pri rezoluciji slike $s = \{0, 1, \dots, S - 1\}$, korišćenjem Gaborove banke filtara sa n_c orijentacija. Konkretno, izračunavan je jedan $BGP_Y^{(s)}(n_Y)$ histogram na Y kanalu korišćenjem n_Y orijentacija, dok je za izračunavanje $BGP_{Cb}^{(s)}(n_{Cb})$ i $BGP_{Cr}^{(s)}(n_{Cr})$ histograma nad hrominentnim komponentama korišćen redukovani broj orijentacija $n_{Cb} = n_{Cr} < n_Y$. Višekanalni BGP deskriptor na rezoluciji s se zatim formira kao

$$MCBGP^{(s)} = [BGP_Y^{(s)}(n_Y), BGP_{Cb}^{(s)}(n_{Cb}), BGP_{Cr}^{(s)}(n_{Cr})] \quad 5.4$$

Da bi se objedinile korisne informacije o slici pri različitim rezolucijama, pristup podrazumeva korišćenje Gaborove banke filtara uz variranje parametara σ i λ iz jednačine (3.4). Ovi parametri utiču na prostornu širinu filtra, što implicitno stvara efekat kao da se ulaznoj slici promenila rezolucija, a filter ostao iste širine. Nakon izračunavanja deskriptora primenom više filtarskih banki sa različitim parametrima, svi dobijeni deskriptori se nadovezuju u jedan konačni deskriptor slike. Tako su autori publikacije [24] koristili tri para vrednosti parametara σ i λ da bi obuhvatili detalje na tri različite rezolucije. Ovakva vrsta višerezolucione analize uobičajeno poboljšava rezultate automatske klasifikacije slika u praktičnim aplikacijama. Međutim, ovakav pristup ima nedostatak u pogledu računске kompleksnosti usled povećanja prostornih dimenzija filtra. Zato je u radu [11] primenjen drugačiji pristup koji podrazumeva da se rezolucija slike umanjuje nekoliko puta, uz fiksne parametre Gaborovog filtra. Nad slikom sa različitim rezolucijama se primenjuje ista banka Gaborovih filtara relativno male prostorne širine. Izdvojeni deskriptori pri različitim rezolucijama se nadovezuju u jedan konačni deskriptor slike. Korišćenje iste banke Gaborovih filtara relativno male prostorne širine značajno povećava efikasnost izračunavanja deskriptora. Eksperimentalno će biti pokazano da je korišćenje iste slike na više različitim rezolucija od velikog uticaja na tačnost klasifikacije; posebno u slučaju banke Gaborovih filtara sa malim brojem orijentacija.

U konkretnom slučaju, originalna slika rezolucije $\text{širina} \times \text{visina}$ je umanjena više puta na rezoluciju $\frac{\text{širina}}{2^s} \times \frac{\text{visina}}{2^s}$, gde je $s = \{0, 1, 2\}$. Konačni višekanalni BGP descriptor (MCBGP) je formiran nadovezivanjem prethodno izračunatih deskriptora pri različitim rezolucijama $MCBGP^{(s)}$, kao i CLD deskriptora slike:

$$MCBGP = [MCBGP^{(0)}, MCBGP^{(1)}, \dots, MCBGP^{(S-1)}, CLD] \quad 5.5$$

Izbor optimalnih parametara n_c i S zavisi od rezolucije i vizuelnog kvaliteta test slika. Dobar kompromis između dimenzionalnosti deskriptora i tačnosti klasifikacije se može postići

sledećim parametrima: $n_Y = 6$ orijentacija za Y komponentu (28-dimenzionalni BGP_Y), $n_{Cb} = n_{Cr} = 4$ orijentacija za Cb i Cr komponente boje (12-dimenzionalni BGP_{Cb} i BGP_{Cr}) i $S = 3$ rezolucije slike. Ovo rezultira formiranje robusnog 178-dimenzionalnog vektora $MCBGP$ deskriptora slike ($3 \times (28 + 12 + 12) + 22 = 178$). Uticaj izbora različitih vrednosti parametara na tačnost klasifikacije, biće detaljno eksperimentalno evaluiran u narednim poglavljima.

S obzirom da je prilikom izračunavanja $MCBGP$ obavljena fuzija heterogenih deskriptora BGP i CLD , čiji je opseg vrednosti različit, neopodno je obaviti normalizaciju dobijenih vrednosti $MCBGP$ elemenata vektora. Ovde je korišćena “z-score” metoda koja obavlja normalizaciju na nultu srednju vrednost i jediničnu varijansu [91].

5.3.1 Eksperimentalna evaluacija

U nastavku su predstavljeni rezultati evaluacije prethodno opisanog algoritma rane fuzije deskriptora koji je predložen u radu [11]. Prilikom narednih testiranja korišćena je javno dostupna kolekcija slika u boji “Landuse21”, koja sadrži 2100 slika snimljenih iz vazduha i klasifikovanih u 21 kategoriju.

Prva grupa eksperimenata je obavljena sa ciljem da pokaže uticaj rane fuzije boje i tekstore na tačnost klasifikacije. U eksperimentima su analizirani različiti parametri deskriptora boje, kao i različite rezolucije slike prilikom više-rezolucijskog pristupa za izračunavanje $MCBGP$ deskriptora slike [11]. Prilikom izračunavanja odziva Gaborovog filtra korišćen je standardni skup parametara $\lambda = 1$, $\sigma = 0.56\lambda$, $\gamma = 0.5$ (prethodno opisanih u Glavi 3.1.3). Izračunavanje $MCBGP$ deskriptora slike se dodatno kontroliše sledećim parametrima: broj različitih rezolucija slike pri kojima se izdvaja deskriptor, broj orijentacija banke Gaborovih filtara za Y , Cb i Cr komponentu boje ($n_Y, n_{Cb}, i n_{Cr}$), i integracija CLD deskriptora (da/ne). Treba uočiti da vrednosti parametara $n_{Cb} = n_{Cr} = 0$ i $CLD = ne$, odgovaraju monohromatskom („sivom“) $MCBGP$ deskriptoru, sličnom onom koji je predstavljen u radu [24]. Tabela 3 prikazuje rezultate tačnosti klasifikacije u zavisnosti od vrednosti parametara $MCBGP$ deskriptora. U ovim eksperimentima je kao klasifikator korišćen ELM-E ansambl neuronskih mreža. ELM-E sadrži $k=10$ zasebnih ELM mreža sa po $L=1000$ neurona u skrivenom sloju, pri čemu se agregacija rezultata vrši načinom opisanim u [11]. Kao parametar individualnih ELM mreža korišćena je fiksna vrednost $C = 0.1$.

Tabela 3 Uticaj parametara koji definišu deskriptor slike MCBGP, na tačnost rezultata klasifikacije. Oznake pojedinih kolona u tabeli: rezol. – broj različitih rezolucija slike na kojima su izračunavani deskriptori; $n_Y, n_{Cb}, i n_{Cr}$ – broj orijentacija banke Gaborovih filtara za Y, Cb i Cr komponentu boje respektivno; CLD – označava da li je bilo integracije CLD deskriptora u konačni MCBGP deskriptor slike, dimenz. – označava konačnu dimenziju vektora deskriptora slike.

Deskriptor	rezol.	n_Y	n_{Cb}	n_{Cr}	CLD	dimenz.	Tačnost (\pm st.dev.)
MCBGP-gray	1	6	0	0	da	28	69.74 (\pm 1.79)
MCBGP-gray	2	6	0	0	ne	56	78.17 (\pm 1.68)
MCBGP-gray	3	6	0	0	ne	84	78.97 (\pm 1.91)
MCBGP	1	6	4	4	ne	52	78.45 (\pm 1.73)
MCBGP	2	6	4	4	ne	104	82.14 (\pm 1.67)
MCBGP	3	6	4	4	ne	156	82.86 (\pm 1.59)
MCBGP	1	6	4	4	da	74	78.50 (\pm 1.57)
MCBGP	2	6	4	4	da	126	82.33 (\pm 1.60)
MCBGP	3	6	4	4	da	178	84.31 (\pm 1.63)
MCBGP-gray	1	8	0	0	ne	72	77.29 (\pm 1.92)
MCBGP-gray	2	8	0	0	ne	144	79.40 (\pm 1.99)
MCBGP-gray	3	8	0	0	ne	216	79.60 (\pm 1.64)
MCBGP	1	8	6	6	ne	128	81.11 (\pm 1.72)
MCBGP	2	8	6	6	ne	256	82.59 (\pm 1.98)
MCBGP	3	8	6	6	ne	384	81.71 (\pm 1.70)
MCBGP	1	8	6	6	da	150	82.43 (\pm 1.67)
MCBGP	2	8	6	6	da	278	84.29 (\pm 1.53)
MCBGP	3	8	6	6	da	406	83.77 (\pm 1.70)

Rezultati koje prikazuje Tabela 3 impliciraju da rana fuzija informacija o boji i teksturi, u okviru predloženog MCBGP deskriptora, konzistentno unapređuje rezultate klasifikacije. Kao propratni efekat fuzije javlja se određeno povećanje dimenzionalnosti deskriptora, koje zanemarljivo utiče na brzinu klasifikatora. Takođe se može zaključiti da više-rezolucijski pristup prilikom izračunavanja deskriptora ima veliki uticaj na uspešnost banke Gaborovih filtara, posebno u slučaju “grubljih” diskretizacija orijentacije. Pri tome je preporučljivo koristiti minimalno $S = 2$ različite rezolucije, za test slike dimenzija 256×256 piksela.

Kao kompromis između tačnosti klasifikacije i dimenzionalnosti deskriptora, preporuka je koristiti sledeće parametre MCBGP deskriptora: $n_Y = 6, n_{Cb} = n_{Cr} = 4, CLD = da$, i broj rezolucija jednak 3. Ukoliko nije drugačije naglašeno, u narednim eksperimentima će svuda biti korišćeni ovi parametri prilikom izračunavanja MCBGP deskriptora.

6 KASNA FUZIJA KLASIFIKATORA

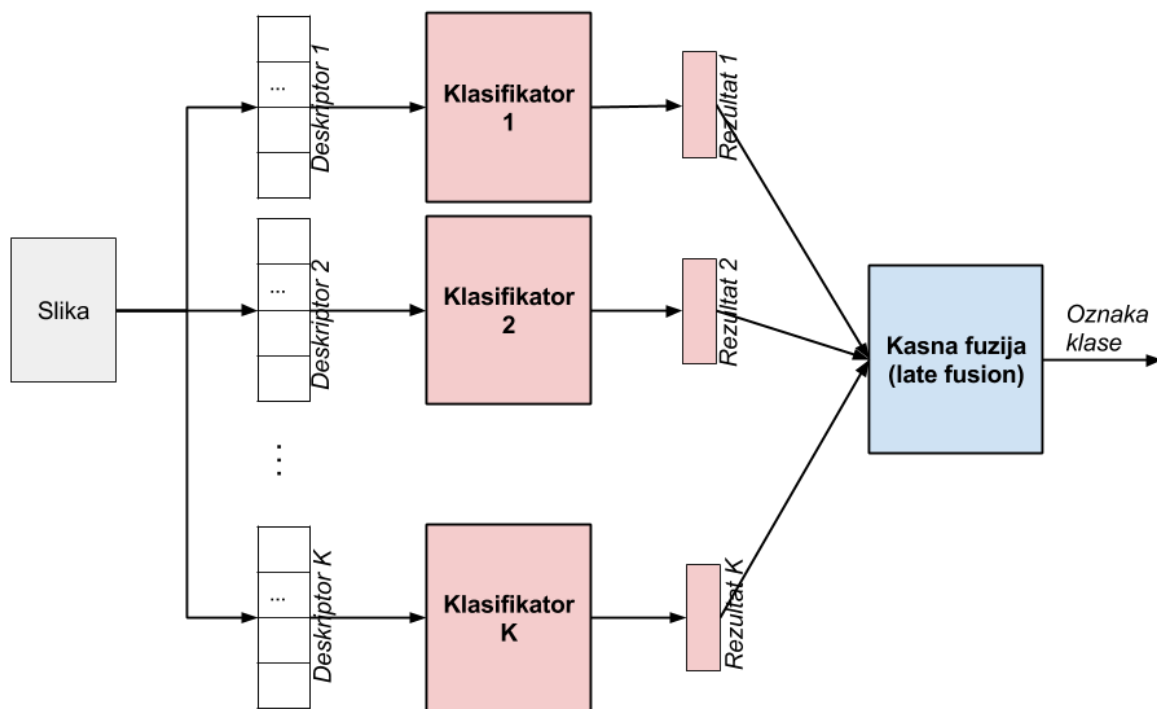
Napredniji pristup za fuziju prvo izračunava rezultat nezavisnih klasifikatora nad zasebno izdvojenim deskriptorima, a zatim integriše rezultate pojedinačnih klasifikatora u konačni jedinstveni rezultat kalsifikacije. Ovakva tehnika se naziva “kasna fuzija”. S obzirom da se fuzija vrši nad rezultatima klasifikatora, ova tehnika se često naziva “fuzija klasifikatora” (engl. classifier fusion).

U nastavku će prvo biti opisan opšti postupak kasne fuzije i predstavljene tehnike za agregaciju rezultata svih klasifikatora, sa akcentom na fuziji ELM klasifikatora. Zatim će biti predstavljeni algoritmi za kasnu fuziju klasifikatora koji se predloženi od strane autora. Ovde će biti opisan i najznačajniji doprinos ove teze, kroz predlog hijerarhijske metode za kasnu fuziju ELM klasifikatora. Na kraju će biti predstavljeni rezultati eksperimentalne evaluacije, uključujući i uporednu analizu sa aktulenim razultatima iz literature.

6.1 Postupak kasne fuzije klasifikatora

Prema definiciji iz [87], kasna fuzija je postupak koji prvo preslikava deskriptore u rezultate zasebno treniranih klasifikatora, a zatim ove rezultate integriše u konačni izlaz. Opšti postupak kasne fuzije prikazuje Slika 24.

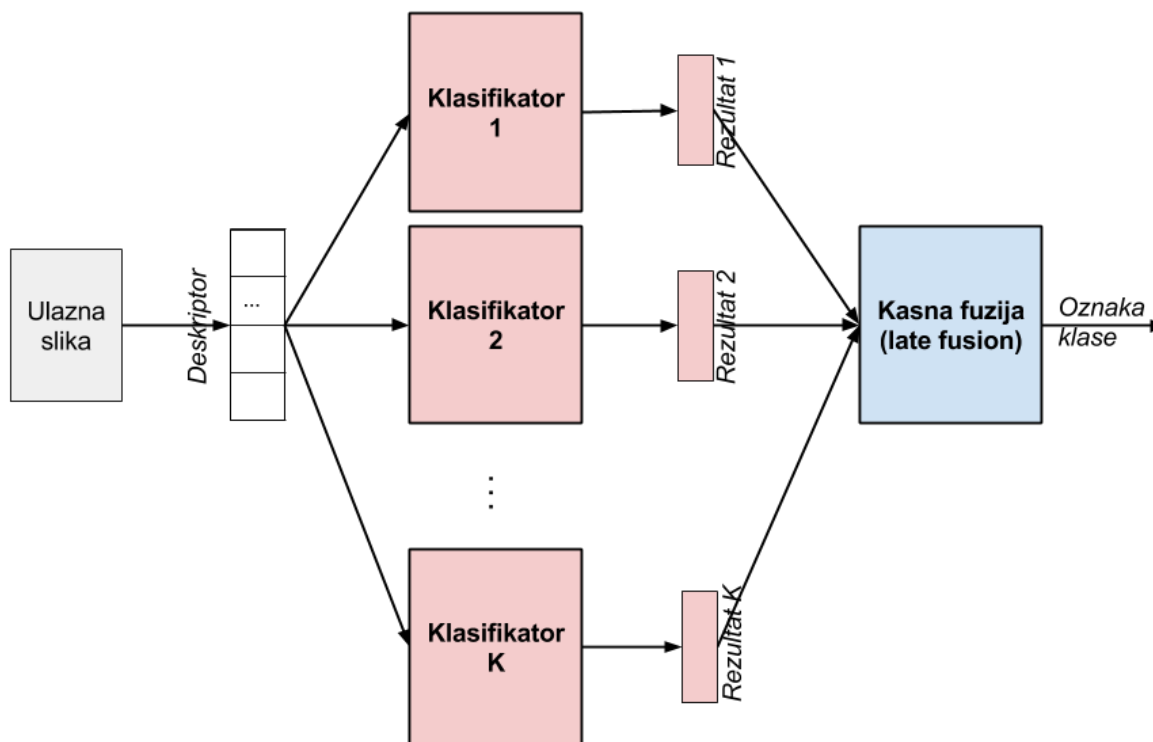
Prilikom postupka kasne fuzije se podrazumeva da koristimo veći broj nezavisnih klasifikatora (K). Pri tome, svaki klasifikator reprezentuje po jednu hipotezu, čiji se parametri treniraju odgovarajućom tehnikom mašinskog učenja. Moguće je koristiti istu tehniku mašinskog učenja za sve nezavisne klasifikatore (npr. po jednu neuronsku mrežu za svaki klasifikator), ili kombinovati različite tehnike (npr. SVM, Neuronske mreže, ELM, itd.). Ukoliko svi klasifikatori koriste istu tehniku mašinskog učenja, ovakve metode se uobičajeno nazivaju *homogeni ansambli* klasifikatora (engl. homogeneous ensembles), ili skraćeno - ansambli. Ako se koriste različite komplementarne tehnike, onda se nazivaju *heterogeni* ili *hibridni* klasifikatori. Broj individualnih klasifikatora K zavisi od primenjenih tehnika mašinskog učenja i može imati veliki uticaj na tačnost konačnog rezultata. Optimalna vrednost K se najčešće određuje primenom statističke analize nad konkretnim trening skupom podataka.



Slika 24 Ilustracija postupka kasne fuzije

U najopštijem obliku, postupak kasne fuzije koji ilustruje Slika 24, koristi više različitih vrsta deskriptora slike, pri čemu za svaku vrstu deskriptora može biti korišćen drugačiji klasifikator tj. drugačija tehnika mašinskog učenja. U slučaju kada se koriste različiti deskriptori slike, dovoljno je primeniti istu vrstu klasifikatora za svaki deskriptor tj. formirati homogeni ansambl, jer ulazni deskriptori već opisuju različite karakteristike slike. Pri tome se svaki klasifikator trenira nezavisno na osnovu jedne vrste deskriptora, korišćenjem celokupnog trening skupa podataka.

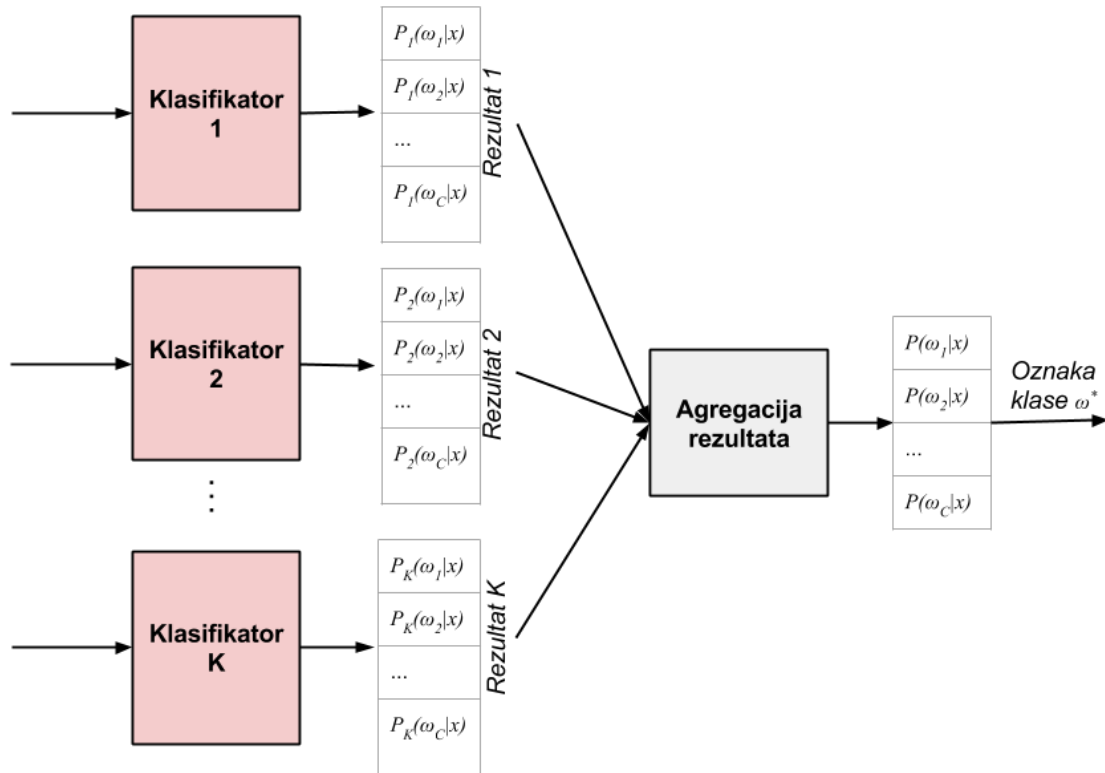
Ukoliko se koristi samo jedna vrste deskriptora slike, kasna fuzija se takođe može iskoristiti za povećanje tačnosti klasifikacije. U ovom slučaju se isti deskriptor dovodi na ulaze više različitih vrsta klasifikatora (Slika 25). Ovde je neophodno koristiti različite vrste klasifikatora, jer bi korišćenje istog klasifikatora sa istim ulaznim deskriptorom dovelo do jednakih rezultata. Cilj ovakve kasne fuzije je kombinovanje što većeg broja komplementarnih i nezavisnih klasifikatora, da bi se povećala tačnosti konačnog rezultata klasifikacije.



Slika 25 Primer kasne fuzije kada se koristi jedinstven deskriptor ulazne slike

Jedan od najvažnijih problema prilikom fuzije više klasifikatora je način na koji će se rezultati individualnih klasifikatora kombinovati u jedinstveni rezultat klasifikacije. Postupak kasne fuzije se u jednostavnom obliku može obaviti izračunavanjem srednje vrednosti izlaznih rezultata svih K nezavisnih klasifikatora [11], [88], [92]. Umesto srednje vrednosti, mogu se koristiti i druge statističke mere [93]. Ovakve tehnike podrazumevaju da svi klasifikatori imaju isti značaj prilikom računanja konačnog rezultata (tj. istu težinu), što u praksi nije uvek slučaj. Zato se kombinovanje rezultata klasifikatora može obaviti uvođenjem dodatnog klasifikatora, čiji zadatak je da u postupku treniranja “nauči” značaj rezultata svakog od nezavisnih klasifikatora, kroz izračunavanje odgovarajućih težinskih faktora [13], [40], [89], [94–96]. Ovakve tehnike kasne fuzije su pokazale povećanu tačnost rezultata, ali uz povećanje računске složenosti postupka usled korišćenja dodatnih klasifikatora.

Treba odmah istaći da je prilikom kasne fuzije rezultata različitih klasifikatora (heterogeni klasifikatori), neophodno prvo izvršiti normalizaciju izlaznih vrednosti klasifikatora primenom postupaka opisanih u Poglavlju 5.2. U postupcima kasne fuzije, normalizaciju je neophodno primeniti samo kada se kombinuju rezultati potpuno različitih (heterogenih) klasifikatora, npr. SVM, Neuronske mreže, itd. U slučajevima kada se koristi veći broj klasifikatora iste vrste (npr. ELM), nije neophodno vršiti normalizaciju.



Slika 26 Ilustracija postupka kombinovanja (agregacije) rezultata klasifikatora prilikom postupka kasne fuzije.

Da bismo matematički opisali postupak kasne fuzije, izlazne vrednosti klasifikatora ćemo predstaviti kao verovatnoće pojavljivanja ulaznih vektora deskriptora. Neka je sa $P_k(\omega_j|\mathbf{x})$ označen rezultat klasifikacije k -tog klasifikatora, t.j. posteriorna verovatnoća k -tog klasifikatora da ulazni vektor \mathbf{x} pripada klasi ω_j . Neka sa ω^* označimo jedinstvenu klasu koja će konačno biti predviđena. U svim narednim tehnikama, konačna klasa ω^* se određuje tako što se prvo izlazni vektori verovatnoća svih klasifikatora agregiraju u novi izlazni vektor (množenjem, sumiranjem, itd.). Zatim se u ovakvom vektoru određuje indeks elementa sa maksimalnom vrednošću verovatnoće ($\omega^* = \underset{j}{\operatorname{argmax}}(\dots)$). Slika 26 ilustruje opisani postupak.

U nastavku će biti predstavljene dve grupe metoda za agregaciju rezultata klasifikatora prilikom kasne fuzije. Prva grupa metoda koristi relativno jednostavne statističke mere, dok druga grupa metoda uvodi dodatne klasifikatore koji se obučavaju na osnovu izlaza osnovnih klasifikatora.

6.2 Agregacija rezultata klasifikatora statističkim merama

Teorijski okvir za agregaciju rezultata statističkim kombinovanjem više različitih klasifikatora, razvijen je u radu [75]. Razmatran je problem klasifikacije ulaznog vektora (deskriptora) u jednu od C klasa, na osnovu klasifikacionih rezultata K individualnih klasifikatora. Za izračunavanje oznake konačne klase ω^* , najčešće se koriste sledeće tehnike (pravila) kombinovanja rezultata klasifikatora:

1. **Pravilo proizvoda:** Ulaznom vektoru (deskriptoru) \mathbf{x} se dodeljuje konačna klasa ω^* prema sledećoj formuli:

$$\omega^* = \operatorname{argmax}_{1 \leq j \leq C} \prod_{k=1}^K P_k(\omega_j | \mathbf{x}) \quad 6.1$$

Ovo praktično znači da se agregiranje izlaznih vektora verovatnoća svih klasifikatora vrši množenjem odgovarajućih izlaza individualnih klasifikatora, nakon čega se određuje indeks maksimalnog elementa u novodobijenom vektoru. Na osnovu teorije verovatnoće, jasno je da je ovo pravilo zasnovano na pretpostavci o statističkoj nezavisnosti svih K klasifikatora koji učestvuju u postupku. Prilikom konkretne primene, potrebno je pažljivo razmotriti da li je ovaj uslov ispunjen, jer u pojedinim sistema ovakvu pretpostavku nije moguće direktno koristiti (npr. u multimodalnim biometrijskim sistemima [97]).

2. **Pravilo sumiranja:** Prema ovom pravilu, jedinstvena konačna klasa ω^* se dodeljuje ulaznom deskriptoru \mathbf{x} na sledeći način:

$$\omega^* = \operatorname{argmax}_{1 \leq j \leq C} \sum_{k=1}^K P_k(\omega_j | \mathbf{x}) \quad 6.2$$

Za primenu pravila sumiranja, osim pretpostavke o statističkoj nezavisnosti klasifikatora iz prethodno definisanog pravila proizvoda, neophodno je uvesti dodatnu pretpostavku da se posteriorne verovatnoće $P_k(\omega_j | \mathbf{x})$ ne razlikuju značajno od „a priori“ verovatnoća $P_k(\omega_j)$. Ovakva pretpostavka je moguća u slučaju kada postoji visok stepen sličnosti među različitim klasama, koji dovodi do nejednoznačnosti prilikom klasifikacije. Ovo je u praksi slučaj prilikom klasifikacije slika, gde slike koje pripadaju različitim klasama mogu imati veoma sličan vizuelni sadržaj.

Treba imati na umu da je prethodna analiza pravila proizvoda i sumiranja, podrazumevala da su verovatnoće $P_k(\omega_j | \mathbf{x})$ potpuno tačno predviđene (proračunate) od strane svih automatskih klasifikatora. Međutim, u praksi, svaki klasifikator vrši samo približno predviđanje verovatnoće pripadnosti klasi na osnovu deskriptora \mathbf{x} , unoseći pritom određenu grešku predviđanja.

Ukupna greška nakon kombinovanja rezultat klasifikacije svih K klasifikatora, u slučaju pravila proizvoda se dobija množenjem svih individualnih grešaka. Ovo može biti značajno veća vrednosti u odnosu na ukupnu grešku dobijenu primenom pravila sumiranja, gde se ukupna greška dobija sabiranjem individualnih grešaka [93]. Ovo je značajna prednost pravila sumiranja i predstavlja prihvatljivo teorijsko objašnjenje superiornih eksperimentalnih rezultata dobijenih primenom pravila sumiranja (i izvedenih pravila) u praktičnim studijama [11], [12].

Pravila sumiranja i proizvoda predstavljaju osnovu za kombinovanje rezultata individualnih klasifikatora prilikom fuzije deskriptora. Brojne druge tehnike se mogu izvesti iz prethodna dva pravila, aproksimirajući granične vrednosti proizvoda ili sume verovatnoća [93]. U nastavku navodimo dva najčešće korišćena izvedena pravila.

3. **Pravilo maksimuma:** Ovo pravilo aproksimira sumu posteriornih verovatnoća sa maksimalnom vrednošću odgovarajućih verovatnoća. Matematički se može zapisati kao:

$$\omega^* = \operatorname{argmax}_{1 \leq j \leq C} (\max_k P_k(\omega_j | \mathbf{x})) \quad 6.3$$

4. **Pravilo većinskog glasanja (engl. majority voting)** Prvo ćemo izvršiti „binarizaciju“ posteriornih verovatnoća unutar jednog klasifikatora, na načina da se dodeli vrednost 1 samo maksimalnoj verovatnoći, a vrednost 0 svim ostalim:

$$v_{k,j} = \begin{cases} 1 & , \text{if } P_k(\omega_j | \mathbf{x}) = \max_i P_k(\omega_i | \mathbf{x}) \\ 0 & , \text{ostalo} \end{cases} \quad 6.4$$

Sada se na osnovu pravila sumiranja može matematički formulisati pravilo većinskog glasa

$$\omega^* = \operatorname{argmax}_{1 \leq j \leq C} \sum_{k=1}^K v_{k,j} \quad 6.5$$

Dakle, klasa koja dobije najveći broj glasova biće odabrana kao konačna (većinska) klasa ω^* , za zadati ulazni deskriptor \mathbf{x} .

Prethodna dva pravila pokazuju dobre rezultate u situacijama kada se kombinuje veliki broj klasifikatora, usled ublažavanja eventualne greške predikcije individualnih klasifikatora. Ovo je npr. slučaj kod metode mašinskog učenja Random Forest [98], gde se kombinuje stotine individualnih stabala odlučivanja.

6.2.1 Ansambli ELM klasifikatora

Prilikom fuzije klasifikatora, cilj je povezati što veći broj komplementarnih i nezavisnih klasifikatora koji se karakterišu što većom tačnošću rezultata. Intuitivno je jasno da povećanje

raznolikosti upotrebljenih klasifikatora dovodi do povećane tačnosti konačnog rezultata [99]. U radu [100] je uočeno da se raznolikost i visoka tačnost klasifikatora može postići korišćenjem nezavisnih klasifikatora baziranih na tehnikama generisanja slučajnih brojeva. Upravo u tome je nastala motivacija za kombinovanje većeg broja ranije opisanih ELM klasifikatora iste strukture, ali čije su težine ulaznog sloja različito generisane slučajnim procesom. Na ovaj način se postiže raznolikost individualnih ELM klasifikatora i značajno povećava tačnost sistema, iako se koriste klasifikatori potpuno iste strukture.

ELM algoritam za treniranje jednoslojnih neuronskih mreža pokazuje određene nedostatke u pogledu stabilnosti i stepena tačnosti klasifikacije, koji su posledica nasumičnog izbora ulaznih težina i pomeraja. Da bi dodatno prevazišli ove probleme i značajno povećali stabilnost i tačnost rezultata, vrši se kombinovanje (fuzija) većeg broja ELM mreža koje se inicijalizuju različitim ulaznim težinama [80], [101–104]. Ovakvi skupovi ELM klasifikatora, koji se treniraju nad istim skupom podataka i čiji se individualni rezultati klasifikacije kombinuju u jedinstveni izlaz, uobičajeno se nazivaju *ELM ansambli* (engl. ELM ensembles). U nastavku ćemo koristiti skraćenicu *ELM-E* da bismo označili ELM ansambl.

U slučaju kombinovanja više ELM klasifikatora, najčešće korišćena tehnika za kombinovanje rezultata individualnih ELM klasifikatora i izračunavanje konačnog rezultata klasifikacije celog ELM ansambla, je pravilo većinskog glasanja [64]. Nedostatak ove tehnike se može uočiti u slučaju kada verovatnoće većeg broja klasa imaju približno iste vrednosti dobijene primenom jednog ELM-a. Tada će postupak „binarizacije“ anulirati sve vrednosti osim one maksimalne, što može dovesti do velike greške predikcije ukoliko je načinjena mala greška u predviđanju posteriornih verovatnoća klasa. Ovakva aproksimacija kroz postupak „binarizacije“ ima smisla u slučaju ansambla sa velikim brojem individualnih klasifikatora (red veličine stotine), kakvi su Random Forest [70]. U ovakvim slučajevima se usled velikog broja klasifikatora smanjuje osetljivost celog ansambla na eventualne greške individualnih klasifikatora. Međutim, u slučaju kombinovanja relativno malog broja klasifikatora, pravilo većinskog glasanja može prouzrokovati značajno smanjenje tačnosti rezultata.

Zato u radu [11] predložem jednu tehniku za primenu pravila sumiranja u okviru ansambla ELM-a. Ovakva tehnika akumulira izlazne verovatnoće individualnih ELM klasifikatora, umesto akumuliranja njihovih binarnih glasova. Naime, ukoliko izlaz jednog ELM klasifikatora sadrži više bliskih vrednosti za verovatnoće pripadnosti različitim klasama, dodeljivanjem vrednosti „1“ samo klasi sa maksimalnom vrednošću izgubile bi se informacije o svim ostalim približnim vrednostima verovatnoća. Ovakva aproksimacija je opravdana u

slučaju velikog broja klasifikatora, npr. $K=500$ u metodi Random Forest [70], jer ima za posledicu smanjenu osetljivost ansambla na eventualne greške predikcije (šum) pojedinačnih rezultata. Međutim, u slučaju ELM ansambla (ELM-E), kada se kombinuje relativno mali broj klasifikatora (npr, 10), pokazali smo da ovo dovodi do umanjnja tačnosti od 3% prilikom klasifikacije slika.

Kompletan algoritam za treniranje i testiranje ELM ansambla (ELM-E) je predstavljen u nastavku. Da bismo predstavili algoritam za ELM-E koji sadrži K nezavisnih ELM klasifikatora, koristićemo prethodno uvedene oznake za ELM (Poglavlje 4.7), uz dodavanje indeksa klasifikatora (k) u eksponentu. Tako će npr. matrica ulaznih težina skrivenog sloja k -tog ELM-a biti označen kao $W_{in}^{(k)}$.

Algoritam ELM ansambla (ELM-E)

Trening faza algoritma

Ulaz: Trening skup podataka $T = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}; i = 1, \dots, M$.

for $k = 1$ *to* K // za svaki klasifikator $ELM^{(k)}$:

1. Dodeliti nasumične male vrednosti težinama $W_{in}^{(k)}$ i pomerajima $\mathbf{b}_{in}^{(k)}$ na ulazu skrivenog sloja klasifikatora $ELM^{(k)}$.
2. Izračunati matricu izlaza skrivenog sloja Z , na osnovu svih $\mathbf{x}^{(i)}, W_{in}^{(k)}, \mathbf{b}_{in}^{(k)}$, prema (4.43).
3. Izračunati težine na izlazu skrivenog sloja $W_{out}^{(k)} = YZ^\dagger$, korišćenjem pseudo-inverza $Z^\dagger = Z^T(ZZ^T)^{-1}$, opisanog u (Poglavlju 4.7).

endfor

Test faza algoritma

Ulaz: Test skup podataka $S = \{\mathbf{x}^{(j)}\}; j = M + 1, \dots, N$.

for $k = 1$ *to* K // za svaki klasifikator $ELM^{(k)}$:

1. Izračunati matricu izlaza skrivenog sloja Z , za primere test skupa S , korišćenjem težina $W_{in}^{(k)}$ i pomeraja $\mathbf{b}_{in}^{(k)}$ dodeljenih u 1. koraku trening faze.
2. Izračunati izlaz k -tog ELM klasifikatora mreže $Y_{test}^{(k)} = W_{out}^{(k)}Z$, korišćenjem težina $W_{out}^{(k)}$ izračunatih u 3. koraku trening faze.

endfor

-
- Sumirati svih K matrica rezultata: $Y_{test} = \sum_{k=1}^K Y_{test}(k)$.
 - Za svaki test primer (tj. za svaku vrstu matrice Y_{test}), odrediti oznaku klase, kao indeks maksimalnog elementa u toj vrsti.
-

U prethodnom algoritmu se uočava da prilikom testiranja, prvo izračunavamo srednju vrednost izlaznih rezultata svih ELM mreža u ansamblu (dovoljno je sumirati pošto je uvek isti broj klasa), a tek nakon toga određujemo labelu klase. Ovo je različito u odnosu na uobičajeni pristup primenom pravila većinskog glasanja, kada se prvo predvidi klasa na osnovu svakog od ELM klasifikatora, a zatim se ovo objedinjuje u konačan rezultat. Nedostatak metode većinskog glasanja je uočljiv u slučaju relativno malog broja klasifikatora u ansamblu (npr. 10), kada se usled ranog odlučivanja o klasi gube realni podaci o izlaznim rezultatima mreže (tj. verovatnoće pripadanja svakoj klasi). Na ovaj način se vrši rana „binarizacija“ rezultata individualnih ELM-a, koji su podložni nestabilnosti usled inicijalizacije slučajnim težinama.

Treba uočiti da uvođenje većeg broja ELM klasifikatora objedinjenih u ansamblima u cilju značajnog povećanja tačnosti klasifikacije, može povećati računsku složenost algoritma. Međutim, s obzirom da ELM ansamblu koriste individualne ELM klasifikatore potpuno iste strukture, kojima se samo dodeljuju različite težine veza. Ovo otvara mogućnosti za veoma efikasnu paralelizaciju ELM ansambla na CPU i/ili GPU (engl. Graphics Processing Unit). Relativno jednostavne i veoma efikasne tehnike paralelizovanih implementacija mogu ubrzati izvršenje klasifikacije primenom ansambla i za nekoliko redova veličine [70], [104].

Detaljni rezultati evaluacije tačnosti i vremenske kompleksnosti algoritma ELM-E su dati u narednom poglavlju. Takođe je analiziran uticaj izbora različitih parametara sistema na tačnost klasifikacije slika.

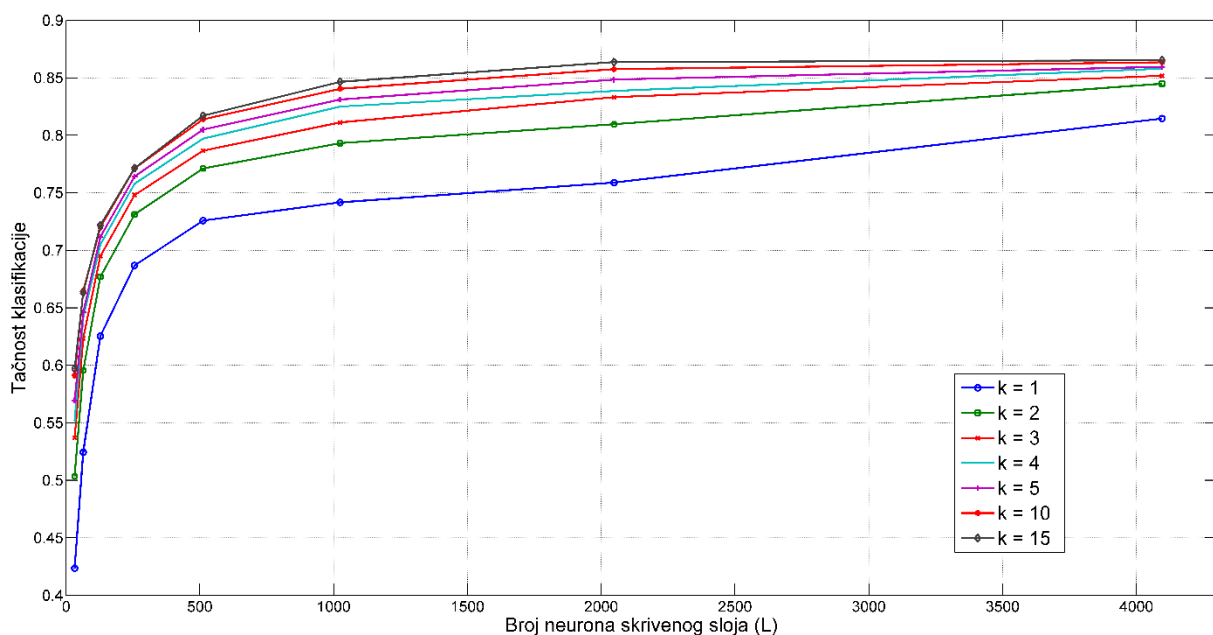
6.2.2 Eksperimenti sa ansamblima ELM klasifikatora

Naredni eksperimenti analiziraju tehniku kasne fuzije ELM klasifikatora, kroz formiranje ansambla klasifikatora, koja je predložena u radu [11]. Prilikom testiranja korišćena je prethodno opisana “Landuse21” kolekcija slika.

6.2.2.1 Parametri ELM-E

Ovaj skup eksperimenata sproveden je sa ciljem određivanja optimalnih parametara ELM ansambla (ELM-E). Obavljeno je merenje tačnosti klasifikacije u zavisnosti od promene parametara ELM-E. Ovi testovi su vršeni za fiksirane parametre MCBGP deskriptora slike (n_Y

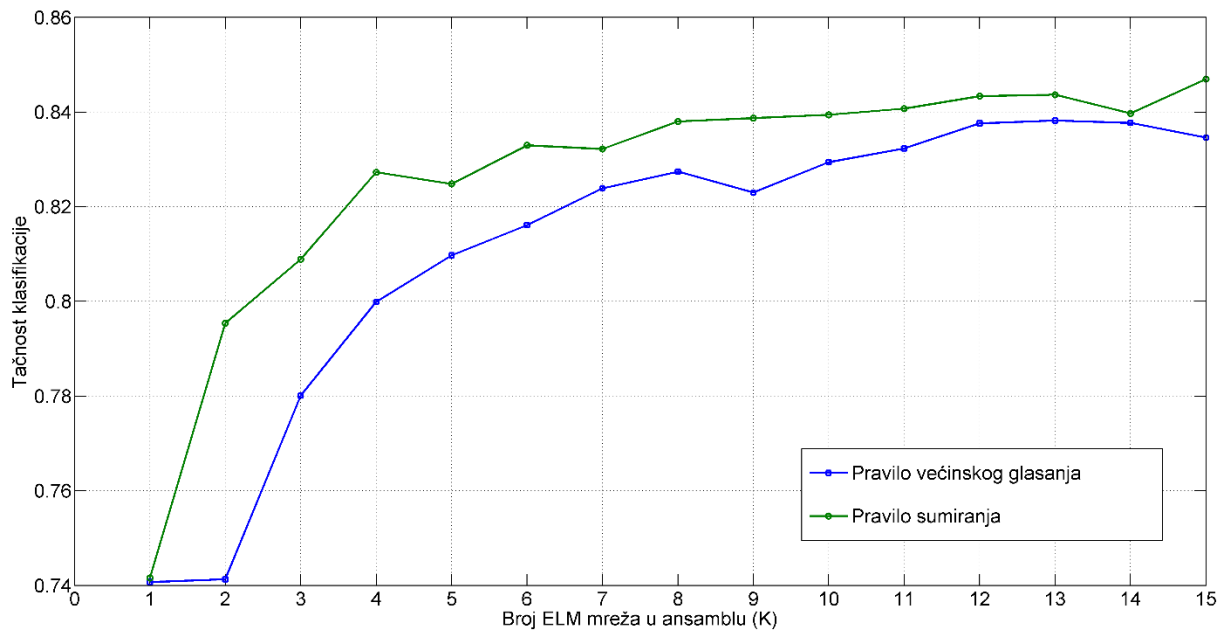
= 6, $n_{cb} = n_{Cr} = 4$, $rezol = 3$, $CLD=yes$), pri čemu je variran parametar L koji predstavlja broj neurona skrivenog sloja u okviru jednog ELM, i broj ELM klasifikatora u ansamblu (parametar k). Za agregaciju individualnih klasifikatora unutar ELM-E, korišćena je opisana tehnika agregacije rezultata sumiranjem [11]. Rezultate ovih eksperimenata ilustruje Slika 27. Uočava se da kombinovanje većeg broja ELM mreža u ansamblu ($k > 1$) povećava drastično tačnost klasifikacije. U poređenju sa korišćenjem jednog nezavisnog ELM klasifikatora ($k = 1$), uočava se značajan skok tačnosti od 10%, već za vrednost $k = 5$. Dalje povećanje vrednosti k dodatno poboljšava tačnost klasifikacije, uz primetno zasićenje za $k = 10$, nakon čega je unapređenje tačnosti zanemarljivo. Zaključuje se da je za primenu u praktičnim aplikacijama, preporučljivo eksperimentisati sa ansamblima koji sadrže više od 5 ELM mreža.



Slika 27 Uticaj broja neurona skrivenog sloja (L) i broja ELM klasifikatora u ansamblu (k), na tačnost klasifikacije slika iz kolekcije Landuse21. Svi individualni ELM klasifikatori unutar ELM-E ansambla imaju istu arhitekturu.

Naredni eksperimenti su obavljani sa ciljem testiranja uticaja različitih tehnika agregacije (kombinovanja) rezultata ELM klasifikatora unutar ansambla ELM-E. Konkretno je upoređena predložena tehnika sumiranja [11], u odnosu na najčešće korišćenu tehniku većinskog glasanja [101]. Prilikom ovog testiranja merena je tačnost klasifikacije, pri čemu je broj neurona skrivenog sloja svakog od ELM klasifikatora fiksiran na $L = 1000$, a variran je broj ELM unutar ELM-E. Rezultati koje prikazuje Slika 28 impiciraju da tehnika sumiranja rezultata ELM klasifikatora može povećati tačnost i do 3% u odnosu na najčešće primenjivanu tehniku većinskog glasanja, bez dodatne računске kompleksnosti. Objašnjenje za ovakve rezultate se može pronaći u slučaju kada verovatnoće većeg broja klasa imaju približno iste vrednosti

dobijene primenom jednog ELM-a. Tada će u postupku glasanja, kroz „binarizaciju“, biti eliminisane sve vrednosti osim one maksimalne, što može dovesti do velike greške konačne predikcije. Prethodna aproksimacija ima smisla u slučaju ansambla sa velikim brojem individualnih klasifikatora (npr. Random Forest), gde se usled velikog broja klasifikatora smanjuje osetljivost celog ansambla na eventualne greške individualnih klasifikatora. Međutim, u konkretnom slučaju gde je relativno mali broj ELM klasifikatora ($k < 15$), pravilo većinskog glasanja može prouzrokovati smanjenu tačnost rezultata.



Slika 28 Uticaj primenjene tehnike agregacije rezultata unutar ansambla, na tačnost klasifikacije.

Da bismo stekli precizniji uvid u tačnost klasifikacije za svaku od klasa, kao i detektovali klase u kojima se najčešće javljaju greške, u nastavku prikazujemo matricu konfuzije nad kolekcijom slika „Landuse21“ (Tabela 4). Od najvećeg značaja prilikom analize matrice konfuzije su vrednosti na glavnoj dijagonali, koje pokazuju tačnost klasifikacije za svaku od klasa ponaosob. Na osnovu rezultata prikazane konfuzione matrice, zaključuje se da najmanju tačnost prilikom klasifikacije pokazuju klase 5. „zgrade“ (70%) i 20. „cisterne“ (60%). Uočava se da klasa „zgrade“ pokazuje najveću sličnost sa klasom „naseljeno područje srednje gustine“, što je i očekivano. Takođe, klasa „cisterne“ pokazuje veliku sličnoost sa klasom „kamp-prikolice“. Ovde se otvara prostor za dodatnu analizu načina na koji bi se povećala raznolikost u prostoru deskriptora, kroz eventualno uvođenje novih deskriptora specifičnih za pomenute klase „zgrade“ i „cisterne“.

Tabela 4 Matrica konfuzije za kolekciju slika "Landuse21". Zbog kompaktnosti prikaza, nazivi klasa su označeni rednim brojevima: 1. poljoprivreda, 2. avioni, 3. bejzbol igralište, 4. plaža, 5. zgrade, 6. prerija, 7. gusto naseljeno područje, 8. šuma, 9. autoput, 10. teren za golf, 11. luka, 12. raskrsnica, 13. naseljeno područje srednje gustine, 14. kamp-prikolice, 15. nadvožnjak, 16. parking, 17. reka, 18. aerodromska pista, 19. retko naseljeno područje, 20. cisterne i 21. teniski tereni.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	
1.	0.95	0	0	0	0	0	0	0	0.05	0	0	0	0	0	0	0	0	0	0	0	0	
2.	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3.	0	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4.	0	0	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0	0	0	0	
5.	0	0	0	0.05	0.70	0	0	0	0	0	0	0	0.15	0	0	0	0.05	0	0	0.05	0	
6.	0	0	0	0	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7.	0	0	0	0	0	0	0.90	0	0	0	0	0	0.05	0	0	0	0	0	0	0	0.05	
8.	0	0	0	0	0	0	0	0.85	0	0	0	0	0	0	0	0	0	0	0.15	0	0	
9.	0.05	0	0	0	0	0	0	0	0.90	0	0	0	0	0	0.05	0	0	0	0	0	0	
10.	0	0	0.05	0	0	0	0	0	0	0.95	0	0	0	0	0	0	0	0	0	0	0	
11.	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0	0	0	0	0	0	
12.	0	0	0	0	0	0	0	0	0	0	0	0.95	0	0	0	0	0	0	0	0	0.05	
13.	0	0	0	0	0	0	0	0.05	0	0	0	0	0.85	0	0	0	0.05	0	0.05	0	0	
14.	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0	0	0	
15.	0	0	0	0	0	0	0	0	0.05	0	0	0.10	0	0	0.85	0	0	0	0	0	0	
16.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0	
17.	0	0	0	0	0	0	0	0	0.05	0	0	0	0.10	0	0	0	0.85	0	0	0	0	
18.	0	0	0	0	0	0	0	0	0.05	0	0	0	0	0	0	0	0	0.95	0	0	0	
19.	0.05	0	0.05	0	0	0	0	0.05	0	0	0	0	0	0	0	0	0	0	0	0.85	0	
20.	0	0	0.05	0	0.05	0	0	0	0.05	0	0	0.05	0	0.10	0	0	0	0	0	0	0.60	0.10
21.	0	0	0.05	0	0	0	0	0	0	0	0	0	0.05	0	0	0	0	0	0	0.10	0.80	

U narednom eksperimentu je obavljena analiza uticaja veličine trening skupa na rezultate klasifikacije, primenom različitih tehnika klasifikacije. Kao deskriptor je korišćen prethodno opisani MCBGP. Veličina trening podskupa je varirana u opsegu od 80% do 20%. Dodatno su prikazani rezultati koji su prethodno objavljeni u literaturi za kolekciju slika „Landuse21“. Tabela 5 pokazuje visok stepen tačnosti predložene metode ELM-E u kombinaciji sa MCBPG [11]. Relativno male vrednosti standardne devijacije su pokazale da ansambl ELM-E poseduje visoku stabilnost rezultata klasifikacije, za razliku od individualnih ELM klasifikatora. Važno je uočiti da ELM-E postiže tačnost rezultata koja je značajno veća od linearnog SVM, a približna po vrednosti sa nelinearnim SVM-RBF, uz znatno kraće vreme treniranja od SVM-

RBF. Dakle, ELM-E u kombinaciji sa odgovarajuće tehnike agregacije rezultata, može poslužiti kao veoma brza alternativa nelinearnoj SVM metodi.

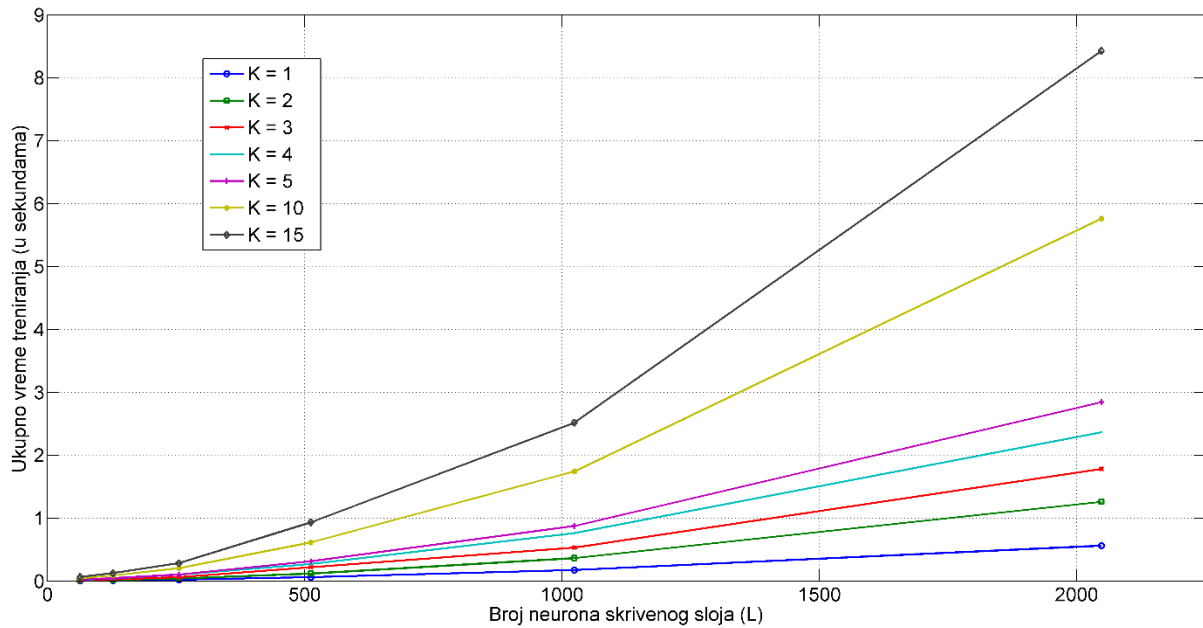
Tabela 5 Analiza uticaja veličine trening skupa na rezultate klasifikacije, primenom različitih metoda klasifikacije. Prikazana je tačnost klasifikacije uz standardnu devijaciju.

Metoda	Procenat trening slika od ukupnog broja slika u kolekciji		
	80%	50%	20%
ELM-E (L=4000, k=10)	86.52 (± 1.29)	82.11 (± 0.99)	70.29 (± 1.22)
ELM-E (L=2000, k=10)	85.75 (± 1.43)	81.36 (± 0.93)	69.52 (± 1.31)
ELM-E (L=1000, k=10)	84.31 (± 1.63)	80.23 (± 1.33)	69.27 (± 1.32)
RBF SVM	86.47 (± 1.19)	81.46 (± 1.16)	70.41 (± 1.29)
Random Forest	84.12 (± 1.41)	79.61 (± 1.23)	70.18 (± 1.48)
Linear SVM	80.30 (± 1.58)	76.83 (± 1.08)	66.90 (± 1.14)
SPCK++ [38]	77.38	-	-
SPM [16]	74.00	-	-

6.2.2.2 Vremenske performanse

Vremenske performanse prethodnih algoritama su analizirane na primeru MATLAB implementacije koja je testirana na procesoru Intel Core i5 2.9GHz. Prosečno izmereno vreme za izračunavanje MCBGP deskriptora iznosilo je 120ms za parametre ($n_Y = 6, n_{Cb} = n_{Cr} = 4, rezol. = 3, CLD = da$) i slike rezolucije 256×256 piksela. Merenje vremena potrebnog za treniranje ELM-E klasifikatora je obavljeno nad skupom od 1680 trening slika, pri čemu je variran broj neurona u skrivenom sloju (L), kao i broj ELM mreža u ELM-E ansamblu (K). Vremenske performanse, izražene u sekundama, prikazuje Slika 29. Uočava se da vreme treniranja ELM-E ansambla raste približno linearno u zavisnosti od (L) i od (K), s obzirom da prilikom implementacije nije primenjena nikakva tehnika paralelizacije.

Vremenske performanse ELM-E mogu biti značajno poboljšane ukoliko se metoda implementira primenom modernih tehnika paralelizacije na CPU ili GPU. U poređenju sa drugim popularnim metodama mašinskog učenja, npr. nelinearnim SVM, paralelizacija ELM-E je relativno jednostavna za implementaciju [70], [104], jer koristi ELM instance koje su potpuno iste strukture. Ovakve paralelizovane implementacije na modernim GPU mogu smanjiti vreme izvršenja ELM-E i za nekoliko redova veličine u odnosu na prezentovane rezultate prototipa implemntiranog u Matlab-u.



Slika 29 Vreme treniranja ELM-E za celokupan trening skup od 1680 slika. Parametar K označava broj individualnih ELM klasifikatora u ansamblu ELM-E.

U nastavku je obavljeno poređenje vremenskih performansi metodel ELM-E i ostalih popularnih klasifikacionih metoda: Random Forest [98], SVM-linearni i SVM-RBF [1]. Tabela 6 prikazuje uporedne rezultate vremena treniranja i testiranja pomenutih metoda. Za Random Forest metodu je korišćeno 500 stabala, dok je broj elemenata deskriptor vektora koji je korišćen za svaku podelu jednak kvadratnom korenu dimenzija vektora. U slučaju SVM-RBF, optimalni parametri C i γ su dobijeni unakrsnom validacijom (5 delova) nad diskretnim vrednostima u opsegu $C = [2^{-4}, 2^{-2} \dots, 2^{10}]$ i $\gamma = [2^{-10}, 2^{-8} \dots, 2^4]$. Parametar margine za linearni SMV je postavljen na konstantnu vrednost $C = 1$.

Tabela 6 Poređenje vremena treniranja (1680 slika) i vremena testiranja (420 slika), primenom različitih klasifikacionih metoda.

Metoda	Vreme treniranja (s)	Vreme testiranja (s)
ELM-E (L=4000, k=10)	23.91	0.59
ELM-E (L=2000, k=10)	6.30	0.28
ELM-E (L=1000, k=10)	1.89	0.13
SVM-RBF	345.73	0.23
Random Forest	27.85	0.04
SVM-linearni	0.49	0.17

Rezultati u prethodnoj tabeli pokazuju da linearni SVM zahteva minimalno vreme za treniranje, međutim pokazuje znatno manju tačnost klasifikacije. Treniranje ELM-E ansambla traje značajno manje u odnosu na SVM-RBF, pri čemu obe metode pokazuju veoma visoku tačnost klasifikacije. U pogledu vremenskih performansi u fazi testiranja, metoda Random Forest pokazuje najbolje rezultate, dok ostale metode imaju slična vremena izvršenja. Dodatna analizu vremenskih performansi metoda zasnovanih na ELM, može se pogledati u nekolicini drugih radova [72], [73], [101].

S obzirom da SVM-RBF predstavlja jedan od najzastupljenijih metoda klasifikacije, treba uočiti potencijal primene ELM-E u odnosu na ovu metodu. Naime, pokazano je u ranijim studijama da je tačnost nelinearne SVM-RBF metode veoma osetljiva na izbora optimalnih parametara C i γ tokom treniranja modela [72]. Zato je prilikom treniranja SVM-RBF neophodno obaviti vremenski zahtevnu proceduru pronalaženja optimalnih vrednosti ovih parametara. Nasuprot tome, ELM-E ansambl pokazuje stabilnost u pogledu broja neurona i regularizacionog parametra, sve dok je broj neurona u skrivenom sloju dovoljno veliki (Slika 29). Dakle, ELM-E ne zahteva računski složene operacije prilikom treniranja, i zato pokazuje dominantno bolje vremenske performanse postupka treniranja u odnosu na SVM-RBF. Osim toga, ELM-E se efikasno prilagođava porastu broja primera trening skupa [105], dok RBF-SVM pokazuje eksponencijalni rast vremena treniranja prilikom povećanja broja trening primera [72].

Generalni zaključak koji se može izvesti na osnovu svih prethodnih eksperimenata je da ELM-E metoda uz adekvatni deskriptor slike i tehniku agregacije rezultata, postiže veoma visoku tačnost klasifikacije uz malu vremensku zahtevnost procesa treniranja. Osim toga, u odnosu na druge metode, ELM-E koristi relativno jednostavan matematički aparat koji je pogodan za efikasnu implementaciju.

6.3 Agregacije rezultata klasifikatora primenom mašinskog učenja

Prethodno opisana tehnika podrazumeva da svi klasifikatori imaju podjednak značaj prilikom kombinovanja njihovih rezultata. Naprednija tehnika agregacije rezultatata klasifikatora se može realizovati uvođenjem dodatnog klasifikatora koji će biti treniran na osnovu rezultata individualnih klasifikatora predstavljenih kao novi deskriptor. Ovakva tehnika se uobičajeno sreće pod nazivom *slaganje klasifikatora* (engl. stacking, stacked generalization) [94].

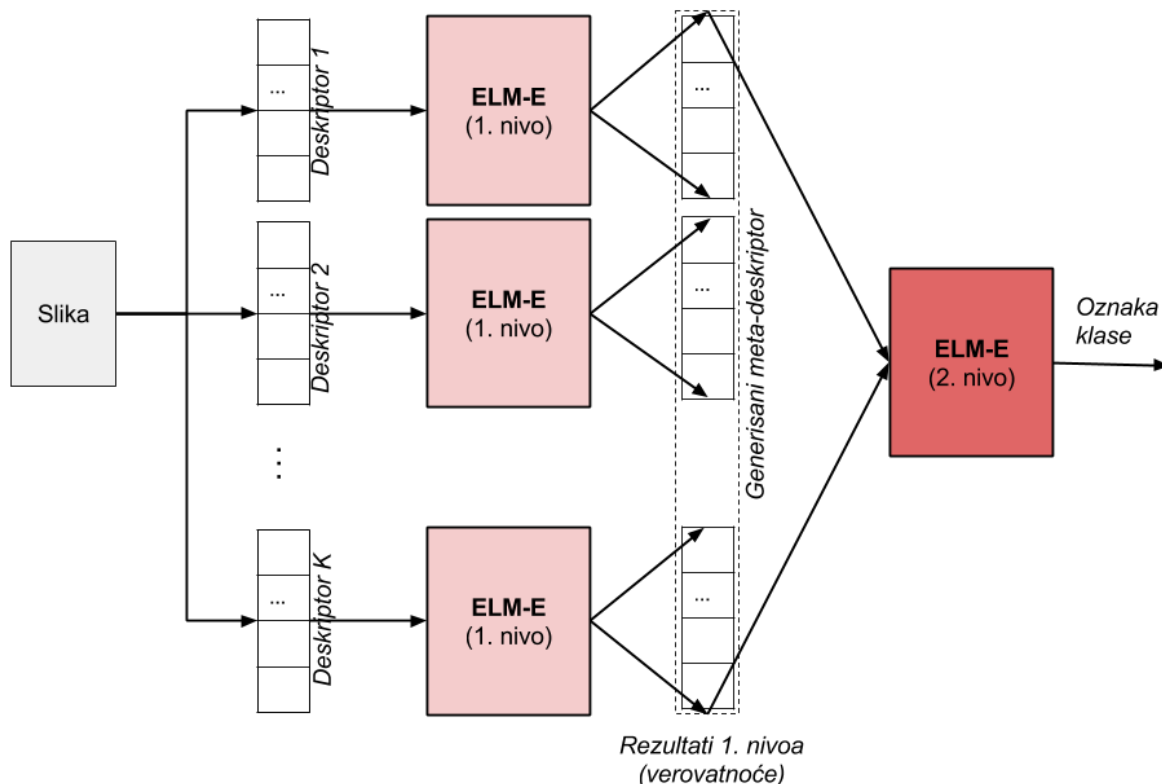
Postupak slaganja klasifikatora se obavlja u dva (ili više) nivoa. U prvom nivou se vrši treniranje više nezavisnih klasifikatora, pri čemu ulaz svakog klasifikatora može biti isti ili drugačiji deskriptor. U drugom nivou, rezultati iz prvog nivoa se dovode na ulaz novog jedinstvenog klasifikatora. Na ovaj način, klasifikator u drugom nivou može naučiti optimalnu kombinaciju klasifikatora za svaku od klasa. Pojedini algoritmi nadgledanog mašinskog učenja su uspešno iskoristili ovu tehniku za unapređenje tačnosti individualnih modela za probleme regresije [95] i klasifikacije [13], [96].

6.3.1 Predložena hijerarhijska metoda za kasnu fuziju ELM klasifikatora

Metoda koja će biti opisana u nastavku, omogućava kombinovanje više komplementarnih deskriptora kroz primenu kasne fuzije ELM-E klasifikatora. Predložena metoda obezbeđuje da deskriptori koji nose najvažniju informaciju za konkretnu sliku, dobiju veći značaj u konačnom rezultatu predikcije. Metoda je detaljno predstavljena u radu [13].

Princip na kome se zasniva predložena hijerarhijska metoda kasne fuzije ELM klasifikatora (*H-ELM-E*) [13] prikazuje Slika 30. Celokupan postupak klasifikacije se obavlja hijerarhijski, u dva nivoa, pri čemu se kao osnovni klasifikator koristi ELM ansambl (ELM-E). U prvom nivou se vrši treniranje više nezavisnih ELM-E, pri čemu ulaz svakog ELM-E predstavlja drugačiji deskriptor slike. U drugom nivou, izlazne verovatnoće iz prvog nivoa formiraju novi deskriptor (engl. mid-level descriptor) koji se dovodi na ulaz novog jedinstvenog ELM-E klasifikatora. Na ovaj način, u prvom nivou, svaki ELM-E će naučiti kako da obavi klasifikaciju na osnovu jedne vrste deskriptora slike, dok će ELM-E u drugom nivou naučiti optimalnu kombinaciju deskriptora za svaku od klasa. Ovakav pristup obezbeđuje da iz skupa različitih deskriptora slike, za svaku od klasa budu odabrani oni deskriptori koji na optimalan način reprezentuju tu klasu.

Neka trening skup sadrži M označenih slika (slika čije su klase poznate), gde je za svaku sliku izdvojeno K različitih vrsta deskriptora označenih kao: $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K$. U prvom nivou predloženog H-ELM-E postupka, nezavisno se trenira K osnovnih ELM-E klasifikatora, po jedan za svaki tip deskriptora. Za svaki ELM-E klasifikator, na ulaz se dovodi jedan tip deskriptora slike, dok se na izlazu dobija C -dimenzioni vektor verovatnoća pripadanja klasama, gde je C broj klasa. U skladu sa oznakama sa Slika 26, označićemo izlazni vektore i -tog klasifikatora iz prvog nivoa kao $\mathbf{P}_i = [P_i(\omega_1|\mathbf{x}_i), P_i(\omega_2|\mathbf{x}_i), \dots, P_i(\omega_C|\mathbf{x}_i)]$; $i = 1, \dots, K$.



Slika 30 Hijerarhijski postupak kasne fuzije (H-ELM-E), primenom ELM ansambla

Da bismo formirali ulazni vektor za jedinstveni ELM-E klasifikator u drugom nivou, nadovezaćemo C -dimenzione izlazne vektore svih K klasifikatora iz prvog nivoa. Na ovaj način smo generisali novi ulazni vektor dimenzija $K \cdot C$, tzv. meta-deskriptor (engl. mid-level descriptor, meta-descriptor) koji se može predstaviti kao $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_K]$. Treniranje ELM-E klasifikatora drugog nivoa se vrši na osnovu formiranih meta-deskriptora \mathbf{P} koji predstavljaju ulaze, pri čemu se koriste poznate klase ulaznih slika trening skupa. Izlaz drugog sloja predstavlja konačni izlazni vektor verovatnoća, gde indeks maksimalnog elementa predstavlja konačno predviđenu klasu.

U nastavku je predstavljen algoritam za treniranje i testiranje H-ELM-E klasifikatora. Za označavanje svakog od K nezavisnih ELM-E klasifikatora u 1. nivou, korišćemo indeks k u oznaci $ELM-E_k$.

Algoritam H-ELM-E

Trening faza algoritma

Ulaz: Trening skup podataka $T = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}; i = 1, \dots, M$.

for $k = 1$ *to* K // za svaki klasifikator $ELM-E_k$ u 1. nivou

- Obaviti treniranje $ELM-E_k$ na osnovu celokupnog trening skupa T , prema algoritmu ELM-E iz poglavlja 6.2.1.
- *for* $i = 1$ *to* M // za svaku sliku iz trening skupa
Izračunati vektor izlaznih verovatnoća $\mathbf{P}_k^{(i)}$ za klasifikator $ELM-E_k$.

endfor

endfor

- Formirati objedinjeni meta-deskriptor za svaku trening sliku $\mathbf{P}^{(i)} = [\mathbf{P}_1^{(i)}, \mathbf{P}_2^{(i)}, \dots, \mathbf{P}_K^{(i)}]; i = 1, \dots, M$.
- Obaviti treniranje $ELM-E$ u 2. nivou, na osnovu generisanih meta-deskriptora za celokupni trening skup $T' = \{(\mathbf{P}^{(i)}, \mathbf{y}^{(i)})\}; i = 1, \dots, M$, prema Algoritmu ELM-E (Poglavljje 6.2.1).

Test faza algoritma

Ulaz: Test skup podataka $S = \{\mathbf{x}^{(j)}\}; j = M + 1, \dots, N$.

for $k = 1$ *to* K // za svaki klasifikator $ELM-E_k$ u 1. nivou

for $j = M+1$ *to* N // za svaku sliku iz test skupa

- Izračunati vektor izlaznih verovatnoća $\mathbf{P}_k^{(j)}$ za klasifikator $ELM-E_k$.

endfor

endfor

for $j = M+1$ *to* N // za svaku sliku iz test skupa

- Formirati objedinjeni meta-deskriptor za test sliku $\mathbf{P}^{(j)} = [\mathbf{P}_1^{(j)}, \mathbf{P}_2^{(j)}, \dots, \mathbf{P}_K^{(j)}]$.
- Izračunati vektor izlaznih verovatnoća iz $ELM-E$ u 2. nivou, na osnovu generisanog meta-deskriptora $\mathbf{P}^{(j)}$, prema algoritmu ELM-E iz poglavlja 6.2.1.

endfor

Treba uočiti nekoliko važnih osobina predložene hijerarhijske metode kasne fuzije ELM klasifikatora (H-ELM-E). Naime, prilikom treniranja celog H-ELM-E klasifikatora, ulazne težine svih ELM-E klasifikatora iz prvog i drugog nivoa postavljaju se nezavisno, na vrednost

nasumično generisanih brojeva, u skladu sa ELM teorijom [71]. Prilikom treniranja svih ELM-E klasifikatora u oba nivoa, koristi se isti trening skup slika uz odgovarajuću labelu za svaku sliku (tj. oznaku klase). Treba primetiti da u predloženom H-ELM-E metodu nije neophodno vršiti normalizaciju različitih tipova deskriptor vektora, s obzirom da se svaka vrsta deskriptora nezavisno klasifikuje u prvom sloju ELM-E, a da se zatim izlazni vektori ELM-E, koji su već normalizovani na opseg $[0, 1]$, kombinuju u drugom sloju. Dakle, ne postoji kombinovanje raznorodnih deskriptora, kao u slučaju rane fuzije kada je normalizacija neophodna.

U konkretnoj implementaciji [13], koja je upotrebljena za evaluaciju predloženog opšteg hijerarhijskog postupka H-ELM-E, korišćeno je $K = 4$ klasifikatora ELM-E u prvom nivou. Pri tome, svaki ELM-E je treniran sa drugačijom vrstom deskriptora ulazne slike, čiji su detalji predstavljeni u narednom poglavlju. Međutim, treba istaći da je predloženi hijerarhijski postupak dovoljno opšti da se može koristiti sa proizvoljnim brojem i drugačijim vrstama deskriptora.

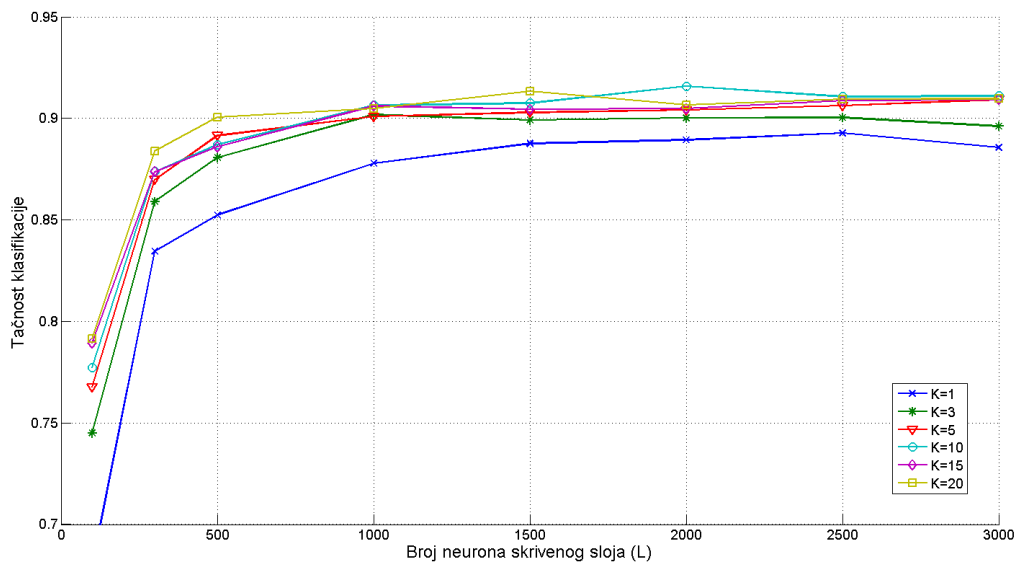
6.3.2 Eksperimentalna evaluacija hijerarhijske fuzije

U nastavku se daju rezultati eksperimentale evaluacije predložene H-ELM-E metode za kasnu fuziju, od kojih je većina objavljena u radu [13]. Svi eksperimenti su obavljani uporedno nad tri javno dostupne kolekcije slika: “Landuse 21”, “8-scenes” i “AID”.

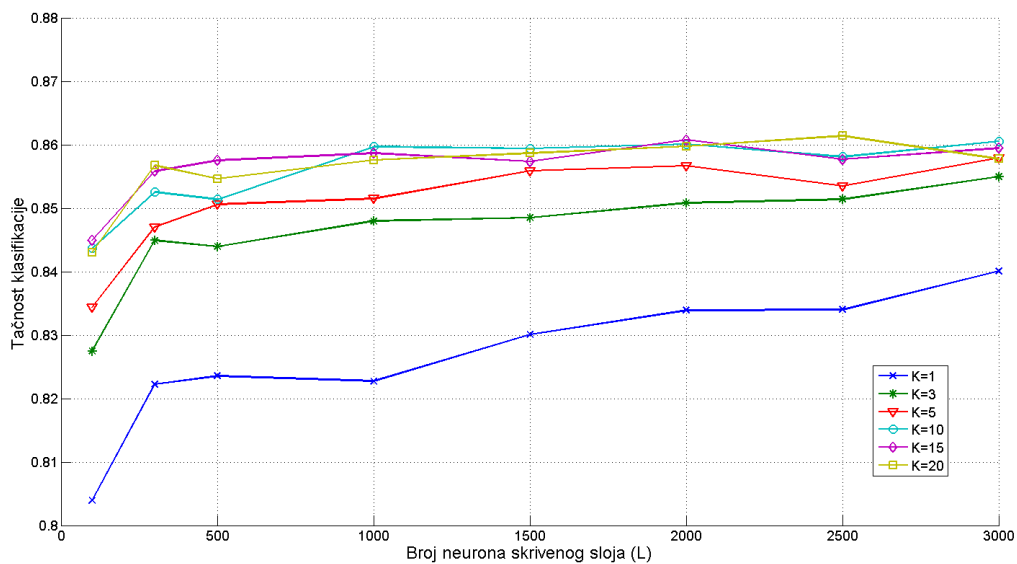
6.3.2.1 Parametri osnovnih ELM-E klasifikatora

Prva grupa eksperimenata je sprovedena sa ciljem određivanja optimalnih parametara osnovnih klasifikatora ELM-E. Merena je tačnost klasifikacije u zavisnosti od različitih parametara ELM-E, pri čemu je variran broj neurona (L) u skrivenom sloju svakog ELM, kao i broj ELM u ansamblu (K). U svim testovima, fiksirana je vrednost ELM parametra $C = 0.1$. Eksperimentalni rezultati koje prikazuju Slika 31, Slika 32 i Slika 33, pokazuju da povećanje broja neurona, kao i korišćenje ansambla od više ELM klasifikatora ($K > 1$), doprinosi boljim konačnim rezultatima klasifikacije. Ovo pokazuje značaj korišćenja ansambla ELM-E umesto korišćenja jednog ELM kao osnovnog klasifikatora u okviru predložene hijerarhijske metode. Osim toga, ELM-E doprinosi stabilnosti rezultata i otpornosti na šumove u podacima. Iz prikazanih eksperimentalnih rezultata, zaključuje se da je značajan porast tačnosti rezultata klasifikacije uočljiv do $L = 1000$ i $K = 5$. Prilikom daljeg povećanja vrednosti L i K , uočava se pojava zasićenja tačnosti pri čemu se javlja samo neznatni rast tačnosti, uz usložnjavanje

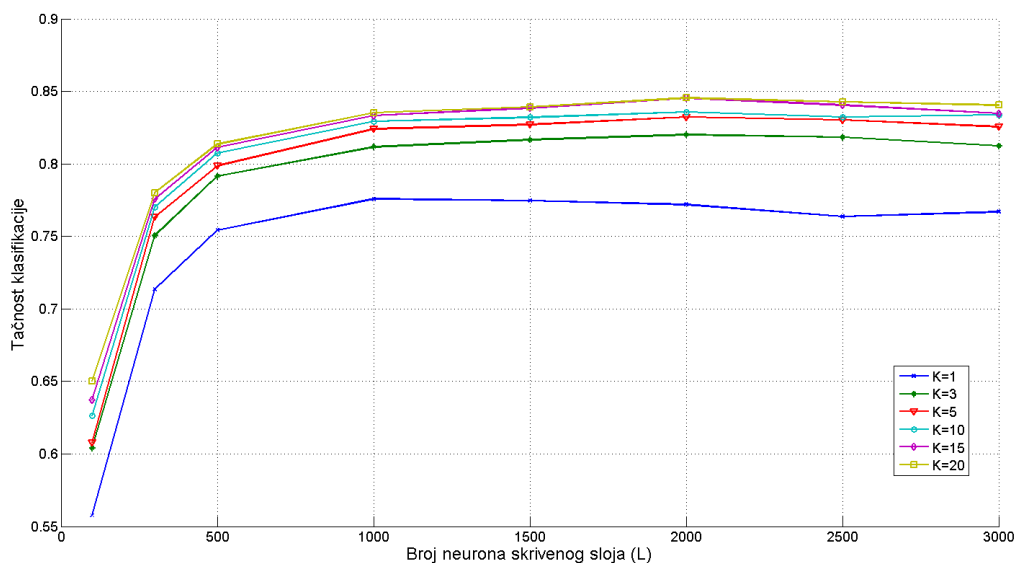
modela koje dovodi do značajnog povećanja računске kompleksnosti. Zato se ove vrednosti parametara mogu uzeti kao optimalne za kolekcije slika nad kojima je obavljeno testiranje.



Slika 31 Tačnost klasifikacije predložene H-ELM-E metode nad kolekcijom slika „Landuse21“, u zavisnosti od broj neurona (L) skrivenog sloja svakog ELM, kao i broja ELM mreža u ansamblu (K).



Slika 32 Tačnost klasifikacije predložene H-ELM-E metode nad kolekcijom slika „8-scenes“, u zavisnosti od broj neurona (L) skrivenog sloja svakog ELM, kao i broja ELM mreža u ansamblu (K).



Slika 33 Tačnost klasifikacije predložene H-ELM-E metode nad kolekcijom slika „AID“, u zavisnosti od broj neurona (L) skrivenog sloja svakog ELM, kao i broja ELM mreža u ansamblu (K).

Prilikom testiranja na najzahtevnijoj kolekciji slika „AID“ (Slika 33) posebno je uočljiv porast tačnosti i do 10% prilikom korišćenja $K > 1$ ELM mreža u okviru ansambla. Ovo pokazuje pravi značaj korišćenja ansambla ELM u okviru predložene H-ELM-E metode.

6.3.2.2 Uporedna analiza tačnosti klasifikacije H-ELM-E

Da bismo analizirali značaj fuzije deskriptora primenom H-ELM-E, uporedili smo tačnost klasifikacije u odnosu na korišćenje nezavisnih deskriptora, i u odnosu na često korišćenu tehniku rane fuzije (Tabela 7). Napominjemo da je prilikom eksperimenata u kojima je korišćena metoda rane fuzije, vršena normalizacija deskriptor vektora metodom “z-score” koja je ranije opisana. Ovom metodom se elementi deskriptor vektora normalizuju na nultu srednju vrednost i jediničnu varijansu, pre primene klasifikatora. Ovo ima efekat umanjavanja šuma u podacima, kroz eliminisanje ekstremno velikih vrednosti. U ovim eksperimentima je korišćeno 80% slika za treniranje i 20% slika za testiranje. Eksperimentalni rezultati koje prikazuje Tabela 7 impliciraju da tačnost klasifikacije može biti značajno unapređena fuzijom više komplementarnih deskriptora, uz primenu odgovarajuću strategiju za fuziju, kakva je H-ELM-E. Važan zaključak je da predložena H-ELM-E metoda može povećati tačnost klasifikacije slika za 3-6%, u odnosu na masovno korišćenu tehniku rane fuzije. Posebno treba obratiti pažnju na analizu rezultata nad zahtevnom „AID“ kolekcijom slika. Ova kolekcija poseduje izuzetno veliku sličnost među klasama i ovde se uočava prava snaga kasne fuzije klasifikatora u odnosu

na korišćenje nezavisnih klasifikatora i rane fuzije. Na osnovu standardne devijacije svih rezultata se može zaključiti da H-ELM-E pokazuje visok nivo stabilnosti rezultata.

Tabela 7 Tačnost klasifikacije H-ELM-E u odnosu na korišćenje nezavisnih deskriptora i masovno korišćenu tehniku rane fuzije. Rezultati testiranja su prikazani za tri kolekcije slika "Landuse21", "8-scenes" i "AID".

Deskriptor	Klasifikator	Tačnost klasifikacije ("Landuse21")	Tačnost klasifikacije ("8-scenes")	Tačnost klasifikacije ("AID")
BGP (156-dim)	ELM-E (1-nivo)	82.64 (± 1.65)	79.65 (± 0.59)	69.85 (± 0.86)
LBP (174-dim)	ELM-E (1-nivo)	83.42 (± 1.76)	80.34 (± 0.73)	62.36 (± 1.59)
CLD (22-dim)	ELM-E (1-nivo)	48.34 (± 1.89)	56.42 (± 1.02)	42.46 (± 0.60)
RGB (48-dim)	ELM-E (1-nivo)	68.54 (± 1.52)	47.62 (± 0.86)	50.96 (± 1.12)
BGP+LBP+CLD+RGB (rana fuzija)	ELM-E (1-nivo) L=1000, K=10	88.41 (± 1.54)	83.52 (± 0.43)	80.21 (± 0.56)
BGP+LBP+CLD+RGB (rana fuzija + PCA)	ELM-E (1-nivo) L=1000, K=10	83.12 (± 1.59)	81.43 (± 0.62)	74.13 (± 0.71)
BGP, LBP, CLD, RGB (kasna fuzija)	H-ELM-E (2-nivoa) L=1000, K=10	90.54 (± 1.43)	85.63 (± 0.57)	85.03 (± 0.42)
BGP, LBP, CLD, RGB (kasna fuzija)	H-ELM-E (2-nivoa) L=2000, K=10	91.45 (± 1.25)	86.62 (± 1.12)	86.17 (± 0.88)

Dodatni eksperimenti su sprovedeni sa ciljem analize uticaja PCA tehnike za redukciju dimenzionalnosti podataka prilikom rane fuzije [68], [88]. Ova tehnika za redukovanje dimenzionalnosti vektora deskriptora se uobičajeno primenjuje zbog ubrzanja izračunavanja, kao i nemogućnosti pojedinih klasifikatora da uspešno rade sa deskriptorima velike dimenzionalnim i velikog stepena korelacije. Međutim, neuronske mreže (uključujući i ELM-E) pokazuju visok stepen uspešnosti u radu sa visoko-dimenzionalnim i korelisanim deskriptorima [44], [105]. Tabela 7, između ostalog, prikazuje rezultat rane fuzije gde je korišćena PCA redukcija na 100 dominantnih dimenzija nakon konkatencije svih deskriptor vektora. Rezultati pokazuju da primena PCA za redukciju dimenzionalnosti dovodi do

značajnog smanjenja tačnosti prilikom opisanih eksperimenata, uz povećanje efikasnosti izračunavanja.

Naredni eksperimenti imaju za cilj da uporede tačnost predložene H-ELM-E metode sa relevantnim rezultatima objavljenim u literaturi. Rezultate tačnosti klasifikacije nad kolekcijama slika "Landuse21", "8-scenes" i "AID" u zavisnosti od broja trening slika, prikazuju Tabela 8, Tabela 9 i Tabela 10. Da bi rezultati bili uporedivi sa onim koji su objavljeni u literaturi, za kolekciju "Landuse21" je variran broj trening slika svake klase kao 80, 50 i 20; za kolekciju "8-scenes" korišćeno 200, 100 i 50 slika po klasi; dok je za kolekciju „AID“ iskorišćeno 80% i 50% slika za trening. Na osnovu rezultata koje prikazuju Tabela 8, Tabela 9 i Tabela 10 može se zaključiti da H-ELM-E pokazuje veoma visok stepen tačnosti klasifikacije, koji prevazilazi većinu dosadašnjih rezultata iz literature.

Tabela 8 Poređenje tačnosti predložene metode H-ELM-E u odnosu na aktuelne rezultate iz literature, u zavisnosti od broja trening slika (kolekcija slika „Landuse21“).

Metoda	Broj trening slika za svaku klasu		
	80	50	20
SPM [16]	74.00	-	-
SPCK++ [38]	77.38	-	-
MCBGP [11]	86.52	82.11	70.29
MCFI [106]	88.20	-	-
mCENTRIST [41]	89.90	-	-
H-ELM-E (L=1000, K=10) [13]	90.54	86.76	75.93
H-ELM-E (L=2000, K=10) [13]	91.45	86.81	76.15

Tabela 9 Poređenje tačnosti predložene metode H-ELM-E u odnosu na aktuelne rezultate iz literature, u zavisnosti od broja trening slika (kolekcija slika „8-scenes“).

Metoda	Broj trening slika za svaku od klasa		
	200	100	50
LCVBP [107]	-	76.00	-
PM [64]	-	82.00	-
MCBGP [11]	84.87	82.31	78.97
CENTRIST [37]	-	86.22	-
H-ELM-E (L=1000, K=10)	87.89	85.63	82.53
H-ELM-E (L=2000, K=10)	87.92	86.62	82.64

Tabela 10 Poređenje tačnosti predložene metode H-ELM-E u odnosu na aktuelne rezultate iz literature, u zavisnosti od broja trening slika (kolekcija slika „AID“).

Metoda	Procenat trening slika za svaku od klasa	
	80%	50%
BoW+SIFT [32]	-	68.37
FisherKernel+SIFT [28]	-	78.99
DeepLearning-GoogLeNet [50]	-	86.39
H-ELM-E (L=1000, K=10)	85.03	83.32
H-ELM-E (L=2000, K=10)	86.17	83.62

Treba uočiti da su najbolji rezultati koje prikazuje Tabela 10 dobijeni primenom veoma složenih Deep Learning mreža [45]. Ovakve mreže, za razliku od predložene H-ELM-E arhitekture, zahtevaju višednevno treniranje na veoma naprednim klasterima GPU, uz korišćenje miliona slika. Konkretna mreža koja je korišćena u radu [45] trenirana je na kolekciji ImageNet koja sadrži milion slika klasifikovanih u 1000 kategorija, nakon čega je iskorišćena za izračunavanje 1024-dimenzionalnih deskriptora slike na osnovu izlaza poslednjeg sloja. Ovakvi deskriptori su zatim iskorišćeni za treniranje i testiranje linearnog SVM [104] na konkretnoj “AID” kolekciji. Nasuprot tome, H-ELM-E je kompletno trenirana i testirana isključivo na znatno manjoj “AID” kolekciji.

Tabela 11 Primer matrice konfuzije za kolekciju slika “8-scenes”.

	obala	šuma	autoput	grad	planina	proplanak	ulica	zgrada
obala	0.91875	0	0.00625	0	0.03125	0.04375	0	0
šuma	0	0.92968	0	0	0.06250	0.00781	0	0
autoput	0.05000	0	0.81666	0.01666	0.03333	0.03333	0.05000	0
grad	0	0	0	0.90740	0	0.02777	0.03703	0.02777
planina	0.01724	0.02873	0	0.00574	0.89080	0.03448	0.01149	0.01149
proplanak	0.03809	0.03809	0.05714	0	0.05238	0.80000	0.00952	0.00476
ulica	0	0	0.02173	0.032608	0	0.01086	0.89130	0.04347
zgrada	0	0.00641	0.01282	0.01923	0.01282	0	0.05128	0.89743

Da bismo dodatno stekli uvid u tačnost klasifikacije po klasama i da bismo detektovali klase za koje se najčešće javljaju greške pri klasifikaciji, u nastavku prikazujemo matricu konfuzije nad kolekcijom slika „8-scenes“ (Tabela 11). Od najvećeg značaja prilikom analize

matrice konfuzije su vrednosti na glavnoj dijagonali, koje pokazuju tačnost klasifikacije za svaku od klasa ponaosob. Na osnovu rezultata prikazane konfuzione matrice, može se zaključiti da najniže performanse prilikom klasifikacije pokazuju klase „autoput“ (81.666%) i „proplanak“ (80%). Ovde se otvara prostor za dodatnu analizu metoda za klasifikaciju, u pogledu povećanja tačnosti celog sistema.

6.3.2.3 Vremenske performanse

Konačni eksperimenti su obavljani kroz merenje i upoređivanje vremenskih performansi predloženog H-ELM-E postupka. Analizirana je MATLAB implementacija na procesoru Intel Core i7 3.2GHz. Prvo je obavljeno merenje vremena treniranja nad podskupom od 1680 slika, i vreme testiranja nad 420 slika iz „Landuse21“ kolekcije, a zatim je obavljeno treniranje nad 8000 slika i testiranje nad 2000 slika iz kolekcije „AID“. Tabela 12 i Tabela 13 prikazuju rezultate vremenskih performansi. Na osnovu rezultata se može zaključiti da H-ELM-E pokazuje visoku efikasnost izračunavanja, uz odgovarajuću stabilnost u pogledu variranja parametara mreže, sve dok je broj neurona dovoljno veliki. Generalno, H-ELM-E ne zahteva računski složene operacije za pronalaženje optimalnih parametara prilikom treniranja, i efikasno prilagođava porastu broju primera trening skupa [105].

Tabela 12 Vremenske performanse H-ELM-E metode u zavisnosti od konfiguracije mreže (na kolekciji slika „Landuse21“ sa 1680+420 slika).

Klasifikator	Vreme treniranja (s)	Vreme testiranja (s)
H-ELM-E (L=1000, K=10)	5.6	0.4
H-ELM-E (L=2000, K=10)	17.3	0.9
H-ELM-E (L=2000, K=20)	36.2	1.68

Tabela 13 Vremenske performanse H-ELM-E metode u zavisnosti od konfiguracije mreže (na kolekciji slika „AID“ sa 8000+2000 slika).

Klasifikator	Vreme treniranja (s)	Vreme testiranja (s)
H-ELM-E (L=1000, K=10)	23.74	2.25
H-ELM-E (L=2000, K=10)	69.86	4.43
H-ELM-E (L=2000, K=20)	132.48	8.65

Dodatno unapređenje vremenskih performansi H-ELM-E metode moguće je postići primenom modernih tehnika paralelizacije na CPU ili GPU. Paralelizaciju H-ELM-E je moguće vema efikasno i relativno jednostavno realizovati, jer se koriste ELM-E (tj. ELM) instance koje su potpuno iste strukture [70], [104]. Ovakve paralelizovane implementacije na modernim GPU mogu redukovati vreme izvršenja i za nekoliko redova veličine u odnosu na prezentovane rezultate prototipa algoritama implementiranih u Matlab-u.

7 ZAKLJUČAK

Prilikom izrade ove disertacije, sproveden je veliki broj eksperimenata nad pet različitih kolekcija slika, sa ciljem izvođenja zaključaka potrebnih za dalje unapređenje predloženih algoritama. Prilikom fuzije deskriptora slike, razmatrana su četiri različita deskriptora tekture i boje, koja su u literaturi pokazala veoma dobre rezultate. Analizirana je tačnost klasifikacije prilikom njihovog nezavisnog korišćenja, i izvedeni su zaključci koji su iskorišćeni za predlog novog algoritma za ranu fuziju deskriptora. Rezultati eksperimenata su pokazali da rana fuzija informacija o boji i teksturi, konzistentno unapređuje rezultate klasifikacije uz određeno povećanje dimenzija deskriptora. Takođe je zaključeno da više-rezolucijski pristup prilikom izračunavanja deskriptora ima veliki uticaj na uspešnost deskriptora.

Da bi se prethodno izdvojeni deskriptori slike praktično primenili za klasifikaciju slika, treba obezbediti algoritme mašinskog učenja koji postižu visoku tačnost i stabilnost rezultata uz brz postupak treniranja i testiranja. Iako veštačke neuronske mreže pokazuju dominantne rezultate u pogledu tačnosti klasifikacije, algoritmi za njihovo treniranje su veoma vremenski zahtevni. Zato je u ovoj disertaciji fokus istraživanja bio na relativno novom algoritmu za treniranje jednoslojnih neuronskih mreža, tzv. metodi ekstremno brzog učenja – ELM. Ova metoda pokazuje određene nedostatke u pogledu stabilnosti i stepena tačnosti klasifikacije, koji su posledica nasumičnog izbora ulaznih težina i pomeraja. Zato su razmatrane metode za povećanje stabilnosti i tačnost rezultata ELM klasifikatora, kroz kombinovanje (fuziju) većeg broja ELM mreža. Posebno su razmatrane tehnike kojima se rezultati individualnih ELM klasifikatora kombinuju u jedinstveni rezultat klasifikacije, i predložena je adekvatna tehnika za primenu pravila sumiranja prilikom kombinovanja rezultata ELM klasifikatora. Eksperimentalno je pokazano da se na ovaj način može povećati tačnost za 3-6% u odnosu na druge često korišćene tehnike, bez dodatne računске kompleksnosti. Prilikom ovih eksperimenata, razmatran je uticaj vrednosti parametara prilikom kombinovanja ELM klasifikatora na tačnost klasifikacije slika u boji. Ustanovljeno je da već prilikom fuzije relativno malog broja ELM mreža u ansamblu ($K=5$), moguće postići porast tačnosti klasifikacije i do 10%. Rezultati ove grupe eksperimenata, koji postižu tačnost od 86.52% uz varijansu od $\pm 1.29\%$ nad kolekcijom “Landuse21”, pokazuju da ELM ansamblu klasifikatora

uz adekvatni deskriptor slike, postižu veoma visoku tačnost klasifikacije uz veliku brzinu treniranja.

Izvedeni zaključci su iskorišćeni za predlog najznačajnijeg doprinosa ove disertacije – novu hijerarhijsku metodu za kombinovanje više komplementarnih deskriptora kroz primenu kasne fuzije ELM klasifikatora. Predložena metoda, označena kao H-ELM-E, obezbedila je da deskriptori koji nose značajniju informaciju za određenu sliku dobijaju veći značaj u konačnom rezultatu. Na ovaj način je moguće naučiti optimalnu kombinaciju deskriptora za svaku od kategorija slika. Rezultati eksperimenata sa H-ELM-E metodom, koji postižu tačnost klasifikacije od 91.45% uz varijansu od $\pm 1.25\%$ nad kolekcijom “Landuse21”, demonstrirali su izuzetno visoku tačnost i stabilnost predložene metode. Rezultati eksperimenata i nad ostalim korišćenim kolekcijama slika potvrdili su visok stepen tačnosti H-ELM-E metode. Merenjem vremenskih performansi je pokazano da vreme treniranja H-ELM-E mreže nad skupom od 1680 slika iznosi samo 5.6 sekundi, što predstavlja značajno unapređenje u odnosu na standardni algoritam propagacije unazad prilikom treniranja neuronskih mreža.

7.1 Buduće istraživanje

Postoji nekoliko smerova za buduće istraživanje u ovoj oblasti. Prvi je uključivanje informacija iz dela spektra koji nije vidljiv ljudskom oku, prilikom reprezentacije slike. U ovoj disertaciji su razmatrane isključivo RGB slike snimane u vidljivom delu spektra koji obuhvata opseg od približno 350nm do 750nm. Međutim, za reprezentaciju slike je moguće uključiti i informacije iz spektralnih opsega koji nisu vidljivi ljudskom oku, kao što je NIR deo spektra (engl. Near Infra Red - NIR) [39]. Ovaj deo spektra, u opsegu od približno 750nm do 1100nm, može se snimati većinom standardnih CMOS senzora. Iako je korišćenje ovakvih multispektralnih slika uobičajeno prilikom snimanja iz vazduha, metode za klasifikaciju ovakvih snimaka još uvek su retke u literaturi. Ovde se otvara mogućnost za proširenje i primenu prethodno opisanih metoda za klasifikaciju NIR slika. Arhitektura opisanih metoda je dovoljno opšta da omogućava relativno jednostavno proširenje.

Drugi smer za buduće istraživanje odnosi se na primenu konvolucione neuronskih mreža koje pokazuju superiorne rezultate u slučaju kada postoji dovoljno velika količina trening podataka [9–11]. Ovakve mreže imaju nedostatak da je treniranje izuzetno zahtevno u računskom i vremenskom pogledu. Ovde se otvara prostor za integraciju sa ELM klasifikatorima, koji su detaljno opisani u ovoj disertaciji, i koji imaju osobinu ekstremno brzog treniranja [108]. Takođe, konvolucione neuronske mreže zahtevaju treniranje nad velikim

kolekcijama slika koje sadrže milione slika klasifikovanih u hiljadama klasa. Međutim, za potrebe klasifikacije slika iz specifičnog domena primene (npr. snimci terena iz vazduha, medicinski snimci, itd.), ne postoje dovoljno velike kolekcije označenih slika, pri čemu njihovo formiranje može biti veoma zahtevno i skupo. Zato treba istražiti načine prilagođavanja modela koji se trenirani nad velikim kolekcijama opštih slika, kroz dodatno treniranje nad slikama iz specifičnog domena [53].

Količina podataka koja se koristi prilikom automatske klasifikacije slika je ogromna. Kako bi se implementirali praktično upotrebljivi sistemi za klasifikaciju slika, neophodno je pažljivo razmotriti računsku složenost algoritama, kao i mogućnosti za njihovu efikasnu implementaciju na specijalizovanim grafičkim procesorima (GPU). Ovakva vrsta optimizovanih implementacija, otvara mogućnost za buduće istraživanje u ovom smeru.

LITERATURA

- [1] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.
- [2] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. Springer-Verlag London, 2011.
- [3] D. Forsyth and J. Ponce, *Computer vision: a modern approach*. Upper Saddle River, NJ; London: Prentice Hall, 2011.
- [4] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [5] S. Cvetkovic, S. V. Nikolic, and S. Ilic, "Effective combining of color and texture descriptors for indoor-outdoor image classification," *Facta Universitatis Series: Electronics and Energetics*, vol. 27, no. 3, pp. 399–410, 2014.
- [6] S. Cvetkovic and S. V. Nikolic, "Merged MPEG-7 visual descriptors for image classification," in *2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS)*, 2011, vol. 1, pp. 345–348.
- [7] S. Cvetkovic, M. Jelenkovic, and S. V. Nikolic, "Video summarization using color features and efficient adaptive threshold technique," *Przeglad Elektrotechniczny (Review on Electrical Engineering)*, vol. 89, no. 2a, pp. 247–250, 2013.
- [8] D. Djordjevic, S. Cvetkovic, and S. V. Nikolic, "An accurate method for 3D object reconstruction from unordered sparse views," *Signal, Image and Video Processing*, vol. 11, no. 6, pp. 1147–1154, Sep. 2017.
- [9] S. Cvetkovic, M. Stojanovic, S. V. Nikolic, and G. Stancic, "Kernel based Extreme Learning Machines for Image Classification," in *Proceedings of International Scientific Conference on Information, Communication and Energy Systems and Technologies - ICEST 2016, Ohrid, Macedonia*, 2016, pp. 167–170.
- [10] S. Cvetkovic, M. Stojanovic, S. V. Nikolic, and G. Stancic, "Extreme Learning Machines for Real-Time Image Classification," in *Proceedings of International Scientific Conference on Information, Communication and Energy Systems and Technologies - ICEST 2015, Sofia, Bulgaria*, 2015, pp. 62–65.

- [11] S. Cvetkovic, M. B. Stojanovic, and S. V. Nikolic, "Multi-channel descriptors and ensemble of Extreme Learning Machines for classification of remote sensing images," *Signal Processing: Image Communication*, vol. 39, Part A, pp. 111–120, 2015.
- [12] S. Cvetkovic, B. Rajkovic, and S. V. Nikolic, "Real-time image classification using LBP and ensembles of ELM," *Scientific Publications of the State University of Novi Pazar Series A: Applied Mathematics, Informatics and Mechanics*, vol. 8, no. 1, pp. 101–109, 2016.
- [13] S. Cvetkovic, M. B. Stojanovic, and S. V. Nikolic, "Hierarchical ELM ensembles for visual descriptor fusion," *Information Fusion*, vol. 41, pp. 16–24, 2018.
- [14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.
- [15] M. Popovic, *Digitalna obrada slike*. Beograd: Akademska misao, 2006.
- [16] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, vol. 2, pp. 2169–2178.
- [17] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 01*, 2005, pp. 886–893.
- [18] A. Bosch, A. Zisserman, and X. Munoz, "Image Classification using Random Forests and Ferns," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.
- [19] C. Harris and M. Stephens, "A combined corner and edge detector.," in *Alvey vision conference*, 1988, vol. 15, no. 50, pp. 10–5244.
- [20] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [21] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 2, pp. 1150–1157 vol.2.
- [22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.
- [23] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

- [24] L. Zhang, Z. Zhou, and H. Li, “Binary Gabor Pattern: An efficient and robust descriptor for texture classification,” in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 81–84.
- [25] J. Sivic and A. Zisserman, “Video Google: a text retrieval approach to object matching in videos,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 1470–1477 vol.2.
- [26] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *2006 IEEE computer society conference on Computer vision and pattern recognition*, 2006, vol. 2, pp. 2161–2168.
- [27] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1794–1801.
- [28] F. Perronnin, J. Sanchez, and T. Mensink, “Improving the Fisher Kernel for Large-Scale Image Classification,” in *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 143–156.
- [29] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, “Image classification using super-vector coding of local image descriptors,” in *European conference on computer vision*, 2010, pp. 141–154.
- [30] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained Linear Coding for image classification,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.
- [31] Y. Yang and S. Newsam, “Bag-of-visual-words and Spatial Extensions for Land-use Classification,” in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 270–279.
- [32] G. S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, “AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.
- [33] A. Oliva and A. Torralba, “Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, May 2001.

- [34] W. Kim, J. Park, and C. Kim, “A Novel Method for Efficient Indoor—Outdoor Image Classification,” *Journal of Signal Processing Systems*, vol. 61, no. 3, pp. 251–258, Dec. 2010.
- [35] L. Gupta, V. Pathangay, A. Patra, A. Dyana, and S. Das, “Indoor versus Outdoor Scene Classification Using Probabilistic Neural Network,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 094298, 2006.
- [36] Y. Zhao, “Theories and Applications of LBP: A Survey,” in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, vol. 6839, D.-S. Huang, Y. Gan, P. Gupta, and M. M. Gromiha, Eds. Springer Berlin Heidelberg, 2012, pp. 112–120.
- [37] J. Wu and J. M. Rehg, “CENTRIST: A Visual Descriptor for Scene Categorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.
- [38] Y. Yang and S. Newsam, “Spatial pyramid co-occurrence for image classification,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 1465–1472.
- [39] M. Brown and S. Susstrunk, “Multi-spectral SIFT for scene category recognition,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 177–184.
- [40] V. Risojevic and Z. Babic, “Fusion of Global and Local Descriptors for Remote Sensing Image Classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 4, pp. 836–840, Jul. 2013.
- [41] Y. Xiao, J. Wu, and J. Yuan, “mCENTRIST: A Multi-Channel Feature Generation Mechanism for Scene Categorization,” *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 823–836, Feb. 2014.
- [42] A. Avramovic and V. Risojevic, “Block-based semantic classification of high-resolution multispectral aerial images,” *Signal, Image and Video Processing*, vol. 10, no. 1, pp. 75–84, Jan. 2016.
- [43] R. Arandjelovic and A. Zisserman, “All about VLAD,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.
- [44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing*

- Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [46] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, “BRIEF: Computing a Local Binary Descriptor Very Fast,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, Jul. 2012.
- [47] A. Garcia Seco de Herrera, H. Muller, and S. Bromuri, “Overview of the ImageCLEF 2015 medical classification task,” in *Working Notes of CLEF 2015 (Cross Language Evaluation Forum)*, 2015.
- [48] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, “Multimodal fusion for multimedia analysis: a survey,” *Multimedia Systems*, vol. 16, no. 6, pp. 345–379, 2010.
- [49] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *CoRR*, vol. abs/1409.1556, 2014.
- [50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [52] I. Sevo and A. Avramovic, “Convolutional Neural Network Based Automatic Object Detection on Aerial Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 5, pp. 740–744, May 2016.
- [53] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, “Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery,” *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015.
- [54] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [55] D. K. Park, Y. S. Jeon, and C. S. Won, “Efficient Use of Local Edge Histogram Descriptor,” in *Proceedings of the 2000 ACM Workshops on Multimedia*, 2000, pp. 51–54.

- [56] T. Ojala, M. Pietikainen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [57] T. Ahonen, A. Hadid, and M. Pietikainen, “Face Description with Local Binary Patterns: Application to Face Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
- [58] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters.,” *Journal of the Optical Society of America A*, vol. 2, no. 7, pp. 1160–1169, 1985.
- [59] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, “Color and texture descriptors,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 703–715, Jun. 2001.
- [60] H. Eidenberger, “Statistical analysis of content-based MPEG-7 descriptors for image retrieval,” *Multimedia Systems*, vol. 10, no. 2, pp. 84–97, 2004.
- [61] G. K. Wallace, “The JPEG still picture compression standard,” *Communications of the ACM*, pp. 30–44, 1991.
- [62] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1582–1596, 2010.
- [63] F. S. Khan, R. M. Anwer, J. van de Weijer, M. Felsberg, and J. Laaksonen, “Compact color-texture description for texture classification,” *Pattern Recognition Letters*, vol. 51, no. 0, pp. 16–22, 2015.
- [64] F. S. Khan, J. van de Weijer, S. Ali, and M. Felsberg, “Evaluating the Impact of Color on Texture Recognition,” in *Computer Analysis of Images and Patterns: 15th International Conference, CAIP 2013, York, UK, August 27-29, 2013, Proceedings, Part I*, R. Wilson, E. Hancock, A. Bors, and W. Smith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 154–162.
- [65] P. Klein, *Coding The Matrix: Linear Algebra Through Computer Science Applications*. Newtonian Press, 2013.
- [66] G. Strang, *Introduction to Linear Algebra*, Fifth Ed. Wellesley-Cambridge Press, 2016.
- [67] Y. Freund and R. E. Schapire, “A Short Introduction to Boosting,” *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

- [68] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [69] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain," *Psychological Review*, pp. 65–386, 1958.
- [70] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, 2015.
- [71] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [72] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
- [73] J. Cao and L. Xiong, "Protein Sequence Classification with Improved Extreme Learning Machine Algorithms," *BioMed Research International*, pp. 1–12, 2014.
- [74] J. Tang, C. Deng, and G.-B. Huang, "Extreme Learning Machine for Multilayer Perceptron," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.
- [75] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An Efficient Method for Traffic Sign Recognition Based on Extreme Learning Machine," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 920–933, Apr. 2017.
- [76] Y. Kaya, L. Kayci, R. Tekin, and O. F. Ertugrul, "Evaluation of texture features for automatic detecting butterfly species using extreme learning machine," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 26, no. 2, pp. 267–281, 2014.
- [77] P. Du, A. Samata, P. Gamba, and X. Xie, "Polarimetric SAR image classification by Boosted Multiple-Kernel Extreme Learning Machines with polarimetric and spatial features," *International Journal of Remote Sensing*, vol. 35, no. 23, pp. 7978–7990, 2014.
- [78] W. Li, C. Chen, H. Su, and Q. Du, "Local Binary Patterns and Extreme Learning Machine for Hyperspectral Imagery Classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
- [79] Z. Xie, K. Xu, W. Shan, L. Liu, Y. Xiong, and H. Huang, "Projective Feature Learning for 3D Shapes with Multi-View Depth Images," *Computer Graphics Forum*, vol. 34, no. 7, pp. 1–11, 2015.

- [80] J. Cao, C. Tao, and F. Jiayuan, “Landmark recognition with compact BoW histogram and ensemble ELM,” *Multimedia Tools and Applications*, pp. 1–19, 2015.
- [81] J. Cao, T. Chen, and J. Fan, “Fast online learning algorithm for landmark recognition based on BoW framework,” in *2014 IEEE 9th Conference on Industrial Electronics and Applications (ICIEA)*, 2014, pp. 1163–1168.
- [82] G.-B. Huang, L. Chen, and C.-K. Siew, “Universal Approximation Using Incremental Constructive Feedforward Networks with Random Hidden Nodes,” *Trans. Neur. Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.
- [83] G.-B. Huang, “What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt’s Dream and John von Neumann’s Puzzle,” *Cognitive Computation*, vol. 7, no. 3, pp. 263–278, 2015.
- [84] C. R. Rao and S. K. Mitra, “Generalized inverse of a matrix and its applications,” in *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*, 1972, pp. 601–620.
- [85] X. Tian, L. Jiao, X. Liu, and X. Zhang, “Feature integration of {EODH} and Color-SIFT: Application to image retrieval based on codebook,” *Signal Processing: Image Communication*, vol. 29, no. 4, pp. 530–545, 2014.
- [86] H. Ghassemian, “A review of remote sensing image fusion methods,” *Information Fusion*, vol. 32, Part A, pp. 75–89, 2016.
- [87] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, “Early Versus Late Fusion in Semantic Video Analysis,” in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 2005, pp. 399–402.
- [88] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
- [89] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Chapman & Hall/CRC, 2012.
- [90] L. Rokach, “Ensemble-based classifiers,” *Artificial Intelligence Review*, vol. 33, no. 1, pp. 1–39, Feb. 2010.
- [91] A. Jain, K. Nandakumar, and A. Ross, “Score normalization in multimodal biometric systems,” *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.
- [92] P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, “Neural network ensembles: evaluation of aggregation algorithms,” *Artificial Intelligence*, vol. 163, no. 2, pp. 139–162, 2005.

- [93] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar. 1998.
- [94] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [95] L. Breiman, "Stacked regressions," *Machine Learning*, vol. 24, no. 1, pp. 49–64, Jul. 1996.
- [96] G. Sigletos, G. Paliouras, C. D. Spyropoulos, and M. Hatzopoulos, "Combining Information Extraction Systems Using Voting and Stacked Generalization," *Journal of Machine Learning Research*, vol. 6, pp. 1751–1782, Dec. 2005.
- [97] S. Prabhakar and A. K. Jain, "Decision-level fusion in fingerprint verification," *Pattern Recognition*, vol. 35, no. 4, pp. 861–874, 2002.
- [98] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [99] M. Wozniak, M. Grana, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.
- [100] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.
- [101] J. Cao, Z. Lin, G.-B. Huang, and N. Liu, "Voting based extreme learning machine," *Information Sciences*, vol. 185, no. 1, pp. 66–77, 2012.
- [102] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, no. 13–15, pp. 3391–3395, 2009.
- [103] X.-L. Wang, Y.-Y. Chen, H. Zhao, and B.-L. Lu, "Parallelized extreme learning machine ensemble based on min-max modular network," *Neurocomputing*, vol. 128, no. 0, pp. 31–41, 2014.
- [104] M. V. Heeswijk, Y. Miche, E. Oja, and A. Lendasse, "GPU-accelerated and parallelized {ELM} ensembles for large-scale regression," *Neurocomputing*, vol. 74, no. 16, pp. 2430–2437, 2011.
- [105] A. Akusok, K. M. Bjork, Y. Miche, and A. Lendasse, "High-Performance Extreme Learning Machines: A Complete Toolbox for Big Data Applications," *IEEE Access*, vol. 3, pp. 1011–1025, 2015.
- [106] J. Ren, X. Jiang, and J. Yuan, "Learning LBP structure by maximizing the conditional mutual information," *Pattern Recognition*, vol. 48, no. 10, pp. 3180–3190, 2015.

- [107] S. H. Lee, J. Y. Choi, Y. M. Ro, and K. N. Plataniotis, "Local Color Vector Binary Patterns From Multichannel Face Images for Face Recognition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2347–2353, Apr. 2012.
- [108] G. B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local Receptive Fields Based Extreme Learning Machine," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 18–29, May 2015.

SPISAK SKRAĆENICA

ANN – Artificial Neural Network
BGP – Binary Gabor Patterns
BOF – Bag of Features
BOW – Bag of Words
CLD – Color Layout Descriptors
CNN – Convolutional Neural Network
EHD – Edge Histogram Descriptors
ELM – Extreme Learning Machines
ELM-E – ELM Ensembles
GPU – Graphics Processing Unit
ICA – Independent Component Analysis
H-ELM-E – Hierarchical ELM-E
k-NN – k Nearest Neighbours
LBP – Local Binary Patterns
LMS – Least Mean Squares
MLP – Multilayer Perceptron
MPEG – Motion Pictures Experts Group
NIR – Near Infrared
PCA – Principal Component Analysis
RGB – Red Green Blue
SIFT – Scale-Invariant Feature Transform
SVM – Support Vector Machines
SVM-RBF – SVM Radial Basis Function

SPISAK SLIKA

Slika 1 Ilustracija glavnih izazova i problema koje treba rešiti prilikom automatske klasifikacije slika	14
Slika 2 Primeri dominantnih vizuelnih obeležja (deskriptora) na slikama snimljenim iz vazduh	20
Slika 3 Primeri slika za svaku od 21 klase iz kolekcije „Landuse21” [31].....	28
Slika 4 Primeri po 3 slike za svaku od 30 kategorija iz kolekcije „AID“	29
Slika 5 Primeri slika za svaku od 8 klasa iz kolekcije „8-scenes” [33].	30
Slika 6 Primeri slika: a) IITM-SCID2 klasa unutrašnjih slika, b) IITM-SCID2 klasa slika snimljenih spolja, c) COREL-INOUT klasa unutrašnjih slika, d) COREL-INOUT klasa slika snimljenih spolja	31
Slika 7 Prikaz pet tipova ivica slike za izračunavanje EHD uz odgovarajuće maske filtara za detekciju [54]: a) h_0 - vertikalna 90° , b) h_1 - horizontalna 0° , c) h_2 - dijagonalna 45° , d) h_3 - dijagonalna 135° , e) h_4 - neorijentisana.....	34
Slika 8 Primer tri slike sa različitim rasporedom ivica, koje imaju isti histogram ivica slike - EHD.....	35
Slika 9 Primer izračunatog histograma ivica slike – EHD.....	36
Slika 10 Primer izračunavanja lokalnog LBP deskriptora za centralni piksel sa intenzitetom $f_0 = 217$ i lokalnim susedstvom dimenzija 3×3	37
Slika 11 Primeri različito definisanih okolina za izračunavanje LBP [57]. Ukoliko se sused nalazi na granici više piksela, onda se vrši bilinearna interpolacija graničnih piksela.	37
Slika 12 Primer 2D Gaborovog filtra: (a) parni tj. realni deo, (b) neparni tj. imaginarni deo.	39
Slika 13 Postupak izračunavanja SIFT deskriptora na jednom lokalnom regionu.	41
Slika 14 Postupak izračunavanja CLD.....	42
Slika 15 Postupak nadgledanog mašinskog učenja.	46
Slika 16 Aproksimacija trening skupa podataka različitim funkcijama: a) linearna funkcija, b) kvadratna funkcija, c) polinom 5-og stepena.	52
Slika 17 Oblik funkcije greške	55
Slika 18 Model veštačkog neurona sa tri ulaza: x_1, x_2, x_3 . Težinski faktori su označeni sa w_1, w_2, w_3 . U opštem slučaju, broj ulaza može biti proizvoljan.....	59

Slika 19 Aktivacione funkcije veštačkog neurona a) Sigmoidna (logistička) funkcija, b) Tangens hiperbolički.	60
Slika 20 Primer veštačke neuronske mreže sa jednim ulaznim slojem, jednim skrivenim slojem i jednim izlazom.	61
Slika 21 Označavanje težinskih matrica W_{in} , W_{out} i izlaznih vektora z_{in} , z_{out} u ELM neuronskoj mreži.	67
Slika 22 Postupak rane fuzije deskriptora	71
Slika 23 Predloženi postupak izračunavanja višekanalnog BGP deskriptora – MCBGP, kroz fuziju deskriptora teksture i boje [11].	75
Slika 24 Ilustracija postupka kasne fuzije	81
Slika 25 Primer kasne fuzije kada se koristi jedinstven deskriptor ulazne slike.....	82
Slika 26 Ilustracija postupka kombinovanja (agregacije) rezultata klasifikatora prilikom postupka kasne fuzije.	83
Slika 27 Uticaj broja neurona skrivenog sloja (L) i broja ELM klasifikatora u ansamblu (k), na tačnost klasifikacije slika iz kolekcije Landuse21. Svi individualni ELM klasifikatori unutar ELM-E ansambla imaju istu arhitekturu.	89
Slika 28 Uticaj primenjene tehnike agregacije rezultata unutar ansambla, na tačnost klasifikacije.	90
Slika 29 Vreme treniranja ELM-E za celokupan trening skup od 1680 slika. Parametar K označava broj individualnih ELM klasifikatora u ansamblu ELM-E.	93
Slika 30 Hijerarhijski postupak kasne fuzije (H-ELM-E), primenom ELM ansambla.....	96
Slika 31 Tačnost klasifikacije predložene H-ELM-E metode nad kolekcijom slika „Landuse21“, u zavisnosti od broj neurona (L) skrivenog sloja svakog ELM, kao i broja ELM mreža u ansamblu (K).	99
Slika 32 Tačnost klasifikacije predložene H-ELM-E metode nad kolekcijom slika „8-scenes“, u zavisnosti od broj neurona (L) skrivenog sloja svakog ELM, kao i broja ELM mreža u ansamblu (K).	99
Slika 33 Tačnost klasifikacije predložene H-ELM-E metode nad kolekcijom slika „AID“, u zavisnosti od broj neurona (L) skrivenog sloja svakog ELM, kao i broja ELM mreža u ansamblu	100

SPISAK TABELA

Tabela 1 Primer matrice konfuzije za klasifikaciju u 3 klase.	26
Tabela 2 Tačnost klasifikacije pod uticajem različitih tehnika normalizacije deskriptora [5].	74
Tabela 3 Uticaj parametara koji definišu deskriptor slike MCBGP, na tačnost rezultata klasifikacije..	78
Tabela 4 Matrica konfuzije za kolekciju slika “Landuse21”	91
Tabela 5 Analiza uticaja veličine trening skupa na rezultate klasifikacije, primenom različitih metoda klasifikacije. Prikazana je tačnost klasifikacije uz standardnu devijaciju.	92
Tabela 6 Poređenje vremena treniranja (1680 slika) i vremena testiranja (420 slika), primenom različitih klasifikacionih metoda.	93
Tabela 7 Tačnost klasifikacije H-ELM-E u odnosu na korišćenje nezavisnih deskriptora i masovno korišćenu tehniku rane fuzije. Rezultati testiranja su prikazani za tri kolekcije slika “Landuse21“, “8-scenes” i “AID”.....	101
Tabela 8 Poređenje tačnosti predložene metode H-ELM-E u odnosu na aktuelne rezultate iz literature, u zavisnosti od broja trening slika (kolekcija slika „Landuse21“).	102
Tabela 9 Poređenje tačnosti predložene metode H-ELM-E u odnosu na aktuelne rezultate iz literature, u zavisnosti od broja trening slika (kolekcija slika „8-scenes“).	102
Tabela 10 Poređenje tačnosti predložene metode H-ELM-E u odnosu na aktuelne rezultate iz literature, u zavisnosti od broja trening slika (kolekcija slika „AID“).	103
Tabela 11 Primer matrice konfuzije za kolekciju slika “8-scenes”.....	103
Tabela 12 Vremenske performanse H-ELM-E metode u zavisnosti od konfiguracije mreže (na kolekciji slika „Landuse21“ sa 1680+420 slika).....	104
Tabela 13 Vremenske performanse H-ELM-E metode u zavisnosti od konfiguracije mreže (na kolekciji slika „AID“ sa 8000+2000 slika).	104

BIOGRAFIJA AUTORA

Stevica S. Cvetković je rođen 6. decembra 1981. godine u Zaječaru, Republika Srbija. Osnovnu školu je završio kao najbolji učenik generacije, a potom specijalizovano matematičko odeljenje Gimnazije "Svetozar Marković" u Nišu sa prosečnom ocenom 5 (pet). U toku školovanja uspešno je učestvovao na većem broju republičkih takmičenja iz matematike i fizike. Diplomirao je na Elektronskom fakultetu u Nišu, februara 2007. godine, sa prosečnom ocenom 9,40 (devet, 40/100). Diplomski rad iz predmeta Digitalna obrada slike, pod naslovom „Automatsko prepoznavanje otisaka prstiju“, odbranio je sa ocenom 10 (deset) pod mentorstvom prof. dr Saše V. Nikolića. Dobitnik je većeg broja stipendija i nagrada u toku školovanja, od kojih izdvajamo: stipendija kompanije "Philip Morris" (dodeljena 2005. i 2006. godine), stipendija Norveške ambasade (2004.), stipendija grada Niša za darovite učenike i studente (2003, 2004, 2006.), stipendija Ministarstva prosvete (2002.), drugo mesto na ICT takmičenju "Balkan Case Challenge for Serbia" u organizaciji WUS Austria (2001.), diploma Elektronskog fakulteta za uspeh u studiranju (2001.), nagrada na takmičenju „Matematički turnir gradova" u organizaciji Ruskog ministarstva nauke (1998), itd.

Profesionalnu karijeru započinje u kompaniji „Diyomisoft" iz Niša, gde je od aprila 2007. do aprila 2008. bio zaposlen kao inženjer na poslovima projektovanja i programiranja Medicinskog informacionog sistema. Od maja 2008. do decembra 2010. kandidat je bio angažovan kao inženjer razvoja softvera u okviru Inovacionog centra Elektronskog fakulteta u Nišu, na međunarodnom projektu saradnje sa „ABB Switzerland Ltd". Od decembra 2010. godine kandidat je zaposlen na Elektronskom fakultetu u Nišu kao saradnik u nastavi, a zatim kao asistent pri Katedri za elektroniku. Učestvovao je u pripremi i izvođenju računskih i laboratorijskih vežbi iz sledećih predmeta: Digitalna obrada slike, Multimedijalni sistemi, TV sistemi, Tehnike prenosa slike, Video sistemi i Obrada video signala.

U periodu od jula do septembra 2005, boravio je na Politehničkom Univerzitetu u Madridu (Španija), gde je obavio stručnu praksu u Laboratoriji za digitalnu obradu slike. U avgustu 2010. bio je na stručnom usavršavanju u kompaniji "ABB Switzerland Ltd", u Badenu (Švajcarska) u okviru grupe za Proizvodne informacione sisteme. U periodu od juna 2012. do juna 2016. imao je više istraživačkih boravaka na Tehničkom Univerzitetu Minhenu – TUM (Nemačka), u okviru Grupe za kompjuterski vid.

Kao stipendista Ministarstva nauke za doktorske studije, učestvovao je u realizaciji dva nacionalna projekta: „Primena termovizije, razvoj novih metoda ispitivanja i softvera za obradu

termovizijskih slika” (br. 6222) i „Inovacija, integracija i kolaboracija informacionih sistema zdravstvenih ustanova” (br. 13015). U toku 2012. godine inicirao je saradnju sa Grupom za kompjuterski vid pri Tehničkom Univerzitetu u Minhenu (Nemačka), na osnovu koje su realizovana dva međunarodna projekta finansirana od strane DAAD i Bayhost fondacija.

Istraživačka delatnost kandidata je fokusirana u oblastima Digitalne obrade slike i Kompjuterskog vida. Konkretno teme istraživanja obuhvataju automatsku klasifikaciju slika, detekciju objekata i pretraživanje velikih baza slika. Do sada je objavio 31 publikaciju, od čega 3 naučna rada u međunarodnim časopisima sa SCI liste (M21 i M22), 4 rada u ostalim međunarodnim časopisima, 9 radova na simpozijumima međunarodnog značaja (M33), 13 radova na skupovima nacionalnog značaja (M63) i 2 tehnička rešenja (M85).

IZJAVA O AUTORSTVU

Izjavljujem da je doktorska disertacija, pod naslovom

**Automatska klasifikacija slika zasnovana na fuziji deskriptora i nadgledanom
mašinskom učenju**

koja je odbranjena na Elektronskom fakultetu Univerziteta u Nišu:

- rezultat sopstvenog istraživačkog rada;
- da ovu disertaciju, ni u celini, niti u delovima, nisam prijavljivao/la na drugim fakultetima, niti univerzitetima;
- da nisam povredio/la autorska prava, niti zloupotrebio/la intelektualnu svojinu drugih lica.

Dozvoljavam da se objave moji lični podaci, koji su u vezi sa autorstvom i dobijanjem akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada, i to u katalogu Biblioteke, Digitalnom repozitorijumu Univerziteta u Nišu, kao i u publikacijama Univerziteta u Nišu.

U Nišu, _____

Potpis autora disertacije:

Stevica S. Cvetković

**IZJAVA O ISTOVETNOSTI ELEKTRONSKOG I ŠTAMPANOG OBLIKA
DOKTORSKE DISERTACIJE**

Naslov disertacije:

Automatska klasifikacija slika zasnovana na fuziji deskriptora i nadgledanom mašinskom učenju

Izjavljujem da je elektronski oblik moje doktorske disertacije, koju sam predao za unošenje u **Digitalni repozitorijum Univerziteta u Nišu**, istovetan štampanom obliku.

U Nišu, _____

Potpis autora disertacije:

Stevica S. Cvetković

IZJAVA O KORIŠĆENJU

Ovlašćujem Univerzitetsku biblioteku „Nikola Tesla“ da u Digitalni repozitorijum. Univerziteta u Nišu unese moju doktorsku disertaciju, pod naslovom:

Automatska klasifikacija slika zasnovana na fuziji deskriptora i nadgledanom mašinskom učenju

Disertaciju sa svim prilogima predao sam u elektronskom obliku, pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju, unetu u Digitalni repozitorijum Univerziteta u Nišu, mogu koristiti svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons), za koju sam se odlučio.

1. Autorstvo (**CC BY**)

2. Autorstvo – nekomercijalno (**CC BY-NC**)

3. Autorstvo – nekomercijalno – bez prerade (CC BY-NC-ND)

4. Autorstvo – nekomercijalno – deliti pod istim uslovima (**CC BY-NC-SA**)

5. Autorstvo – bez prerade (**CC BY-ND**)

6. Autorstvo – deliti pod istim uslovima (**CC BY-SA**)

U Nišu, _____

Potpis autora disertacije:

Stevica S. Cvetković