Univerzitet Crne Gore
Elektrotehnički fakultet

Tarik Avdović

POLU-KRHKI PRISTUP UMETANJA VODENOG ŽIGA U DIGITALNE SLIKE UPOTREBOM DUBOKOG UČENJA I EVALUACIJA VIŠE FUNKCIJA TROŠKA

– MASTER RAD –

Podgorica, 2025. godine

PODACI I INFORMACIJE O STUDENTU

Ime i prezime: **Tarik Avdović**
Datum i mjesto rođenja: **07. oktobar 2000. godine, Pljevlja**
Naziv završenog osnovnog studijskog programa i godina završetka studija:
Elektronika, telekomunikacije i računari, 2022. godine

INFORMACIJE O MASTER RADU

Naziv master studija: **Master akademske studije, odsjek Elektronika, telekomunikacije i računari, smjer Računari**
Naslov rada: **Polu-krhki pristup umetanja vodenog žiga u digitalne slike upotrebom dubokog učenja i evaluacija više funkcija troška**
Fakultet na kojem je rad odbranjen: **Elektrotehnički fakultet Podgorica**

UDK, OCJENA I ODBRANA MASTER RADA

Datum prijave master rada: **17. jun 2024. godine**
Datum sjednice Vijeća na kojoj je prihvaćena tema: **16. septembar 2024. godine**
Mentor: **Prof. dr Igor Đurović**

Komisija za ocjenu i odbranu rada:

1. **Prof. dr Igor Đurović, ETF Podgorica**
2. **Prof. dr Vesna Popović-Bugarin, ETF Podgorica**
3. **Doc. dr Miloš Brajović, ETF Podgorica**

Datum odbrane: **20. mart 2025. godine**

Ime i prezime autora: Tarik Avdović

Etička izjava

U skladu sa članom 22 Zakona o akademskom integritetu i članom 18 Pravila studiranja na master studijama, pod krivičnom i materijalnom odgovornošću, izjavljujem da je master rad pod naslovom:

„Polu-krhki pristup umetanja vodenog žiga u digitalne slike upotrebom dubokog učenja i evaluacija više funkcija troška"

moje originalno djelo.

Podnosilac izjave:

Tarik Avdović

Tarik Avdović

Podgorica, 29. januar 2025. godine

Sažetak

Digitalizacija i globalna dostupnost multimedijalnog sadržaja značajno su promijenili način na koji dijelimo i čuvamo informacije. Umetanje vodenog žiga podrazumijeva ugrađivanje informacija od značaja u digitalni sadržaj, čime se označava njegov vlasnik i potvrđuje autentičnost. Svrha ovog istraživanja je razvoj sistema zasnovanog na dubokom učenju koji implementira polu-krhki pristup umetanja vodenog žiga, ostvarujući kompromis između neprimjetnosti i robusnosti umetnutog vodenog žiga. Sistem se sastoji od mreže umetača i detektora. Postizanje optimalne konvergencije oba modela ostvaruje se kroz zajednički trening, koji uzima u obzir antagonističke ciljeve obje komponente. Kroz upotrebu konvolucionih neuralnih mreža, za postavljeni zadatak korišćeno je više funkcija troška, sa ciljem identifikacije one koja ostvaruje najbolje rezultate. Za procjenu neprimjetnosti umetanja koriste se PSNR i SSIM, dok se za ocjenu robusnosti koristi BER. Korišćenjem RMSE funkcije troška, sistem ostvaruje najbolje rezultate, s vrijednošću PSNR iznad 31 dB, dok SSIM dostiže 0.962. U pogledu robusnosti, vrijednosti BER-a u prisustvu smetnji u prenosnom kanalu variraju između 0.001 i 0.002, što je mjerljivo sa vodećim radovima u ovoj oblasti.

Ključne riječi: duboko učenje, vodeni žig, konvolucione neuralne mreže, funkcije troška

Abstract

Digitalization and the global availability of multimedia content have significantly transformed the way we share and store information. Watermarking involves embedding significant information into digital content thereby marking its owner and confirming its authenticity. The purpose of this research is to develop a deep learning-based system that implements a semi-fragile approach to watermark embedding. achieving a balance between the imperceptibility and robustness of the embedded watermark. The system consists of an embedder and detector network. Achieving optimal convergence of both models is realized through joint training, which takes into account the antagonistic objectives of both components. Through the use of convolutional neural networks, multiple loss functions were employed for the given task with the goal of identifying the one that delivers the best results. To evaluate the imperceptibility of embedding, PSNR and SSIM are used, while BER is used to assess robustness. By using the RMSE loss function the system achieves the best results, with a PSNR value exceeding 31 dB, while SSIM values reach 0.962. In terms of robustness, BER values in the presence of noise in the transmission channel range between 0.001 and 0.002, which is comparable to the leading works in this field.

Keywords: deep learning, watermark, convolutional neural networks, cost functions

Sadržaj

1 Uvod

Digitalna revolucija i ekspanzija Interneta omogućili su razvoj savremenih oblika komunikacije kakve ih danas poznajemo, olakšali skladištenje podataka, te pojednostavili prikupljanje, obradu i distribuciju informacija. Digitalna transformacija je donijela brojne promjene u društvu, ekonomiji i svakodnevnom životu. Ubrzo se uvidjelo da, pored onih pozitivnih, poput lakšeg pristupa informacijama, globalnoj povezanosti, otvaranju novih industrija i tržišta i generalnom poboljšanju kvaliteta života, postoje i negativne strane nastale usljed ovih promjena. Privatnost i sigurnost podataka korisnika su na udaru zlonamjernih pojedinaca ili organizacija sa ciljem krađe, izmjene i neovlašćene distribucije, što kao krajnji ishod ima povredu autorskih prava i narušavanje intelektualne svojine. Spoj dostupnosti multimedijalnih podataka i demokratizacija softvera za njihovu obradu, koji su sada lako savladivi i običnom korisniku, dodatno pospješuju mogućnost zloupotrebe i narušavanja autorskih prava.

Neovlašćeno dijeljenje i distribucija zaštićenih sadržaja poput filmova, slika, muzike, knjiga i softvera postali su raširena pojava svugdje u svijetu. Ove aktivnosti ne samo da štete ekonomiji kreativnih industrija, već i podstiču rastući problem u vezi sa nelegalnim sadržajem na mreži. Jasnog plana za borbu sa internet piraterijom na svjetskom nivou nema, već se regulative za suzbijanje ovih nelegalnih radnji donose na nivou država ili političkih i ekonomskih saveza. Na primjer, Evropska unija je uvela zakone kao što je Direktiva o autorskim pravima na jedinstvenom digitalnom tržištu (Direktiva (EU) 2019/790), koja ima za cilj da ojača zaštitu autorskih prava u digitalnom okruženju [1]. Nažalost, Crna Gora, zajedno sa ostalim zemljama Zapadnog Balkana, i dalje predstavlja slabo kontrolisano područje kada je riječ o internet pirateriji. Problem nije u nedostatku zakona koji regulišu ovo polje, koji su mahom usklađeni sa evropskim, već neefikasnost u njihovom sprovođenju i procesuiranju odgovornih, kao i u slabijoj platežnoj moći stanovnika ovog podneblja, što ni u kom slučaju ne opravdava bilo koju od ovih radnji. Nedavno je interesovanje šire javnosti za problem internet piraterije obnovila ileaglna distribucija filmskog ostvarenja „Toma", čija kopija je nekoliko dana nakon bioskopske premijere počela da kruži internetom [2].

Umetanje vodenog žiga (eng. *watermarking*) je postupak dodavanja skrivenih informacija u digitalnu sliku, audio ili video zapis radi zaštite autorskih prava i osiguravanja autentičnosti. Ovaj proces ima za cilj zaštitu autorskih prava, autentifikaciju sadržaja te odbranu od zlonamjernih napada. To je suštinski niz bitova koji nose određenu informaciju značajnu autoru digitalnog sadržaja. Sa vodenim žigovima se srećemo svakodnevno, a da toga nismo ni svjesni. Novčanica kojom plaćamo, e-knjiga koju čitamo ili pjesma koju slušamo, sadrže neku vrstu vodenog

žiga. U kontekstu karakteristika kao što su vidljivost, isticanje ili privlačenje pažnje, vodeni žigovi se dijele na upadljive i neprimjetne. Primjer su profesionalni fotografi ili novinske agencije koji svojim logom, najčešće u obliku slike, a to može biti i tekst, označavaju vlasništvo nad fotografijom. Vodeni žigovi, poput onih sa *Getty Images* ili *Shutterstock*-a, služe kao znak prepoznatljivosti i garancija kvaliteta, ističući ugled ovih platformi u industriji vizualnih sadržaja. Kada je riječ o diskretnim vodenim žigovima, zbog sofisticiranih metoda umetanja o kojima će biti riječi u nastavku rada, teže ih je ukloniti, a priroda diskretnosti garantuje manju šansu da će ih neko sa lošim namjerama primijetiti i pokušati ukloniti.

Proces umetanja vodenog žiga u digitalnu sliku (u nastavku će ravnopravno biti korišćena tuđica votermarking), nezavisno od toga da li će žig biti vidljiv ili ne, i njegovog ekstrahovanja na prijemnoj strani, nije jednostavan. Algoritam za umetanje žiga mora biti pažljivo osmišljen kako se ne bi ugrozio prethodno pomenuti princip nenarušavanja originalne informacije. Treba uzeti u obzir i to da signal prilikom prolaska kroz medijum, bilo namjerno ili ne, doživljava određeni stepen degradacije, što utiče i na vodeni žig. Robusniji vodeni žig bi se uspješnije nosio sa manipulacijama i velike su šanse da informacije koje nosi ne bi bile degradirane, međutim, to dolazi u sukob sa načelom očuvanja originalne informacije. Polu-krhki vodeni žig predstavlja upravo jedan od kompromisa kada je riječ o neprimjetnosti i robusnosti.

Fokus ovog rada je na digitalnoj slici, kao ključnom elementu savremene vizuelne komunikacije i raznih industrija, od medija i marketinga do nauke i umjetnosti. Iako su promjene u audio signalu uzrokovane umetanjem vodenog žiga lakše za primijetiti, jer je ljudsko uho izuzetno osjetljivo na promjene u zvuku, čak i na vrlo suptilne varijacije u frekvenciji, amplitudi i vremenskom trajanju, to nikako ne umanjuje izazov votermarkinga digitalne slike. Vizuelni sistem je takođe osjetljiv, ali manje nego slušni kada se radi o detekciji sitnih promjena. Ljudsko oko primjećuje promjene u boji, kontrastu i oštrini, ali je manje osjetljivo na sitne promjene na nivou pojedinačnih piksela ili neznatne varijacije u teksturi, što eksploatišu neke tehnike votermarkinga diskutovane u Poglavlju 3.

Votermarking ima dugu istoriju koja seže do perioda srednjeg vijeka. Prvobitno, vodeni žigovi su bili fizički tragovi na papiru, uvedeni u Italiji u 13. vijeku, kako bi označili porijeklo i kvalitet papira. Prvi vodeni žigovi su imali oblik prozirnih oznaka koje su bile vidljive kada se papir drži prema svijetlu, i služili su kao sredstvo zaštite od falsifikovanja. Dalje, tokom 18. i 19. vijeka, vodeni žigovi su postali standard na novčanicama, važnim dokumentima i državnim obveznicama, takođe sa svrhom očuvanja autentičnosti. U istom vremenskom periodu, vodeni žigovi su počeli da se pojavljuju i na poštanskim markama, što je filatelistima bilo od posebnog značaja, jer su mogli dodatno povećati vrijednost marke.

Krajem prethodnog i početkom sadašnjeg vijeka, koncept vodenog žiga se prilagodio tehnološkim naprecima, prešavši iz fizičke u digitalnu sferu. Vještačka inteligencija i njena zloupotreba kada je riječ o kreiranju visoko kvalitetnih multimedijalnih falsifikata - dipfejkova (eng. *deepfakes*), unijela je pometnju u gotovo sve sfere ljudske djelatnosti. Sa pojavom dipfejkova, provjera autentičnosti informacija postaje predmet sve većeg interesovanja, usljed velikog broja slika i snimaka generisanih upotrebom vještačke inteligencije. Širenje dezinformacija društvenim mrežama, diskreditacija javnih ličnosti i organizacija, političke manipulacije sa nepredvidivim posljedicama, dovode u opasnost onoga koji je meta takvih manipulacija, a samim tim i društvo u cjelini kreirajući atmosferu opšteg nepovjerenja i konfuzije u kojoj niko nije siguran u autentičnost onoga što gleda i sluša. Niska digitalna pismenost i lakovjernost, u nekim slučajevima i želja za potvrdom (eng. *confirmation bias*), neki su od razloga zašto dipfejkovi pronalaze plodno tlo na društvenim mrežama i sumnjivim medijskim portalima. Slučaj koji osvjetljava uticaj dipfejk tehnologija je lažni video snimak koji je uključivao predsjednika Ukrajine, Volodimira Zelenskog, tokom rata u Ukrajini 2022. godine, kako navodno poziva svoje vojnike da polože oružje [3]. Lažni video snimak je brzo postao viralan, izazivajući konfuziju i sumnju među Ukrajincima i međunarodnom zajednicom. Iako je snimak ubrzo razotkriven kao lažan, uspio je da izazove privremenu paniku. Ovaj incident ilustruje kako dipfejkovi mogu ozbiljno narušiti povjerenje u javne figure i destabilizovati društva u kriznim vremenima. Danas, kada terabajti podataka cirkulišu internetom na dnevnom nivou, votermarking sistemi kroz osiguranje autentičnosti i validnosti sadržaja imaju veću relevantnost nego ikada.

Rad je strukturno organizovan na sljedeći način. U Poglavlju 2 razmatrane su osnovne komponente svakog votermarking sistema, umetač i detektor, kao i karakteristike kojima se opisuje efikasnost i funkcionalnost sistema. Poglavlje 3 pruža detaljan pregled tehnika umetanja vodenog žiga, obuhvatajući tradicionalne metode u prostornom i frekvencijskom domenu, kao i savremene pristupe koji se oslanjaju na duboko učenje. Tema Poglavlja 4 su neuralne mreže, njihovi osnovni elementi, kao i pojmovi i tehnike od značaja za predloženi sistem. Poglavlje 5 nudi sveobuhvatan opis komponenti sistema, procesa njihovog zajedničkog treniranja, kao i različitih napada koji se koriste za simulaciju realnih uslova u prenosnom kanalu. Rezultati istraživanja, zajedno s poređenjem sa relevantnim radovima u ovoj oblasti, predstavljeni su u Poglavlju 6. Poglavlje 7 sadrži zaključak, kao i preporuke za dalje poboljšanje sistema.

2 Osnovni koncepti digitalnog votermarkinga

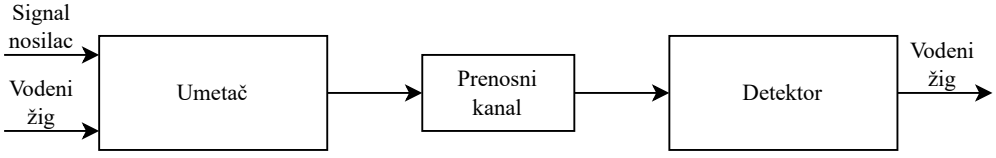
Bazičnu strukturu digitalnog votermarking sistema čine umetač (eng. *embedder*) i detektor (eng. *detector*), kao što je prikazano Slikom 1. Između njih se nalazi prenosni kanal kroz koji signal putuje i može biti podložan različitim vrstama izobličenja, šuma i gubitaka, kao i promjenama izazvanim zlonamjernim postupcima ili nestručnim rukovanjem. Performanse sistema, koje obuhvataju preciznost detekcije bitova vodenog žiga i neprimjetnost umetanja, ne bi smjele biti ugrožene u bilo kojem scenariju. Ovo zahtijeva pažljiv pristup u dizajnu sistema, uz implementaciju odgovarajućih kompromisnih rješenja kako bi se postigla ravnoteža između robusnosti i neprimjetnosti žiga. U nastavku će biti razmotrene glavne komponente sistema, umetač i detektor.

2.1 Umetač vodenog žiga

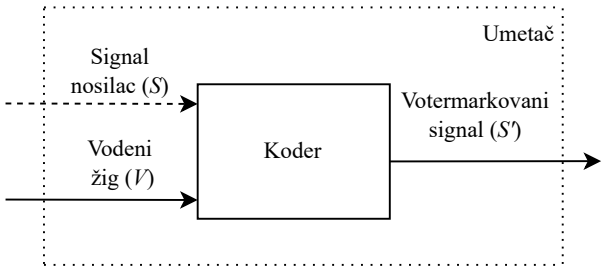
Umetač je odgovoran za umetanje vodenog žiga u signal nosilac, kao što su slike, video, ili audio podaci. Njega nerijetko sačinjava jedan element, koder. Vodeni žig, koji je izvorno sekvenca bitova, se prije umetanja u sliku kodira u oblik najpogodniji za operaciju umetanja. Signal nosilac takođe može biti transformisan u drugi domen u kojem će umetanje žiga biti izvedeno. Transformacija mora ispunjavati uslove linearnosti i inverzibilnosti, kako bi bio omogućen povratak originalnog signala nakon umetanja i spriječio gubitak informacija. Neka je E funkcija kodera, V označava vodeni žig, a S digitalnu sliku. Tada bi umetanje vodenog žiga u sliku moglo biti matematički predstavljeno kao [4]:

$$E(V, S) = S', \tag{1}$$

gdje S' predstavlja digitalnu sliku sa ugrađenim žigom. U određenim implementacijama, u cilju poboljšanja robusnosti i kvaliteta sistema uopšte, potrebno je vodeni žig prilagoditi specifičnostima signala nosioca, tj. omogućiti koderu vodenog žiga da „vidi" signal nosilac. Ovakvi umetači se nazivaju informisanim. Suprotno, ako ne postoji veza između signala nosioca i kodera vodenog žiga, za takve umetače kažemo da su neinformisani. Uzimajući u obzir prethodno rečeno, u primjeru sa Slike 2,



Slika 1: Opšta struktura digitalnog votermarking sistema



Slika 2: Blok šema umetača

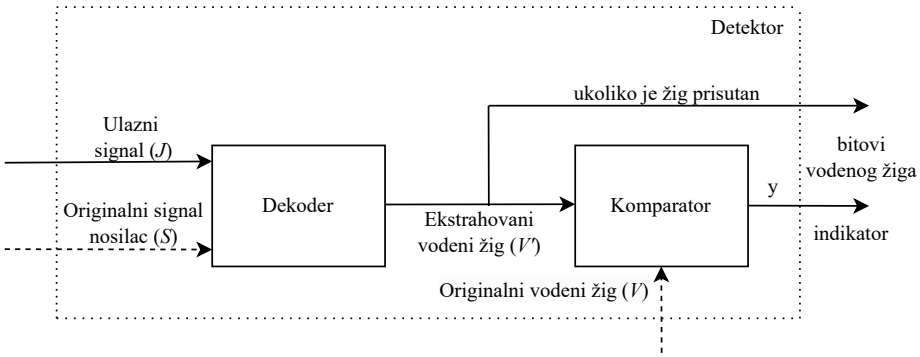
signal nosilac je predstavljen isprekidanom linijom, kao opcioni ulaz. U predloženoj realizaciji, umetač je informisan, jer koristi kodovane karakteristike slike (signala) kako bi odredio optimalnu reprezentaciju za umetanje vodenog žiga. Detaljnije o ovome će biti riječi u Poglavlju 5.1.

2.2 Detektor vodenog žiga

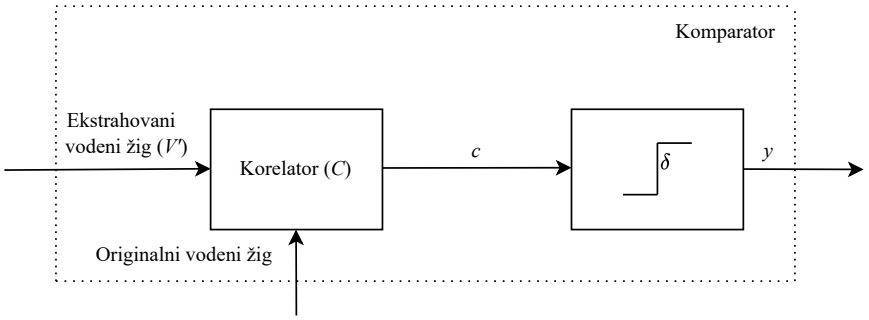
Detektor ima ulogu detekcije i izdvajanja vodenog žiga iz nosioca podataka. On se sastoji od dekodera i komparatora. Na ulaz dekodera dolaze ulazni signal, koji u ovom slučaju može predstavljati sliku sa vodenim žigom ili bez njega, ili sliku sa oštećenim vodenim žigom, u zavisnosti od stepena promjena u prenosnom kanalu, i originalni signal koji predstavlja digitalnu sliku sa ulaza u koder. Analogno koderu, dekoder koji kao dodatni ulaz sadrži originalni signal nosilac, naziva se informisanim. Dekoder koji ima samo jedan ulaz nazivamo neinformisanim. Kao i u slučaju umetača, opcioni ulazi detektora prikazani su isprekidanim linijama (Slika 3). Arhitektura predloženog detektora, koja će biti podrobnije diskutovana u Sekciji 5.2, svrstava se u grupu neinformisanih, jer na svom ulazu nema originalni signal. Slično relaciji (1), funkciju dekodera možemo matematički predstaviti kao:

$$D(J, S) = V', \tag{2}$$

gdje su J ulazni signal, S originalni signal nosilac, a V' ekstrahovani vodeni žig. Važno je napomenuti da J i S' predstavljaju dva različita signala, budući da se S', kojim je označen signal na izlazu iz umetača, može oštetiti ili izmijeniti u prenosnom kanalu. Dalje, na izlazu iz dekodera, ekstrahovani vodeni žig (ukoliko je prisutan) se vodi u kolo komparatora gdje se upoređuje sa originalnim žigom i donosi se odluka, što je prikazano Slikom 4. Ekstrahovani vodeni žig predstavlja takođe i jedan od izlaza detektora. Relacijom (3) prikazan je proces odlučivanja u komparatoru, gdje je $y = C_\delta(V, V')$ korelator, c predstavlja korelaciju između originalnog i vodenog žiga na ulazu u komparator, dok je δ prag odlučivanja.



Slika 3: Blok šema detektora



Slika 4: Komparator

$$C_\delta(V, V') = \begin{cases} 1, & \text{ako je } c \geq \delta \\ 0, & \text{ako je } c < \delta. \end{cases} \tag{3}$$

Korelator je sastavni dio komparatora koji računa sličnost između dva signala, dok je korelacija matematička mjera koja kvantifikuje tu sličnost. Ukoliko je vodeni žig, iz bilo kog razloga, pretrpio velika oštećenja u prenosnom kanalu, sličnost sa originalom će biti niska, što neće biti dovoljno da se „preskoči" postavljeni prag, te će logička nula značiti odsustvo žiga u signalu. Ako vodeni žig tokom prenosa nije značajno oštećen, ili bar u mjeri koja je predviđena prvobitnim dizajnom, stepen korelacije će biti veći od praga i na izlazu će se naći logička jedinica.

2.3 Karakteristike vodenog žiga

Votermarking sistemi se odlikuju velikim brojem karakteristika, od kojih je veliki broj pokriven u poznatoj knjizi iz ove oblasti [5]. Prevalentni uticaj ili veći značaj jedne karakteristike u odnosu na drugu definisan je namjenom votermarking sistema.

U nastavku će, radi konciznosti i značaja za ovaj master rad, biti predstavljene samo neke od njih.

2.3.1 Robusnost

Robusnost vodenog žiga određuje njegovu sposobnost da očuva integritet i prepoznatljivost čak i kada je podvrgnut različitim vrstama napada ili obrada. Sistemi koji insistiraju na robusnosti se bave zaštitom autorskih prava i intelektualne svojine, a razlog je upravo taj što vodeni žig treba da ostane očuvan uprkos različitim promjenama i manipulacijama kako bi se osiguralo da se informacije o vlasništvu i porijeklu sadržaja ne izgube ili ne budu lako uklonjene. U sistemima koji se bave provjerom autentičnosti sadržaja, poželjna je niska robusnost. Takvi žigovi se nazivaju krhkim (eng. *fragile*) i služe za detekciju i najmanjih izmjena u sadržaju, što znači da će bilo koja radnja poput dodavanja šuma, uklanjanja određenih bitova ili niskopropusnog filtriranja odmah biti otkrivena, a sadržaj automatski kompromitovan [6–8]. Kompromis između robusnog i krhkog je polu-krhki vodeni žig (eng. *semi-fragile*) [9–11], koji je koncipiran tako da ostane netaknut pri blagim modifikacijama i šumovima, ali će se oštetiti ili ukloniti ukoliko dođe do značajnijih promjena na signalu.

2.3.2 Kapacitet

Kapacitet digitalnog votermarking sistema odnosi se na količinu informacija, odnosno bitova koji se mogu umetnuti u digitalni sadržaj bez značajnog narušavanja njegovog kvaliteta. Postizanje optimalnog kapaciteta predstavlja još jedan izazov pri dizajnu sistema jer zahtijeva kompromis između želje za umetanjem što više informacija i potrebe za očuvanjem ostalih karakteristika vodenog žiga. Previsok kapacitet može učiniti vodeni žig lako uočljivim ili degradirati kvalitet originalnog signala, dok prenizak kapacitet može ograničiti funkcionalnost žiga, smanjujući količinu informacija koje se mogu prenijeti. U nekim slučajevima, jedan bit može biti sasvim dovoljan jer omogućava binarno tumačenje – na primjer, nula može označavati „ne" ili „zabranjeno", dok se jedinica može tumačiti kao „da" ili „dozvoljeno". Za zaštitu intelektualne svojine, veći broj bitova je neophodan kako bi se označili svi podaci od važnosti. Jedan od pristupa u rješavanju ovog izazova je korišćenje pragmatičnog dizajna, gdje se kapacitet prilagođava na osnovu specifičnih karakteristika sadržaja. Autori u [12] koriste vizuelne modele, koji predstavljaju matematičke modele koji simuliraju način na koji ljudsko oko i vizuelni sistem percipiraju slike i video sadržaj, i shodno tome određuju gornju granicu preko koje bi vodeni žig bio primjetan, praveći kompromise kada je riječ o kapacitetu, robusnosti i neprimjetnosti.

2.3.3 Perceptibilnost

Perceptibilnost se odnosi na sposobnost ljudskog oka da uoči prisustvo vodenog žiga u digitalnoj slici. Kao što je prethodno rečeno, u nekim aplikacijama se iz određenih razloga ne mari za upadljivo prisustvo vodenog žiga na slici, te će oni biti izuzeti dalje diskusije. U velikom broju sistema insistira se na što većem stepenu neprimjetnosti žiga, pritom, naravno, obraćajući pažnju na to da neke od prethodno pomenutih karakteristika budu zadovoljene. Za tu svrhu, eksploatišu se osobine ljudskog vizuelnog sistema, poput frekvencijske osjetljivosti. Naime, poznato je da ljudsko oko manje osjetljivo na promjene u visokim frekvencijama, koje često sadrže sitne detalje poput ivica, u poređenju s niskim frekvencijama, gdje se nalaze ključni oblici i značajniji elementi slike. Ipak, visoke frekvencije igraju važnu ulogu u percepciji detalja, te njihovo odsustvo može dovesti do zamagljenog ili manje definisanog prikaza slike. Razvijene su razne tehnike koje se baziraju upravo na ovom svojstvu i o njima će biti više riječi u narednom poglavlju. Takođe, pokazalo se da su promjene u tamnijim ili kompleksnijim dijelovima obično manje uočljive nego u jednostavnijim, svijetlijim područjima slike.

2.3.4 Složenost implementacije

Kada je riječ o složenosti implementacije, ona zahtijeva pažljivo upravljanje vremenskom i prostornom složenošću, kao dva najbitnija faktora kada govorimo o efikasnosti algoritama. Vremenska složenost odnosi se na količinu vremena potrebnog za umetanje ili ekstrakciju vodenog žiga. Ovaj proces može biti vrlo računski zahtijevan, posebno ako se koriste složeniji algoritmi u frekvencijskom domenu, kao što su diskretna kosinusna transformacija (eng. *discrete cosine transform - DCT*) ili diskretna vejvlet transformacija (eng. *discrete wavelet transform - DWT*). Ovi algoritmi često zahtijevaju obradu velike količine podataka, što povećava vrijeme izvršavanja. Ipak, ako poredimo sa savremenim pristupima umetanja vodenog žiga, poput dubokog učenja, složenost tehnika koji uključuju pomenute transformacije je znatno manja. Kroz Poglavlja 4 i 5 biće jasnije i zašto. U realnim aplikacijama, optimizacija vremenske složenosti je bitna za postizanje brzog i efikasnog rada sistema.

Prostorna složenost, s druge strane, odnosi se na količinu memorije potrebne za implementaciju algoritma. Kada se radi sa velikim fajlovima, poput visoko kvalitetnih slika, algoritmi za umetanje i ekstrakciju vodenih žigova mogu zahtijevati značajne količine memorije za čuvanje međurezultata. Na primjer, frekvencijske transformacije generišu dodatne podatke koji se moraju privremeno skladištiti tokom obrade. Opti-

mizacija prostorne složenosti podrazumijeva efikasno korišćenje memorije kako bi se smanjio njen uticaj na performanse sistema. Danas, velika dostupnost memorije u modernim računarskim sistemima čini ispunjavanje ovog kriterijuma znatno lakšim.

3 Pregled postojećih tehnika umetanja vodenog žiga

Rane devedesete godine prošlog vijeka smatraju se početkom značajnih istraživanja u oblasti digitalnog votermarkinga, kada je postalo jasno da su potrebne nove metode za zaštitu autorskih prava u digitalnom okruženju. Fokus istraživanja bio je prvenstveno na slikama, a većina metoda razvijenih za slike kasnije je prilagođena i proširena za primjenu na audio i video signale. Jedan od radova koji je značajno doprinio razvoju ove oblasti i uticao na mnoge kasnije je [13], gdje autori koriste tehniku širokog spektra (eng. *spread spectrum*), metodu koja se prvobitno koristila u komunikacionim sistemima za smanjenje uticaja šuma i interferencije. Ovom tehnikom se informacija o vodenom žigu raspršuje preko širokog frekvencijskog spektra originalne slike korišćenjem pseudo-slučajnih sekvenci. To za posljedicu ima dobro maskiranje kada je riječ o lokaciji vodenog žiga, a sa dobro odabranim frekvencijskim opsezima i najmanji napad na žig bi vodio ka značajnom narušavanju kvaliteta signala.

Sa prolaskom vremena, zahtjevi postavljeni pred votermarking sisteme postajali su sve veći i kompleksniji, sa potrebom za unaprijeđenjem gdje god je to moguće, što je iznjedrilo veliki broj radova i inovativnih metoda [14–18]. Za klasifikaciju i razvrstavanje votermarking tehnika, kao glavni kriterijum uzima se domen u kojem se vrši umetanje vodenog žiga. Prema tome, razlikujemo tehnike u izvornom domenu, što je za sliku prostorni domen, i tehnike za umetanje žiga u transformacionom domenu. Prostorni domen se fokusira na umetanje žiga direktno u vrijednosti piksela slike, dok frekvencijski domen koristi ranije pomenute ortogonalne transformacije signala, kao što su DCT ili DWT, za umetanje vodenog žiga u koeficijente frekvencijskog spektra.

Razvoj vještačke inteligencije i neuralnih mreža posljednjih godina značajno je uticao na mnoge oblasti, uključujući i digitalni votermarking. Vještačka inteligencija, posebno duboko učenje, omogućila je razvijanje sofisticiranijih i otpornijih tehnika za umetanje i ekstrakciju vodenih žigova. Početni radovi u ovoj oblasti pokazali su veliki potencijal korišćenja neuralnih mreža za poboljšanje procesa votermarkinga. Neuralne mreže, njihova struktura i principi rada, biće predmet obrade u narednom poglavlju.

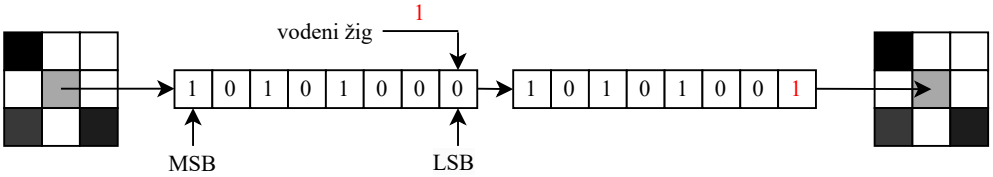
Prvi pokušaji da se duboko učenje primijeni u ovoj oblasti su sa početka 2000-ih i u većini radova realizuju se samo umetač ili detektor, sa ipak nešto većom pažnjom na strani detekcije. U [19, 20] autori predlažu nove šeme dekodiranja koje su bazirane na algoritmima mašinskog učenja, sa ciljem što ispravnije klasifikacije bitova votermarka.

Značajan zamah duboko učenje dobija sredinom prethodne decenije, čemu je značajno doprinio rad [21], čiji uspjeh je poslužio kao katalizator za dalja istraživanja i primjenu dubokih neuralnih mreža. Povećana dostupnost potrebnih računarskih resursa, posebno grafičkih kartica, omogućili su treniranje složenih dubokih modela na način koji ranije nije bio moguć i teško zamisliv. To je otvorilo put za votermarking sisteme visokog kvaliteta koji su u cjelosti zasnovani na modelima dubokog učenja [22–25].

3.1 Tehnike umetanja vodenog žiga u prostornom domenu

Tehnike umetanja vodenog žiga u prostornom domenu najčešće uključuju modifikaciju vrijednosti piksela slike kako bi se ugradili dodatni podaci, odnosno vodeni žig. Prostorni domen je popularan zbog svoje jednostavnosti i niske računske složenosti. Jedna od najraprostranjenih i najosnovnijih tehnika se bazira na promjeni vrijednosti najmanje značajnog bita piksela slike (eng. *least significant bit* - LSB). Ova metoda koristi činjenicu da male promjene u vrijednostima piksela ne utiču značajno na vizuelni kvalitet slike, što omogućava umetanje dodatnih informacija bez uočljivih promjena. Na Slici 5 dat je segment proizvoljne sivoskalirane slike. Sivoskalirane slike sadrže samo jedan kanal i predstavljaju se sa 8 bita po pikselu, pa samim tim mogu imati vrijednost osvjetljaja od 0 (crna boja) do 255 (bijela boja). Piksel u sredini ima dekadnu vrijednost 168 i njegov binarni ekvivalent je 10101000. Umjesto 0 na poziciji najmanje značajnog bita, postavićemo 1 koja predstavlja vodeni žig, pa će umjesto 168, piksel imati vrijednost 169. Uzimajući u obzir da se radi o neznatnoj modifikaciji, dio slike će ostati praktično nepromijenjen.

Ovaj pristup ima nekoliko očiglednih nedostataka. Prvi je prilično niska otpornost na mnoge vrste napada, kao što su šumovi, ili kompresioni algoritmi, koji često uklanjaju ili mijenjaju najmanje značajan bit. Isto tako, zlonamjerni pojedinac može lako detektovati i promijeniti LSB bez značajnog uticaja na kvalitet slike. Problem sa kapacitetom je također prisutan, uzimajući u obzir da se može ugraditi ograničena količina podataka jer se koriste najmanje značajni bitovi.



Slika 5: Ilustracija LSB metode

Istraživači se u nekim radovima odlučuju za izmjenu bitova s većim značajem. Konkretno, autori u [26] unose izmjene u trećem i četvrtom bitu po važnosti, a kao glavni razlog se navodi povećanje sigurnosti, sa premisom da bi neko loših namjera svoju pažnju usmjerio na poziciju najmanje težine. U vezi sa ovim, tehnika umetanja u bite srednje važnosti (eng. *intermediate significant bit* - ISB), kao karakteristična metoda proistekla iz LSB-a, dobila je na značaju u sistemima gdje je potrebna nešto veća robusnost i kapacitet. U radu [27] se kombinuje LSB i ISB pristup, što daje rezultate u pogledu sigurnosti i kapaciteta, bez značajnog narušavanja kvaliteta slike.

Neki od radova koriste histogram kao pomoćno sredstvo za razvijanje algoritama umetanja. U kontekstu digitalne slike, histogramom se pokazuje koliko piksela u slici ima određenu vrijednost osvjetljenja. Na horizontalnoj osi su vrijednosti osvjetljenja od 0 do 255, a na vertikalnoj broj piksela koji imaju odgovarajuće osvjetljenje. U radu [28] se koristi pomjeranje histograma kako bi se u međuprostoru umetnuli bitovi vodenog žiga. Naime, prvo se u histogramu identifikuje maksimum (eng. *peak point*), koji reprezentuje vrijednost osvjetljenja koji ima najveći broj piksela u slici, i nultu tačku (eng. *zero point*), koja predstavlja vrijednost osvjetljaja koja nije prisutna u slici. Recimo da su to vrijednosti 131 (maksimum) i 223 (nulta tačka). Opseg od maksimuma do nulte tačke (132 - 222) se pomjera udesno za 1, tj. povećava im se vrijednost osvjetljaja za 1, ostavljajući iza sebe vrijednost 132 upražnjenom. Pri sljedećem prolasku kroz sliku, kada se dođe do posljednjeg piksela pred upražnjeno mjesto, piksela vrijednosti 131, provjerava se da li sljedeći bit u sekvenci bitova vodenog žiga ima vrijednost 1, i ukoliko ima, vrijednost piksela se povećava za 1 i praznina se popunjava. U suprotnom, vrijednosti piksela se ne mijenjaju. Ovom neznatnom modifikacijom vrijednosti osvjetljenja može se utisnuti značajan broj podataka u sliku, odnosno problem kapaciteta je prevaziđen. Autori kao još jednu bitnu karakteristiku izdvajaju nisku složenost.

Pečvork (eng. *patchwork*) je zanimljiv i jednostavan algoritam baziran na statističkim svojstvima slike [29]. Algoritam nasumično bira parove piksela iz slike, A i B. Zatim, vrijednost osvjetljenja jednog piksela u paru se povećava, dok se vrijednost osvjetljenja drugog smanjuje. Na primjer, vrijednost piksela A se poveća za neku malu vrijednost Δ, dok se vrijednost piksela B smanji za tu istu vrijednost. Ova modifikacija stvara malu, ali ipak mjerljivu razliku između izabranih parova piksela koja je statistički značajna, dok je vizuelno neprimjetna. Postupak umetanja se ponavlja za veliki broj nasumično odabranih parova piksela, čime se osigurava da vodeni žig bude raširen po cijeloj slici, što značajno doprinosi robusnosti. Dekodiranje se vrši po formuli:

$$d = \sum_{i=1}^{n} (A_i + \Delta) - (B_i - \Delta), \tag{4}$$

gdje je d promjena u razlici između parova piksela prije i poslije umetanja vodenog žiga, a n broj iteracija. Ako je razlika prisutna, to ukazuje na prisutnost vodenog žiga. Zbog nasumičnog odabira parova, ovaj algoritam zahtijeva dobru sinhronizaciju između procesa umetanja i ekstrakcije, što predstavlja otežavajući faktor. Kapacitet predstavlja izazov i za ovu tehniku, jer promjene moraju biti minimalne kako bi ostale neprimjetne.

3.2 Tehnike umetanja vodenog žiga u transformacionom domenu

Umetanje vodenog žiga u transformacionom domenu predstavlja sofisticiraniji pristup zaštiti digitalnog sadržaja u poređenju sa tehnikama koje se oslanjaju na prostorni domen. U prostornom domenu, operacije i modifikacije su se sprovodile direktno nad pikselima slike. U transformacionom domenu, signal se prvo matematičkim transformacijama prebacuje u drugi, pogodniji oblik, a zatim se vodeni žig umeće u koeficijente transformisanog signala. Uvođenje ove dodatne aktivnosti u odnosu na jednostavnost umetanja žiga u prostornom domenu pravdamo sa nekoliko činjenica. Prva je značajan dobitak na robusnosti. Koeficijenti transformisanog signala su manje podložni promjenama nego pikseli izvorne slike. Dalje, transformacioni domen omogućava efikasniju analizu i prepoznavanje karakteristika slike koje mogu biti od koristi, dok u prostornom domenu može biti teže izolovati specifične informacije (primjer upotrebe histograma). I na kraju, koeficijenti u transformacionom domenu omogućavaju bolju kompresiju i filtriranje, operacije koje su veliki problem u prostornom domenu.

U oblasti obrade signala, postoji veliki broj transformacionih domena koji se koriste za različite tehnike i namjene. Kao najbitnije za oblast umetanja vodenog žiga u digitalne slike, autori sveobuhvatnog pregleda votermarking tehnika [30] su izdvojili ranije pomenute DWT i DCT, sa dodatkom diskretne Furijeove transformacije (eng. *discrete Fourier transform* - DFT), te ćemo slijediti njihovu selekciju i osvrnuti se na neke od značajnijih radova za svaki domen i ideje iza njih.

3.2.1 Diskretna kosinusna transformacija

Diskretna kosinusna transformacija je matematička transformacija koja preslikava signal iz prostornog domena u frekvencijski domen. U osnovi, DCT razlaže signal na sumu kosinusnih funkcija sa različitim frekvencijama i amplitudama. DCT je široko primijenjena u oblasti obrade signala, jer omogućava efikasno predstavljanje

podataka u obliku koji je pogodniji za kompresiju i analizu. Diskretnu kosinusnu transformaciju u 2D obliku definišemo kao [31]:

$$C(k_1,k_2) = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} 4x(n_1,n_2)\cos\frac{(2n_1+1)\pi k_1}{2N_1}\cos\frac{(2n_2+1)\pi k_2}{2N_2} \quad (5)$$
$$\text{za } 0 \le k_1 \le N_1-1,\ 0 \le k_2 \le N_2-1,$$

dok je inverzna transformacija:

$$x(n_1,n_2) = \sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} w_1(k_1)w_2(k_2)C(k_1,k_2)\cos\frac{(2n_1+1)\pi k_1}{2N_1}\cos\frac{(2n_2+1)\pi k_2}{2N_2}, \quad (6)$$

gdje je $w_i(k_i)$, $i = 1, 2$ dato kao:

$$w_i(k_i) = \begin{cases} \frac{1}{2}, & \text{za } k_i = 0 \\ 1, & \text{za } 1 \le k_i \le N-1. \end{cases} \quad (7)$$

$C(k_1, k_2)$ je koeficijent transformacije, $x(n_1, n_2)$ originalni signal u prostornom domenu, N_1 i N_2 dimenzije signala, dok su $w_i k_i$ skalirajući faktori.

Polu-krhki pristup umetanja pomoću DCT prikazan je u [32] gdje se autori bave dizajnom vodenog žiga za autentifikaciju JPEG slika. Sistem je podešen tako da bude otporan na određene prihvatljive manipulacije, prihvatajući JPEG kompresiju kao neizostavan dio obrade slike, dok obrade poput niskopropusnog filtriranja vode ka uništavanju žiga. Algoritam funkcioniše tako što se slika dijeli na 8×8 blokove, radi se transformacija i kvantizuje na unaprijed određeni kvalitet. Četiri koeficijenta iz viših frekvencija se biraju iz svakog bloka za proces votermarkovanja. Na osnovu znaka i amplitude koeficijenata, generišu se binarni kodovi, a zatim se vrši provjera parnosti i paritet se ugrađuje u koeficijent. Za autentifikaciju, provjerava se paritet, i ukoliko ne dolazi do poklapanja, slika je označena kao manipulisana.

Tehnika predstavljena u [33] koristi DCT u kombinaciji sa selektivnim odabiranjem (eng. *subsampling*). Pod tim podrazumijevamo smanjenje rezolucije slike tako što se zadržava samo određeni broj piksela, dok se ostali ignorišu. U koderu se originalna slika selektivnim odabiranjem dijeli na podslike nad kojima se vrši DCT kako bi se dobili koeficijenti u frekvencijskom domenu. Koeficijenti se biraju u parovima iz podslika. Oni koeficijenti koji se nalaze na istim pozicijama u različitim podslikama su povezani u parove. Algoritam koristi cik-cak redosljed (eng. *zig-zag scan*), koji je isti kao i u JPEG algoritmu, kako bi obišao koeficijente od najnižih do najviših frekvencija. Parovi koeficijenata se prvo procjenjuju na osnovu njihove sličnosti. Ako je razlika između dva koeficijenta prevelika, oni se ne modifikuju, jer bi to moglo ugroziti kvalitet slike. U suprotnom, ako je razlika mala (što ukazuje da su teksture

originalne slike u toj oblasti slične), vrši se umetanje vodenog žiga dodavanjem sitnih modifikacija u oba koeficijenta. Vodeni žig je sekvenca nasumično generisanih bitova. Proces rekonstrukcije ne zahtijeva originalnu sliku, što znači da je riječ o neinformisanom detektoru. Prilikom dekodiranja, sistem računa razlike između parova koeficijenata i iz tih razlika izvlači vodeni žig.

Najniži koeficijenti, koji predstavljaju niske frekvencije, nose glavne informacije o slici kao što su velike, ravne površine ili postepeni prelazi boja i osvjetljenja. Iako bi umetanjem bitova žiga u tim frekvencijama dobili na robusnosti, kvalitet slike bi bio narušen, pa se zbog toga oni nekada preskaču pri umetanju vodenog žiga. To je slučaj u [34], gdje se informacije ugrađuju samo u višim frekvencijama. Slično prethodnoj realizaciji, i ovdje je prisutan cik-cak obilazak i detektor je neinformisan. Dalje, uzmimo da broj N predstavlja broj koeficijenata niske frekvencije, a V broj koeficijenata na višim frekvencijama, i vektor $C = \{c_1, c_2, ..., c_N, c_{N+1}, ..., c_{N+V}\}$, koji sadrži $N+V$ koeficijenata transformacije. Vodeni žig, koji je pseudo-nasumična sekvenca realnih brojeva, predstavimo takođe vektorom $W = \{w_1, w_2, ..., w_V\}$. Vektor koeficijenata više frekvencije označen vodenim žigom $C' = \{c_1, c_2, ..., c_N, c'_{N+1}, ..., c'_{N+V}\}$ dobija se sljedećom jednakošću:

$$c'_{N+1} = c_{N+1} + \alpha \left| c_{N+1} \right| w_i, \tag{8}$$

gdje je α faktor jačine vodenog žiga, a $i = 1, ..., M$. Detekcija vodenog žiga se vrši računanjem korelacije između koeficijenata sa originalnim votermarkom i koeficijenata nakon prolaska kroz prenosni kanal. Ako je vrijednost korelacije iznad određenog praga, detekcija se smatra uspješnom. Algoritam pokazuje zavidan stepen robusnosti kada je riječ o napadima poput filtriranja, dodavanja šuma, skaliranja i odsjecanja.

U nekim situacijama, kada potreba nalaže da žig bude vidljiv i odvraćajuće prirode, to je moguće sprovesti efikasno upotrebom DCT, pri čemu se faktori za umetanje vodenog žiga računaju na osnovu karakteristika tekstura i ivica slike [35]. Neke ideje iz prostornog domena je moguće prenijeti u frekvencijski, i obrnuto, pa tako u [36] autori informacije umeću u najmanje značajne bitove niskofrekventnih koeficijenata transformacije. Nešto složenijim postupkom se u [37] umeću višestruki žigovi u zeleni i plavi kanal slike što daje odlične rezultate u pogledu robusnosti i sigurnosti, dok trpi složenost cijelog procesa.

3.2.2 Diskretna Furijeova transformacija

Diskretna Furijeova transformacija je fundamentalni alat u analizi i obradi digitalnih signala. DFT omogućava pretvaranje diskretnih podataka u vremenskom ili prostornom domenu u njihov ekvivalent u frekvencijskom domenu. Ova transformacija pretvara vremenski ili prostorni signal u niz kompleksnih brojeva koji predstavljaju

amplitudu i fazu njegovih frekvencijskih komponenti. Matematički, transformacija u 2D obliku za signal ograničenog trajanja i njena inverzna verzija su definisane kao [31]:

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) e^{-j2\pi(\frac{n_1 k_1}{N_1} + \frac{n_2 k_2}{N_2})}, \tag{9}$$

$$f(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} F(k_1, k_2) e^{j2\pi(\frac{n_1 k_1}{N_1} + \frac{n_2 k_2}{N_2})}. \tag{10}$$

gdje je $F(k_1, k_2)$ koeficijent transformacije na poziciji (k_1, k_2), $f(n_1, n_2)$ originalni signal u prostornom domenu, a N i M dimenzije signala.

Glavna i osnovna razlika između DCT i DFT jeste razlika u spektru. Rezultat DCT-a je realan spektar koji predstavlja amplitudu frekvencijskih komponenti, dok je rezultat DFT-a kompleksni spektar koji uključuje informacije o amplitudi i fazi frekvencijskih komponenti. Sa dodatkom informacije o fazi, DFT pronalazi veliku primjenu u telekomunikacijama i obradi audio signala. Ipak, kompleksna priroda DFT-a sa sobom nosi veću računsku složenost i memorijske zahtjeve u odnosu na računanje DCT. Ovaj problem je prevaziđen razvojem velikog broja algoritama za brzo računanje, od kojih je najpoznatiji brza Furijeova transformacija (eng. *Fast Fourier Transform* - FFT).

U radu [38] se vodeni žig umeće kružno simetrično u opseg srednjih frekvencija slike koristeći 2D DFT. Simetrija omogućava otpornost na manipulacije slikom koje uključuju rotaciju, dok je odabir srednjih frekvencija za umetanje žiga nastojanje da se izbjegnu vidljive promjene na slici i zadrži robusnost kada je riječ o kompresijama. Proces umetanja vodenog žiga može biti aditivan ili multiplikativan. U aditivnom modelu, nova informacija se dodaje bez obzira na vrijednost originalnog signala, dok multiplikativni model utiče na signal u zavisnosti od njegove trenutne vrijednosti, pa jačina efekta (vodenog žiga) zavisi od intenziteta originalnog signala. Detekcija se vrši putem korelacije između originalnog vodenog žiga i koeficijenata sa umetnutim žigom, bez potrebe za originalnom slikom. Metoda pokazuje zavidan nivo robusnosti i otpornosti na razne napade kao što su skaliranje, rotacija, šum, filtriranje i odsjecanje. Pored kompleksnosti, kao potencijalnu slabost možemo izdvojiti ograničenu otpornost na veće rotacije. Iako kružna simetrija vodenog žiga pruža otpornost na manje rotacije, kod većih uglova rotacije metoda može zahtijevati dodatne pretrage kroz različite rotacione varijante, što povećava rizik da vodeni žig ne bude pravilno detektovan. Radovi [39, 40] takođe naglašavaju robusnost sistema u odnosu na geometrijske transformacije, oslanjajući se na efikasnost DFT-a u odbrani od ovakvih vrsta napada.

3.2.3 Diskretna vejvlet transformacija

Diskretna vejvlet transformacija je metoda za analizu signala koja omogućava razlaganje signala na različite frekvencijske komponente u različitim vremenskim intervalima. Za razliku od prethodno pomenutih transformacija koje pružaju jedinstveni frekvencijski spektar signala, DWT pruža uvid kako se frekvencije i energija signala mijenjaju tokom vremena. Na primjer, pomoću DWT možemo detektovati proizvoljnu promjenu u signalu i vremenski okvir kada se ona dogodila, dok u klasičnoj Furijeovoj transformaciji možemo zapaziti istu tu promjenu, ali bez informacije o vremenu. Sa kratkotrajnom Furijeovom transformacijom (eng. *Short-time Fourier Transform* - STFT) se ovo teži premostiti tako što se uzimaju dijelovi signala, tzv. prozori, u određenim vremenskim trenucima, i nad njima sprovodi Furijeova transformacija. Ipak, kako kaže autor u [41], besplatnog ručka, koji je oduzeo Hajzenbergov princip neodređenosti [42], nema, jer se fokusom na kraće vremenske intervale i boljim uočavanjem promjena u datom vremenskom trenutku gubi na frekvencijskoj rezoluciji, i obrnuto, uzimanje većeg vremenskog prozora zamagljuje vrijeme pojave određene frekvencijske komponente. Iza vejvleta stoji velika teorijska i matematička pozadina, u koju nećemo detaljno zalaziti jer nije centralna tema rada, pa će u nastavku biti izloženi samo osnovni koncepti neophodni za poimanje radova koji koriste ovu transformaciju.

U Furijeovoj transformaciji, originalni signal je podrazumijevano predstavljen nizom sinusoida, dok kod vejvleta sami biramo „talasić" kojim ćemo prikazati originalni signal. Izbor zavisi od oblika signala koji se želi predstaviti, dok u nekim slučajevima empirijski dolazimo do optimalnog rješenja. Najpoznatiji vejvleti kada je obrada slike u pitanju su Haar-ov, koji je ujedno najstariji i najjednostavniji, i Dobeši (Daubechies) vejvleti.

Kada se primijenjuje DWT, signal se dijeli na različite rezolucione nivoe. Na svakom nivou dekompozicije postoje komponente aproksimacija i detalja. Aproksimacija predstavlja generalni oblik signala (nisko-frekvencijske komponente), dok detalji opisuju finije karakteristike (visoko-frekvencijske komponente). Slikom 6 je prikazana DWT dekompozicija na 3 nivoa signala. pri čemu se na svakom nivou dobijaju koeficijenti detalja pomoću niskopropusnog filtra (H_0), dok se aproksimacije dalje razlažu pomoću visokopropusnog filtra (G_0).

Roditeljski vejvlet (eng. *mother wavelet*) je vejvlet koji opisuje aproksimacione koeficijente na višem nivou dekompozicije. S druge strane, dječiji vejvleti (eng. *daughter wavelet*) opisuju detalje na nižim nivoima dekompozicije, koji proizilaze iz roditeljskih komponenti. Oni se mogu skalirati i translirati kako bi se signal analizirao na različitim nivoima detalja. Uzmimo za primjer najprostiji, Haar-ov vejvlet, koji je

definisan kao:

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2} \\ -1, & \frac{1}{2} \leq t < 1 \\ 0, & \text{inače.} \end{cases} \tag{11}$$

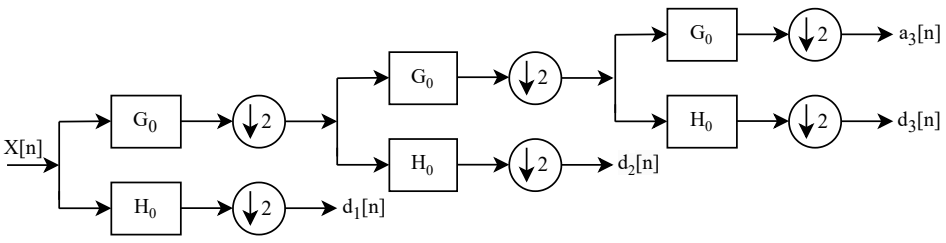
Translirani i skalirani oblik roditeljskog vejvleta (11), dječiji vejvlet, ima oblik:

$$\psi_{j,k} = \psi(2^j t - k) \tag{12}$$

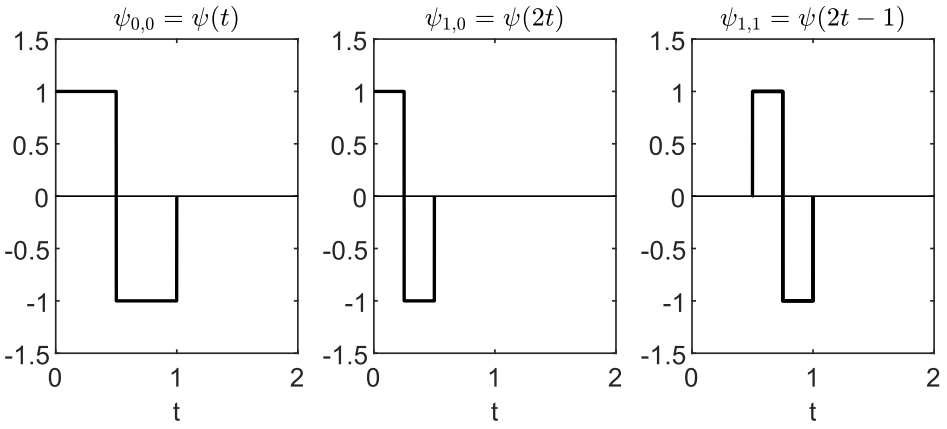
gdje indeksi j i k definišu konkretne oblike funkcije na različitim vremenskim i frekvencijskim nivoima. Veće vrijednosti j povećavaju frekvenciju funkcije (kompresuju je u vremenskom domenu), dok manje vrijednosti j smanjuju frekvenciju (rastežu funkciju u vremenskom domenu), dok je k parametar translacije koji određuje poziciju vejvleta duž vremenske ose. Na Slici 7 je prikazan roditeljski Haar-ov vejvlet, kao i izvedeni oblici signala koji se dobijaju za različite vrijednosti j i k.

Upotrebom DWT, realizovan je veliki broj radova sa polu-krhkim pristupom umetanja vodenog žiga. Primjenom Haar-ovog vejvleta se u [43] omogućava ne samo otkrivanje manipulacije nad slikom, već i precizno lociranje promjena u specifičnim prostornim i frekvencijskim regionima slike. Variranjem parametra kvantizacije, koji određuje koliko fino se vejvlet koeficijenti kvantizuju, reguliše se krhkost vodenog žiga. Kao razlog za izbor Haar-ovog vejvleta, autori naglašavaju to da promjene u vejvlet koeficijentima garantuju cjelobrojne promjene u prostornom domenu, čime se izbjegavaju numeričke poteškoće prilikom zaokruživanja piksela na cijele vrijednosti u digitalnoj slici, što bi moglo uzrokovati odstupanja u umetnutom vodenom žigu. Umjesto kvantizacije pojedinačnih vejvlet koeficijenata, rad [44] koristi kvantizaciju srednje vrijednosti grupe koeficijenata. Ovim pristupom se veća pažnja posvećuje robusnosti na sporadične modifikacije slike, i nastoji napraviti još veća distinkcija između malicioznog djelovanja i slučajnih oštećenja.

U radu [45] vodeni žig i originalni signal su sivoskalirane slike. Nad njima se sprovodi DWT dekompozicija na 3 nivoa, a zatim se vodeni žig umeće tzv. tehnikom alfa miješanja (eng. *alpha blending*). Ova tehnika podrazumijeva da se razložene



Slika 6: DWT dekompozicija na 3 nivoa



Slika 7: Haar-ov vejvlet

komponente originalne slike i vodenog žiga množe faktorima skaliranja i onda sabiraju. Na kraju se vrši inverzna DWT kako bi se dobila konačna slika sa umetnutim žigom. Podešavanjem skalirajućih faktora na odgovarajuće vrijednosti, algoritam daje bolje rezultate u odnosu na realizacije sa manjim nivoima dekompozicije.

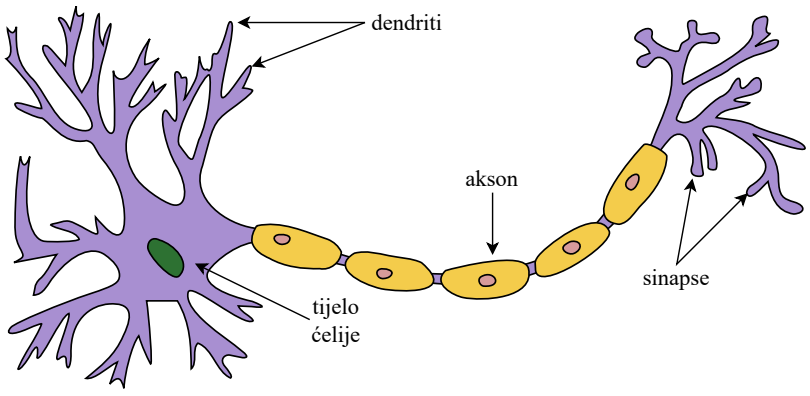
Česta je pojava upotreba više transformacionih domena (najčešće 2, ponekad i 3) u algoritmima za umetanje vodenog žiga kako bi se iskoristile prednosti odabranih transformacija. DWT se često kombinuje sa DCT zbog specifičnih prednosti oba domena, i algoritmi nastali povezivanjem više domena imaju naglašenu robusnost i neprimjetnost žiga kao glavne prednosti. Zajednička osobina svih algoritama nastalih fuzijom domena je složenost i težina implementacije algoritma, i poteškoće prilikom ekstrahovanja vodenog žiga na strani detekcije kao posljedica loše sinhronizacije [46–48].

4 Neuralne mreže

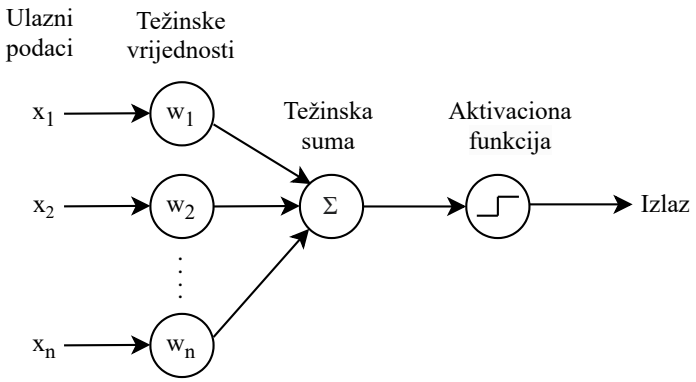
Neuralne mreže su računarski modeli inspirisani strukturom i funkcijom ljudskog mozga. Biološki neuron se sastoji od dendrita, koji primaju ulazne signale koji se obrađuju u tijelu neurona, a zatim se impulsi preko aksona i sinapsi prenose dalje do sljedećeg neurona. Pojednostavljeni prikaz biološkog neurona dat je Slikom 8.

Osnovni elementi vještačke neuralne mreže su čvorovi, poznati kao neuroni. Svaki neuron prima ulazne podatke, vrši nad njima određenu operaciju i zatim šalje rezultat drugim neuronima. Kao što su biološki neuroni povezani u složene mreže kroz sinapse, vještački neuroni su povezani u slojeve, pri čemu su izlazi jednog sloja ulazi sljedećeg. Slojevi su organizovani hijerarhijski, i sastoje se od međusobno povezanih neurona. Broj slojeva i način na koji su neuroni unutar njih povezani čini arhitekturu neuralne mreže. Ulazni sloj predstavlja podatke koji se bez obrade dalje prosljeđuju u sljedeći sloj. Iz ovog razloga, kada pričamo o broju slojeva mreže, prvi ne uzimamo u obzir.

Skriveni slojevi, koji se nalaze između ulaznog i izlaznog sloja, predstavljaju srž svake neuralne mreže. Njihova glavna uloga je da obrađuju, transformišu i ekstrahuju informacije iz ulaza kroz niz matematičkih operacija. Veći broj skrivenih slojeva omogućava mreži da uči i prepoznaje složenije obrasce u podacima. Mreže sa većim brojem skrivenih slojeva nazivaju se dubokim neuralnim mrežama (eng. *deep neural networks*), a oblast koja se bavi izučavanjem i upotrebom ovih tipova mreža je duboko učenje (eng. *deep learning*). Izlazni sloj, na kraju, daje konačnu predikciju ili odluku na osnovu svih prethodnih obrada.



Slika 8: Ilustracija biološkog neurona. Autor: Quasar Jarosz, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7616130



Slika 9: Perceptron

Istorija neuralnih mreža počinje 50-ih godina prošlog vijeka sa [49], gdje je predstavljen jednostavan matematički model koji oponaša osnovne osobine bioloških neurona. Neki autori kao stvarni početak razvoja neuralnih mreža uzimaju rad Frenka Rozenblata [50], u kojem on uvodi jedan od prvih formalnih modela vještačkog neurona, i naziva ga perceptron (Slika 9). Rozenblatov perceptron je značajan sa stanovišta da je omogućio učenje kroz iterativno prilagođavanje težina (eng. *weights*) na osnovu grešaka. Njegova glavna uloga je svrstavanje podataka u dvije klase - binarna klasifikacija.

Perceptron prima niz ulaznih vrijednosti, $x_1, x_2, ..., x_n$, koji mogu biti različiti atributi nekog objekta ili pojave. Težinske vrijednosti $w_1, w_2, ..., w_n$, za koje su vezani ulazni podaci, određuju koliki uticaj određeni ulaz ima na izlaz perceptrona. U procesu učenja, težine se prilagođavaju tako da mreža može bolje generalizovati obrasce i donositi tačnije predikcije na novim, neviđenim podacima. Matematički, za neuron sa n ulaza, težinska suma se računa kao:

$$z = w_1x_1 + w_2x_2 + ... + w_nx_n = \sum_{i=1}^{n} w_ix_i. \tag{13}$$

Za veću sposobnost adaptacije i tačnost u klasifikaciji ili predikciji, modelu se često dodaje slobodni član b (eng. *bias*). Bez slobodnog člana, izlazni neuron bi bio ograničen tako da može generisati samo funkcije koje prolaze kroz koordinatni početak. Sa ovim dodatkom, jednačina (13) poprima sljedeći oblik:

$$z = \sum_{i=1}^{n} w_ix_i + b. \tag{14}$$

Konačan izlaz neurona dobija se kada se zbir težinskih suma dovede na aktivacionu funkciju, kroz koju se u mrežu uvodi nelinearnost ili prag, pomoću koga se na izlazu

donosi odluka. Detaljnija diskusija o ulozi aktivacionih funkcija i njihovim različitim tipovima je u nastavku poglavlja. Na kraju, poslije primjene aktivacione funkcije f, izlazni signal y ima oblik:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right). \tag{15}$$

Najveći nedostatak perceptrona je linearna separabilnost, jer on može rješavati samo one zadatke u kojima se podaci mogu razdvojiti pravom linijom (u 2D prostoru). Da to unosi značajna ograničenja i kod prilično jednostavnih problema, dokazuje slučaj operacije ekskluzivnog ili (eng. *XOR*). Ako stanja ovog logičkog kola prenesemo na koordinatni sistem, uočićemo da ne postoji prava linija koja može razdvojiti tačke koje pripadaju različitim klasama (0 i 1). Ovo ograničenje je razlog zašto su kasnije razvijeni složeniji modeli, kao što su višeslojni perceptroni, koji koriste više skrivenih slojeva i nelinearne aktivacione funkcije, pa stoga mogu rješavati i nelinearne probleme.

Stagnacija u istraživanju neuralnih mreža i nezainteresovanost naučne zajednice za ovu podoblast vještačke inteligencije tokom sedamdesetih godina pripisuje se, između ostalih razloga, nedovoljno moćnom hardveru koji bi mogao da isprati teorijske napretke i nove ideje. Napredak u računarskoj snazi, pristup velikim skupovima podataka i inovacije u algoritmima i arhitekturama, vratile su početkom ovog vijeka neuralne mreže na glavnu pozornicu.

U cilju jasnijeg izlaganja načete teme, nastavak poglavlja je organizovan podsekcijski, pri čemu su u svakoj podsekciji izložene i detaljno pokrivene funkcije, algoritmi i ideje od važnosti za ovu oblast i predloženu arhitekturu.

4.1 Aktivacione funkcije

Aktivacione funkcije su matematičke funkcije koje služe da transformišu izlaz svakog neurona u mreži na način koji omogućava sistemu da prepozna i modeluje složene obrasce u podacima. Bez aktivacione funkcije, svaki sloj mreže bi vršio samo linearnu transformaciju ulaza, što znači da bi se, bez obzira na broj slojeva, neuralna mreža ponašala kao linearni model. Ovo unosi značajna ograničenja u sistem i sužava broj problema koji se mogu ispravno riješiti na vrlo mali broj. Izborom aktivacione funkcije u mrežu se unosi nelinearnost, čime joj dajemo sposobnost da „odlučuje" koje informacije su od značaja i kako ih dalje prenijeti kroz mrežu. Najbolja, univerzalna aktivaciona funkcija koja odgovara svim problemima, ne postoji. Efikasnost aktivacione funkcije zavisi od specifičnosti zadatka, strukture modela i

prirode podataka. Zbog toga se preporučuje testiranje različitih funkcija i pažljiva evaluacija rezultata.

Ranije pomenuti perceptron obično koristi jediničnu (step) funkciju kao aktivacionu. Za vrijednost veću od praga (eng. *threshold*), na izlazu daje 1, dok je za vrijednosti manje od praga na izlazu 0. Matematička forma step funkcije data je jednačinom (16):

$$f(x) = \begin{cases} 1, & \text{ako je } x \geq 0 \\ 0, & \text{ako je } x < 0. \end{cases} \tag{16}$$

Jedinična funkcija je danas rijetko u upotrebi, a kao glavni razlog se može uzeti njena diskontinuitetna priroda. Kao što se vidi na Slici 10, funkcija pravi nagli prelaz između dvije vrijednosti, 0 i 1, kada ulaz pređe određeni prag, u ovom slučaju 0. Ovaj nagli skok onemogućava glatku tranziciju, što znači da funkcija nema definisan izvod u tački prelaza. Ovaj problem se može prevazići, jer se izvod može aproksimirati ili precizno definisati u datim uslovima.

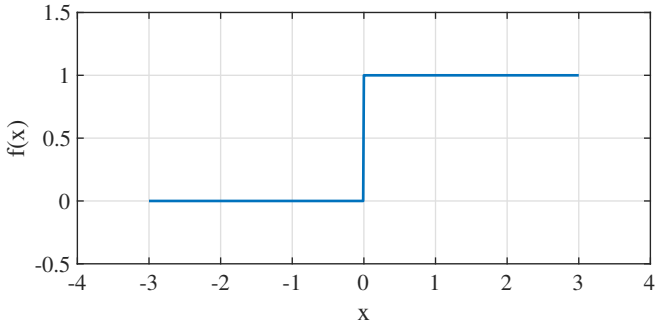
Da bi se nadomjestila ograničenja step funkcije, uvedena je sigmoidna funkcija čiji je oblik:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{17}$$

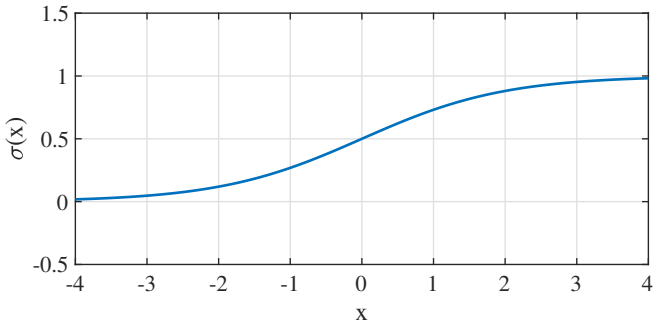
sa prvim izvodom:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)). \tag{18}$$

Izlaz sigmoidne funkcije se kreće između 0 i 1. Funkcija ima karakterističnu krivu u obliku latiničnog slova „S”, tzv. sigmoidnu krivu, što znači da se promjene izlaza događaju postepeno, bez naglih prelaza kao što je to bio slučaj kod step funkcije (Slika 11). Ovaj glatki prelaz kao posljedicu ima lakše učenje i optimizaciju modela, uzimajući u obzir i jednostavnost derivacije što je od velikog značaja za veliki broj optimizacionih algoritama.



Slika 10: Step aktivaciona funkcija



Slika 11: Sigmoid

Međutim, iako značajno unaprijeđenje u odnosu na jediničnu, sigmoidna aktivaciona funkcija ipak ima svoje nedostatke, naročito u kontekstu problema poput iščezavanja gradijenata (eng. *vanishing gradients*). Posmatrajući relaciju (18) zajedno sa Slikom 11, vidimo da kada x uzima veće vrijednosti, $\sigma(x)$ se približava 1, što čini $1 - \sigma(x)$ veoma malim. U tom slučaju, proizvod $\sigma(x)(1 - \sigma(x))$ teži 0, što rezultira malom vrijednošću gradijenta. U slučaju male ulazne vrijednosti x vrijednost prvog izvoda se ne mijenja, samo što sada prvi činilac uzima vrijednost približnu 0. Dakle, za ekstremne ulazne vrijednosti, sigmoidna funkcija ulazi u zasićenje, što dovodi do toga da se težine neurona ažuriraju veoma sporo ili nikako. Ovo može uzrokovati značajne probleme u mrežama sa većim brojem slojeva, otežavajući učenje i samim tim ograničavajući njihove mogućnosti.

Funkcija slična sigmoidu je hiperbolički tangens. Matematički se definiše kao:

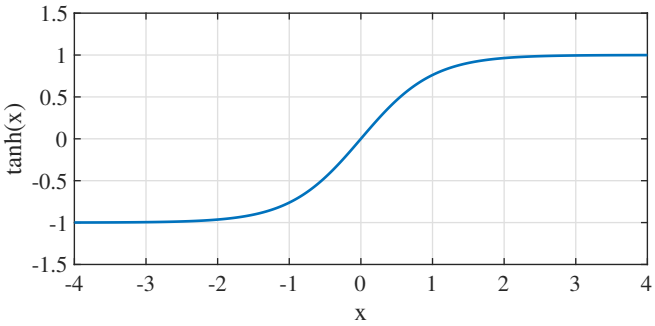
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{19}$$

Hiperbolički tangens ulaz preslikava u simetrični opseg [-1, 1] i tu leži glavna prednost i razlika u odnosu na sigmoid (Slika 12). Problem iščezavanja gradijenata prisutan je i kod ove aktivacione funkcije, dok je računanje gradijenata nešto složenije.

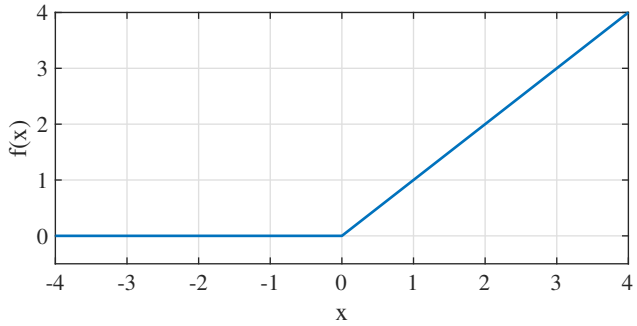
Često korišćena i popularna aktivaciona funkcija u dubokom učenju je ReLU (eng. *Rectified Linear Unit*). Definisana je kao:

$$f(x) = \begin{cases} x, & \text{ako je } x > 0 \\ 0, & \text{ako je } x \leq 0. \end{cases} \tag{20}$$

ReLU je linearna funkcija sa nagibom 1 za pozitivne vrijednosti, dok za negativne ulazne vrijednosti dodjeljuje 0 (Slika 13). Za razliku od sigmoidne funkcije, ReLU ne ulazi u zasićenje za veće pozitivne ulazne vrijednosti, što znači da gradijenti ostaju veći i omogućavaju efikasno propagiranje informacija u mrežama sa većim brojem slojeva. Dodatna prednost u odnosu na sigmoid je u jednostavnosti izračunavanja,



Slika 12: Hiperbolički tangens



Slika 13: ReLU

jer ReLU zahtijeva samo poređenje sa nulom (uzima maksimalnu vrijednost između 0 i ulaza), dok su kod sigmoida i hiperboličkog tangensa prisutne eksponencijalne funkcije. Nedostatak ReLU-a jesu negativne ulazne vrijednosti, kada gradijent postaje 0, što vodi do situacije poznate kao „mrtvi neuroni" (eng. *dead neurons*), u kojoj određeni čvorovi nisu aktivni i gube sposobnost učenja. Ovaj problem se pokušava prevazići razvijanjem varijanti ReLU-a, poput propuštajućeg ReLU-a (eng. *Leaky ReLU*), koji za negativne vrijednosti ulaza dodjeljuje malu negativnu vrijednost različitu od 0.

4.2 Konvolucione neuralne mreže

Konvolucione neuralne mreže (eng. *Convolutional Neural Networks* - CNN) predstavljaju moćan alat u dubokom učenju, i danas se dominantno koriste u zadacima obrade slike. Primjenom CNN realizovana je i predložena arhitektura umetanja vodenog žiga, te će u nastavku biti dat kratak pregled i ideja iza ove vrste neuralnih mreža.

Objasnimo prvo porijeklo imena. Konvolucione neuralne mreže su ime dobile po operaciji konvolucije koja se vrši u njihovim slojevima. Konvolucija se definiše kao skalarni proizvod funkcije x i skalirane, vremenski pomjerene funkcije h, gdje je

rezultat treća funkcija y [51]:

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau. \tag{21}$$

Dobijena funkcija y u suštini pokazuje kako se oblik jedne funkcije mijenja pod uticajem druge funkcije. Konvolucija u diskretnom obliku se definiše kao:

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k). \tag{22}$$

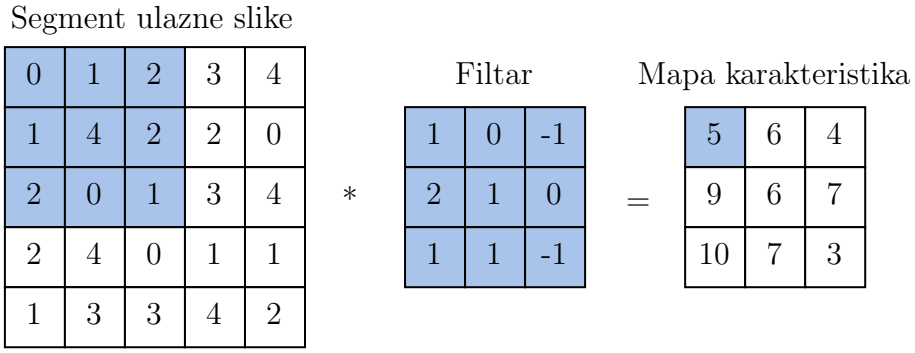
U kontekstu CNN, operaciju konvolucije vežemo za primjenu konvolucionih filtara nad slikom u konvolucionim slojevima neuralne mreže. Filtriranjem se poboljšavaju ili izdvajaju karakteristike slike na osnovu prethodnog poznavanja njene strukture i osobina. Konvolucija u 2D obliku ulaznog signala $x(n_1, n_2)$ i impulsnog odziva sistema $h(n_1, n_2)$ je zapravo filtriranje linearnim prostorno-invarijantnim sistemom [31]:

$$y(n_1, n_2) = \sum_{k_1=0}^{K_1} \sum_{k_2=0}^{K_2} h(k_1, k_2)x(n_1 - k_1, n_2 - k_2). \tag{23}$$

Kao što je već rečeno, ulazni podaci konvolucionih neuralnih mreža su najčešće slike, koje su predstavljene sa 3 dimenzije: širinom (eng. *width*), visinom (eng. *height*) i dubinom, ili brojem kanala, koji je najčešće 3 (RGB aditivni model). Kao što otisak prsta svakog čovjeka sadrži jedinstvene karakteristike koje ga razlikuju od drugih, tako i svaka slika ili objekat imaju specifičnosti koje omogućavaju prepoznavanje i razlikovanje. Ovu logiku koriste CNN tako što se u konvolucionim slojevima primijenjuju konvolucioni filtri nad slikom, u procesu izvlačeći glavne karakteristike slike (eng. *features*) poput ivica, uglova i tekstura.

Rezultati primjene konvolucionih filtara se smještaju u matricu koja se naziva mapa karakteristika (eng. *feature map*). U slojevima nakon prvog, konvolucioni filtri se primijenjuju nad mapom, tretirajući ih kao pseudo-ulaze mreže, tako generišući nove mape kojima se izdvajaju sve složenije i apstraktnije karakteristike. Dimenzije ove matrice zavisne su od više faktora, o kojima će biti riječi u nastavku.

Osvrnimo se detaljnije na filtre i način na koji se dolazi do mape karakteristika. Konvolucioni filtri su matrice realnih vrijednosti dimenzija mnogo manjih od ulazne slike. U praksi se obično uzimaju filtri dimenzija 3×3 ili 5×5, jer hvataju relativno dovoljan broj lokalnih obrazaca slike uz malu računsku složenost. Slika 14 ilustruje pojednostavljen postupak konvolucije nad proizvoljnim segmentom ulazne RGB slike dimenzija 5×5 filtrom veličine 3×3. Filtrom se prelazi preko odgovarajućih podmatrica dijela ulazne matrice, a sabiranjem proizvoda vrijednosti na istim pozicijama dvije matrice dobijaju se vrijednosti mape karakteristika.



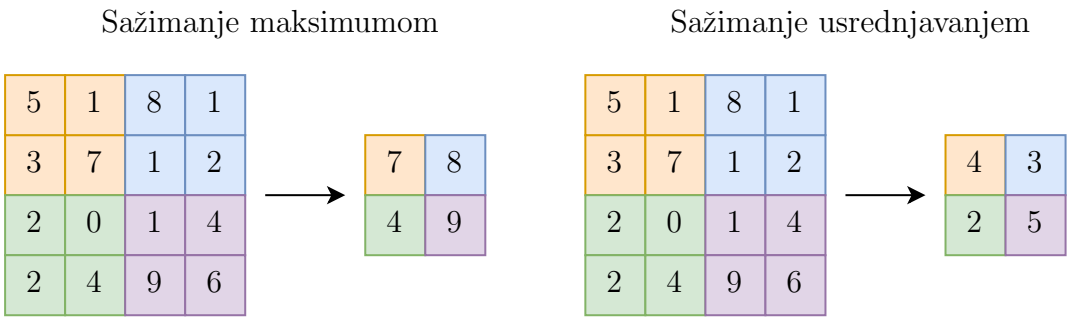
Slika 14: Primjena filtra nad slikom u konvolucionom sloju sa korakom 1

U primjeru sa slike je filtrom obuhvaćen svaki piksel ulazne slike, što ne mora uvijek biti slučaj. To se reguliše parametrom koraka (eng. *stride*). Korak definiše koliko se filtar pomjera horizontalno ili vertikalno, odnosno koliko piksela se preskače. Povećanjem koraka, smanjuju se dimenzije izlazne mape karakteristika i računska složenost, što je od velike koristi u većini situacija. Ne treba izgubiti iz vida da preskočeni pikseli (u slučaju filtara manjih dimenzija) mogu sadržavati bitne detalje slike, pa nam proces ubrzan povećavanjem koraka ne bi značio ništa ukoliko je krajnji rezultat nepovoljan, te je stoga potrebno naći kompromis između veličine koraka i dimenzija filtra. Za korak veličine 2, za primjer za slike, izlazna mapa karakteristika bi bila dimenzija 2×2. Opšta formula po kojoj se računa veličina mape karakteristika za kvadratne filtre i ulazne slike glasi:

$$O = \frac{W - K}{S} + 1, \tag{24}$$

gdje su K veličina filtra, S korak i W širina ili visina ulazne slike. U velikom broju slučajeva, konvolucione neuralne mreže, pored konvolucionih slojeva, sadrže i slojeve sažimanja (eng. *pooling layers*). U kojoj mjeri i na kom mjestu zavisi od arhitekture mreže i njene namjene. Slično povećanju koraka, njihova uloga se prvenstveno ogleda u smanjenju dimenzionalnosti mape karakteristika, dok se među ostalim razlozima izdvajaju kontrola pretreniranja (eng. *overfitting*), slučaj u kome se mreža previše prilagođava ulaznim podacima i gubi sposobnost generalizacije, ubrzanje treninga i translacijska invarijantnost, čime se osigurava da se model bolje nosi s promjenama pozicije objekata na slici [52].

Dvije najzastupljenije operacije sažimanja, sažimanje maksimalnom vrijednošću (eng. *max pooling*) i sažimanje usrednjavanjem (eng. *average pooling*) su prikazane Slikom 15. Segment ulazne slike je dimenzija 4×4, korak $S = 2$ i dimenzije dobijenih mapa su, saglasno (24), 2×2. Filtri u slojevima sažimanja ne sadrže težinske vrijednosti, već se u svakom prozoru ulazne matrice primjenuje funkcija poput



Slika 15: Operacije sažimanja maksimumom i usrednjavanjem sa korakom 2

maksimuma ili prosječne vrijednosti, i dobijeni broj upisuje na odgovarajuće mjesto u mapi. Uprkos prethodno nabrojanim prednostima, ovaj postupak iziskuje određenu dozu opreza. Gubitak informacija usljed izbora samo najveće vrijednosti piksela, ili zamagljivanje značajnih informacija u slučaju usrednjavanja, mogu prouzrokovati probleme dublje u mreži koji će se negativno odraziti na konačni izlaz mreže.

Koristan koncept prenesen iz oblasti obrade signala u CNN jeste dopuna (eng. *padding*). U obradi signala, dopuna se koristi kako bi se izbjeglo „curenje" informacija na ivicama signala, koje filtar ne može u potpunosti da pokrije, ili vrlo površno. Slična potreba, za boljom obradom vrijednosti na ivicama matrica, javila se u konvolucionim neuralnim mrežama.

Kako bi približili pojam, vratimo se na Sliku 14. Primijetimo da prilikom popune prvog reda mape karakteristika, prozor filtra tri puta pokriva vrijednosti piksela u trećoj koloni, dok su pikseli prve i pete kolone pokriveni samo jednom. Dopunom se vrijednosti originalne matrice ne mijenjaju, već samo dodaju dodatni redovi i kolone nula oko originalne matrice (eng. *zero padding*). U konkretnom primjeru sa slike, zadržavanjem istog koraka i veličine filtra, a dopunom originalne matrice do dimenzija 7×7, usaglasili bi veličine originalne matrice i izlazne mape karakteristika, što ponekad olakšava obradu u dubljim slojevima mreže. Osim oivičavanja nulama, u upotrebi su i druge vrste dopune, poput dopunjavanja repliciranjem (eng. *replicate padding*), gdje se vrijednosti sa ivica matrice ponavljaju van matrice.

Na izlazu, nakon slojeva za ekstrakciju i sažimanje, obično se koristi potpuno povezani sloj (eng. *fully connected layer*). Svaki neuron u potpuno povezanom sloju je povezan sa svakim neuronom u prethodnom sloju. Glavna funkcija, a ujedno i razlog zašto se često koristi kao posljedni sloj mreže, jeste donošenje odluka na osnovu karakteristika izdvojenih iz svih prethodnih slojeva. U predloženoj arhitekturi, posljednji sloj je potpuno povezani sloj, a broj neurona u njemu odgovara broju bitova vodenog žiga, pri čemu svaki neuron generiše vjerovatnoću za određenu vrijednost bita.

4.3 Funkcije troška

Kroz prethodna dva potpoglavlja smo vidjeli da, kroz niz konvolucionih, aktivacionih i slojeva sažimanja, konvolucione neuralne mreže izvlače i apstrahuju bitne karakteristike i detalje ulaznih slika, što za cilj ima sposobonost mreže da prepoznaje sve složenije obrasce. Konvolucione neuralne mreže određenu vještinu stiču učenjem. Ovaj postupak inherentno prate greške koje je potrebno uočiti i ispraviti kako bi postigli željeni rezultat. Upravo to je uloga funkcija troška u CNN - usmjeravanje procesa učenja identifikujući greške i omogućavanje mreži da nauči iz njih. Preciznije, funkcije troška omogućavaju mreži da procijeni koliko se trenutne vrijednosti razlikuju od željenih i na koji način treba prilagoditi parametre mreže kako bi se rezultati poboljšali.

Obučavanje neuralne mreže može prerasti u zahtijevan zadatak usljed brojnih problema na koje se može naići. Na početku je potrebno obezbijediti odgovarajući skup podataka, koji se dalje dijeli na tri dijela - trening set, koji se koristi za obuku, validacioni set, koji se koristi za podešavanje hiperparametara i praćenje napretka modela, i test set, koji služi za konačnu evaluaciju performansi modela. Veći broj ulaznih podataka i raznolikost u njima gotovo uvijek znači bolje performanse mreže. To će doprinijeti prevazilaženju dva najveća problema kada je riječ o treningu neuralne mreža, a to su već pomenuto preprilagođavanje i nedovoljno prilagođavanje (eng. *underfitting*).

Do preprilagođavanja dolazi kada model previše dobro nauči specifične detalje i karakteristike iz trening seta, gubeći sposobnost generalizacije nad ukupnim skupom podataka. Ovo dovodi do loših performansi na novim, neviđenim podacima. Uzroka je nekoliko. Prvi možemo tražiti u prevelikom broju parametara, kada složenost modela i količina dostupnih podataka nisu srazmjerni, pa model počinje da uči beznačajne detalje svojstvene određenoj slici. Drugi bitan faktor jeste broj epoha, odnosno koliko puta model prolazi kroz cijeli skup podataka. Veliki broj epoha često vodi preprilagođavanju, jer model, nakon nekog vremena, umjesto generalnih obrazaca, počinje da uočava nebitne šumove i anomalije. Nedovoljno prilagođavanje, sa druge strane, se javlja usljed nepotpunog i oskudnog seta ulaznih podataka, ili kada je model previše pojednostavljen, sa malim brojem parametara, i nije u stanju da se nosi sa promjenama i varijacijama unutar ulaznih slika.

Praćenjem vrijednosti funkcije troška, možemo utvrditi da li model radi ispravno, ili teži jednoj ili drugoj ekstremnoj vrijednosti (Slika 16). Veća vrijednost funkcije troška znači nepodudaranje stvarnih i željenih vrijednosti, i obrnuto. Kada se model previše prilagođava podacima, funkcija troška na trening skupu podataka opada, dok na validacionom skupu počinje da raste poslije određenog broja epoha. U slučaju

nedovoljnog prilagođavanja, funkcija troška na skupu za trening je visoka i sporo konvergira, što sugeriše da mreža nema dovoljnu složenost da adekvatno modeluje ulazne podatke.

Izbor funkcije troška zavisi od prirode problema koji se rješava, strukture ulaznih podataka, i zadatih ciljeva koje model treba da ispuni. Ovdje ćemo napraviti podjelu prema tipu problema, odnosno, da li se radi o regresiji ili klasifikaciji. Klasifikacija podrazumijeva učenje modela iz skupa označenih podataka, gdje svaka instanca podataka pripada jednoj od unaprijed definisanih klasa. Najčešće korišćena funkcija troška za probleme klasifikacije je binarna unakrsna entropija (eng. *binary cross-entropy*) i definiše se kao [53]:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right], \tag{25}$$

gdje je N broj instanci u skupu podataka, y_i stvarna klasa za i-tu instancu, koja može biti 0 ili 1, dok je p_i predviđena vjerovatnoća da i-ta instanca pripada klasi 1. Za višeklasnu klasifikaciju, funkciju generalizujemo kao:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}), \tag{26}$$

pri čemu je C broj klasa, y_{ij} indikator da li instanca i pripada klasi j, a p_{ij} vjerovatnoća klase j.

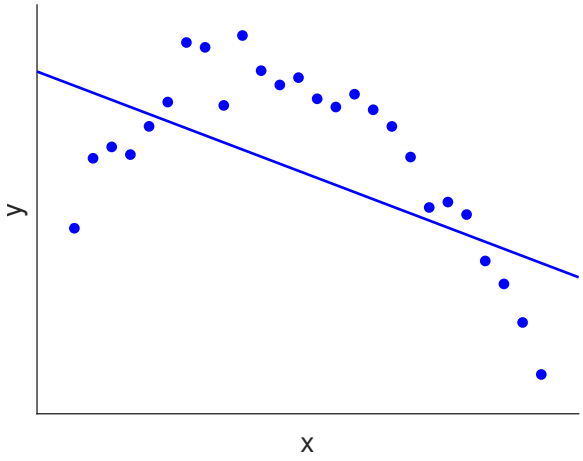
U regresionim zadacima, model se trenira da nauči odnos između ulaznih podataka i odgovarajuće numeričke vrijednosti koju treba predvidjeti. Funkcija troška koja se najčešće koristi u ovim zadacima je srednja kvadratna greška (eng. *mean squared error* - MSE). Data je kao:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{27}$$

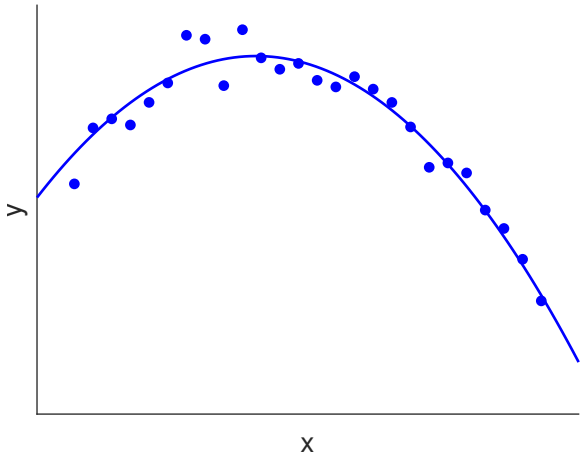
gdje je y_i stvarna vrijednost, $\hat{y}_i$ izlaz modela i N veličina skupa. MSE široku upotrebu pronalazi u jednostavnosti i sposobnosti da kažnjava veće greške više od manjih. Funkcija izvedena iz srednje kvadratne greške je RMSE (eng. *root mean squared error* - RMSE), koja predstavlja kvadratni korijen srednje kvadratne greške. RMSE zadržava istu metriku kao i originalni podaci, čime omogućava intuitivniju interpretaciju greške u odnosu na MSE, koja koristi kvadratnu skalu.

Pored srednje kvadratne greške, u regresiji se često koristi i srednja apsolutna greška (eng. *mean absolute error* - MAE). Računa se na sljedeći način:

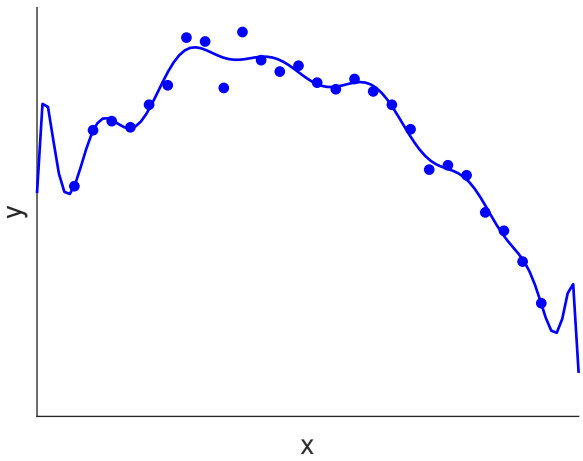
$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|. \tag{28}$$



(a) Loše prilagođen



(b) Dobro prilagođen



(c) Previše prilagođen

Slika 16: Prilagođenost modela ulaznim podacima

MAE je manje osjetljiva na ekstremna odstupanja (eng. *outliers*) u poređenju sa srednjom kvadratnom greškom. Ako se u ulaznim podacima očekuju značajnija odstu-

panja, MAE može pružiti bolju ocjenu modela jer svaki izuzetak ima proporcionalan uticaj na ukupnu grešku. U nekim realizacijama kombinovanje više funkcija troška može biti korisno, jer se nastoji preuzeti najbolje osobine od odabranih funkcija.

4.3.1 Gradijentni spust

Adekvatan izbor funkcije troška i pravilno tumačenje rezultata funkcije od velikog je značaja jer pomaže da ocjenimo koliko i da li model radi dobro na zadatom problemu. Međutim, sama funkcija troška nije dovoljna. Potrebno je optimizovati parametre modela kako bi se ova greška minimizovala. Osnovna i najčešće korišćena metoda optimizacije u dubokom učenju je gradijentni spust (eng. *gradient descent*). Gradijent je vektor koji sadrži parcijalne izvode funkcije u odnosu na sve njene promjenljive. Za funkciju $L(\theta)$, gdje je θ vektor težina, gradijent je definisan kao:

$$\nabla L(\theta) = \left[\frac{\partial L}{\partial \theta_1}, \frac{\partial L}{\partial \theta_2}, \ldots, \frac{\partial L}{\partial \theta_n} \right]. \tag{29}$$

U kontekstu optimizacije, gradijent funkcije troška u odnosu na težine mreže pokazuje pravac najbržeg porasta funkcije. Nasuprot tome, negativni gradijent pokazuje pravac najbržeg opadanja funkcije. Upravo to predstavlja osnovnu zamisao iza ovog optimizacionog algoritma - iterativno ažuriranje težina u pravcu negativnog gradijenta funkcije troška. Proces se ponavlja sve dok se ne pronađe globalni minimum funkcije ili ne ispuni neki drugi kriterijum zaustavljanja, poput maksimalnog broja iteracija. Ažuriranje težina se vrši prema sljedećoj formuli:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t). \tag{30}$$

Hiperparametar η se naziva stopom učenja (eng. *learning rate*) i njime se određuje veličina koraka prilikom ažuriranja. Odabir odgovarajuće stope učenja umnogome diktira proces obuke modela. Mala vrijednost koraka će usporiti proces konvergencije, dok veća vrijednost koraka može uzrokovati preskakanje globalnog minimuma ili, u nekim slučajevima, divergirajuće ponašanje. Korak se najčešće podešava eksperimentalno, uzimanjem početne vrijednosti i posmatranjem njegovog uticaja na funkciju troška tokom obuke. Adaptivni optimizatori, koji su bazirani na gradijentnom spustu, dinamički prilagođavaju stope učenja za svaki parametar tokom treniranja, eliminišući potrebu za ručnim podešavanjem. O njima će biti detaljnije riječi u Sekciji 4.5.

4.4 Propagacija unazad

Tokom propagacije unaprijed (eng. *forward propagation*), ulazni podaci prolaze kroz slojeve mreže, u kojima se vrši njihova transformacija, sve dok se na kraju ne proizvede konačan izlaz. Nakon ovoga, pošto se izračuna funkcija troška, ona se propagira unazad kroz mrežu (eng. *backpropagation*), mjereći doprinos svakog neurona ukupnoj grešci. Ove mjere se koriste kako bi se pronašle optimalne vrijednosti mrežnih parametara, i minimizovala funkcija troška. U ove svrhe se koristi tehnika gradijentnog spusta. Naizmjeničnim propagacijama unaprijed i unazad, postiže se optimalna vrijednost težina i slobodnog člana, čime se minimizuje funckija troška i nastoji dobiti očekivan rezultat na izlazu. Algoritam propagacije unazad je jedan od najvažnijih algoritama u mašinskom učenju, te će stoga biti detaljnije izložen u nastavku.

Krenimo od ulaza mreže i propagacije unaprijed. Kako bi pojednostavili matematiku i približili ideju iza algoritma, uzmimo da se mreža sastoji od L slojeva sa po jednim neuronom u svakom sloju. Težinska suma neurona u posljednjem sloju određena je težinom w^L, aktivacijom neurona u pretposljednjem sloju a^{L-1} i slobodnim članom b^L. Zapišimo to kao [54]:

$$z^L = w^L a^{L-1} + b^L. \tag{31}$$

Predikciju, odnosno izlaz mreže, dobijamo primjenom aktivacione funkcije f nad prethodno izračunatom sumom:

$$\hat{y} = a^L = f(z^L). \tag{32}$$

Na početku smo rekli da je cilj algoritma uvid u to koliko promjena u parametrima mreže utiče na promjenu funkcije troška. Kada je riječ o uticaju promjene jedne varijable na drugu, jasno je da se radi o izvodu. Međutim, iz relacija (31) i (32) uočavamo da će promjena w^L uticati na promjenu z^L, dok će promjena z^L uticati na promjenu a^L. Dakle, koristićemo pravilo izvoda složene funkcije (eng. *chain rule*), krećući se unazad za pronalazak gradijenta funkcije troška u odnosu na težine. Matematički zapis prethodno rečenog je:

$$\frac{\partial J}{\partial w^L} = \frac{\partial J}{\partial a^L}\frac{\partial a^L}{\partial z^L}\frac{\partial z^L}{\partial w^L}. \tag{33}$$

Slično, za promjenu funkcije troška u odnosu na slobodni član, do promjene dolazi samo u posljednjem članu jednakosti:

$$\frac{\partial J}{\partial w^L} = \frac{\partial J}{\partial a^L}\frac{\partial a^L}{\partial z^L}\frac{\partial z^L}{\partial b^L}. \tag{34}$$

Ovom logikom se možemo voditi dalje u slojevima, i vidjeti na koji način parametri iz slojeva manje dubine utiču na funkciju troška. Pošto se slojevi sastoje od velikog broja neurona, parametri i neuroni se smještaju u vektore i matričnim operacijama se vrše računanja, dok jednačine ostaju nepromijenjene, proširene indeksima kojima se neuroni označavaju pojedinačno. Ovaj proces se ponavlja sve dok se kroz određen broj epoha ne pronađu optimalni parametri mreže.

4.5 Optimizacioni algoritmi

Razmatranje optimizacionih algoritama započeto je gradijentnim spustom, kao osnovnim algoritmom neophodnim za razumijevanje koncepta propagacije unazad. Moglo bi se reći da svi ostali algoritmi optimizacije koji se koriste u dubokom učenju predstavljaju gradijentni spust sa varijacijama i poboljšanjima u određenim segmentima. U nastavku ćemo izdvojiti samo one najbitnije.

Prvi je mini-skupni (eng. *mini-batch*) gradijentni spust. Ovakva formulacija je rijetkost u literaturi, obzirom da je tuđica *batch* odomaćena među AI zajednicom, i odnosi se na podskup podataka koji se koristi u jednoj iteraciji treninga. Da ne bi došlo do zabune, bitno je napraviti distinkciju između epohe i iteracije, ne posmatrajući ih kao sinonime. Epoha se odnosi na puni prolazak kroz skup podataka, dok iteracije predstavljaju broj prolazaka kroz podskupove. Na primjer, za ulazni skup veličine 1000 i veličinu podskupa 100, potrebno je 10 iteracija da bi se kompletirala jedna epoha. Za razliku od klasičnog gradijentnog spusta, koji prolazi kroz cijeli skup podataka, računa gradijente, i nakon toga ažurira parametre, mini-skupni spust ulazni skup podataka dijeli u manje podskupove, pri čemu se ažuriranje parametara vrši nakon svakog podskupa, tj. nakon svake iteracije. Ustaljena je praksa da se za veličinu podskupa uzimaju stepeni broja 2, zavisno od veličine ulaznog skupa.

Pristup sličan mini-skupnom je stohastički gradijentni spust (eng. *stochastic gradient descent*). U ovoj verziji algoritma, podskup ima veličinu jednog uzorka, što znači da bi za veličinu ulaznog skupa 1000 i broj epoha 5, broj iteracija bio 5000. Obje modifikacije izvornog algoritma pokazuju bolje rezultate na velikim skupovima podataka, što predstavlja čest slučaj u praksi.

Danas, u upotrebi su dominantno algoritmi koji nemaju fiksnu vrijednost koraka, odnosno algoritmi sa promjenljivim korakom u različitim fazama treninga. Glavni razlog tome je brža konvergencija, koja se postiže uzimanjem veće vrijednosti koraka u početnim fazama treninga, dok se, kako se približavamo minimumu, korak smanjuje, kako ne bi došlo do preskakanja. Konceptom momentuma, koji je preuzet iz fizike, akumulira se brzina na bazi prethodnih gradijenata. Kada se ažuriraju parametri,

koristi se ova akumulirana brzina da bi se pomjerili prema globalnom minimumu [55]. U standardnom gradijentnom spustu, gradijenti se ažuriraju prema relaciji (30). U gradijentnom spustu sa momentumom, trenutna brzina υ_t se računa na osnovu prethodne:

$$\upsilon_t = \gamma\upsilon_{t-1} + \eta\nabla L(\theta). \tag{35}$$

Parametri se zatim ažuriraju na osnovu trenutne brzine:

$$\theta = \theta - \eta\upsilon_t. \tag{36}$$

Hiperparametar γ je koeficijent trenja, i najčešće kao početnu vrijednost uzima 0.9. Veličina koraka se uvećava ukoliko gradijenti pokazuju u istom smjeru, dok se za promjenljive gradijente umanjuje. Analogija sa loptom koja se kotrlja niz brdo velikom brzinom doprinosi shvatanju koncepta.

Modifikacija ovog pristupa, Nesterovljev gradijentni spust, gradijente računa na osnovu buduće pozicije parametara [56]:

$$\upsilon_t = \gamma\upsilon_{t-1} + \eta\nabla L(\theta - \gamma\upsilon_{t-1}). \tag{37}$$

Ažuriranje parametara se vrši na isti način kao u (36). Ako se vratimo analogiji sa brdom i loptom, Nesterovljevim pristupom se nastoji lopta učiniti više „svjesnom" neposrednog okruženja, usporavajući prije nego li dođe do dna brda ili nove kosine.

AdaGrad algoritmom se stopa učenja prilagođava za svaki parametar pojedinačno, uzimajući u obzir istoriju njegovih gradijenata [57]. Ovaj algoritam računa i čuva sumu kvadrata za svaki parametar. Ako se neki parametar često ažurira, što znači da ima veće gradijente, njegova akumulirana suma kvadrata gradijenata će biti veća. To će rezultirati manjom stopom učenja za taj parametar. S druge strane, parametri koji se rjeđe ažuriraju će imati veću prilagođenu stopu učenja, jer će njihova akumulirana suma kvadrata gradijenata biti manja. Glavni nedostatak ove tehnike predstavlja to što stopa učenja, poslije nekog vremena, može postati suviše mala zbog neprekidnog akumuliranja gradijenata, što u kasnijim fazama može usporiti ili zaustaviti proces učenja. Ovaj problem je adresiran RMSprop algoritmom [58], smanjujući efekat gradijenata u „dalekoj prošlosti" i potencirajući uticaj novijih gradijenata. Na ovaj način se stopa učenja održava relevantnom kroz cijeli proces treninga.

Trenutno najzastupljeniji optimizacioni algoritam u dubokom učenju je Adam [59]. Adam kombinuje ideje iz RMSprop-a sa momentumom, po čemu je i dobio ime (ADAptive Moment estimation). Za razliku od RMSprop-a, koji čuva informacije o sumi kvadrata gradijenata svakog parametra, Adam dodatno uključuje i informaciju o prosječnim vrijednostima gradijenata svakog parametra, odnosno brzini njihove promjene, što asocira na momentum.

4.6 Tehnike unaprijeđenja performansi modela

Postizanje visokih performansi modela koji zadovoljavaju unaprijed definisana očekivanja ili nadmašuju postojeća rješenja u određenoj oblasti, nekada može da se svede na niz manjih korekcija u unaprijed osmišljenoj i postavljenoj arhitekturi, koje kumulativno doprinose konačnom uspjehu. Metode koje se dokažu kroz praksu i pokažu korisnim u velikom broju realizacija obično postanu standard i samo jedan u nizu koraka koje treba sprovesti kako bi se šanse za neuspjeh svele na minimum. Regularizacionim tehnikama koje će u nastavku biti diskutovane se nastoji dodatno poboljšati proces treninga i doprinijeti boljoj sposobnosti generalizacije modela. Normalizacija, sa druge strane, za cilj ima stabilnost i ubrzanje obučavanja mreže kroz skaliranje podataka i aktivacija. U literaturi se, zavisno od autora, termini regularizacije i normalizacije često preklapaju ili svrstavaju u isti koš, te se u drugim izvorima može sresti drugačija klasifikacija.

4.6.1 Regularizacija

Regularizacija se koristi za prevenciju prekomjernog prilagođavanja modela trening podacima. Problem preprilagođavanja i uzroci njegove pojave diskutovani su u podsekciji 4.3, kao dio sveobuhvatne priče o funkciji troška. Raznolika struktura ulaznih podataka i optimalno vrijeme treninga značajno doprinose, ali nisu garant da do pojave problema neće doći. Isto se, s pravom, može reći za regularizaciju, ali riječ je o svođenju šansi na domen puke teorije. Učenje napamet nigdje nije poželjno, pa ni u dubokom učenju, jer se teži generalizaciji, odnosno obučiti model da naučeno primjeni na novim podacima.

L1 regularizacijom (eng. *lasso regression*) se originalnoj funkciji troška dodaje dodatni član, suma apsolutnih vrijednosti težinskih koeficijenata, kojim se „kažnjavaju" velike vrijednosti težinskih koeficijenata. Bez ovog člana, model može postati previše složen i prilagodljiv ulaznim podacima, što dovodi do prekomjernog prilagođavanja šumu i specifičnim karakteristikama koje umanjuju njegovu efikasnost. Sa dodatkom kazne, model se teži načiniti jednostavnijim i više sposobnim da ignoriše karakteristike od manjeg značaja. Ova tehnika regularizacije vrijednosti težina, koje ne doprinose krajnjem pozitivnom učinku, postavlja na nulu. Ako kao funkciju troška uzmemo MSE, onda ukupnu funkciju troška sa L1 regularizacijom definišemo kao:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{M}|\theta_i|. \tag{38}$$

Hiperparametrom λ se diktira oštrina kazne. Veća vrijednost λ će dovesti do značajnijeg pojednostavljivanja modela, ali treba biti oprezan kako spriječavanjem

jednog ekstrema ne bi proizveli drugi - *underfitting*. Za manje vrijednosti λ, regularizacija gubi smisao, te stoga treba tražiti kompromis.

L2 regularizacija (eng. *ridge regression*) kao kaznu originalnoj funkciji troška dodaje sumu kvadrata težinskih vrijednosti. Slično kao i L1 regularizacija, i ovom tehnikom se smanjuju vrijednosti težinskih koeficijenata. Međutim, umjesto da ih postavlja na nulu, L2 regularizacija smanjuje koeficijente na dovoljno male vrijednosti, različite od nule. Matematički se predstavlja kao:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{M}{\theta_i}^2. \tag{39}$$

Sve prethodno rečeno kod L1 regularizacije u pogledu hiperparametra λ važi i kod L2. Pravilnim podešavanjem λ postiže se ravnoteža između prilagodljivosti modela i njegove sposobnosti da generalizuje.

Dropout je regularizaciona tehnika novijeg datuma predstavljena u [60]. Kada neuroni u određenom sloju nauče specifične karakteristike zajedno, to može dovesti do toga da model previše zavisi od određenih kombinacija između neurona, što povećava rizik od preprilagođavanja. Osnovna ideja iza *dropout*-a je nasumično isključivanje neurona u procesu treninga, čime se mreža primorava da nauči generalnije obrasce koji nisu zavisni od specifičnih neurona. Napravimo paralelu sa sportom zarad boljeg razumijevanja. Zamislimo ekipu koja se priprema za važan turnir. Trener na treningu uigrava sve igrače sa specifičnom ulogom u timu. Ukoliko trener odluči da na svakom narednom treningu izostavi nekoliko igrača iz tima, to će značiti da različiti igrači moraju preuzimati različite uloge od onih na koje su prethodno navikli. Na taj način, tim mora naučiti da funkcioniše efikasno, bez obzira na to ko je prisutan na terenu.

Dakle, tokom svake iteracije treninga, neuron ima određeni procenat šanse da bude isključen, što znači da neće učestvovati u propagaciji unaprijed i unazad. To se reguliše hiperparametrom p, koji se naziva stopa isključivanja (eng. *dropout rate*). Ukoliko bi postavili *dropout rate* $p = 0.2$, to bi značilo da neuron ima 20% šanse da bude isključen. Tokom testiranja, izlazi neurona se skaliraju sa $(1 - p)$, čime se osigurava da ukupna aktivnost neurona tokom treninga i testiranja bude konzistentna.

Jedna od tehnika regularizacije je i augmentacija podataka (eng. *data augmentation*). Augmentacija podataka omogućava vještačko proširivanje skupa podataka za obuku kroz različite transformacije postojećih uzoraka. Uobičajene transformacije uključuju rotaciju, skaliranje, isjecanje i dodavanje šuma. Slika 17 sadrži primjere iz korpusa podataka korištenog u obuci modela, nakon primjene različitih modifikacija poput rotacija, dodavanja šuma i promjene osvjetljenja, respektivno.

(a) Rotacija (b) So i biber šum (c) Osvjetljenje

Slika 17: Primjena augmentacije podataka

Popularna tehnika regularizacije je rano zaustavljanje (eng. *early stopping*). Podrazumijeva zaustavljanje treninga kada performanse na validacionom skupu podataka prestanu da se poboljšavaju. Kao mjera performansi se najčešće koristi funkcija troška. U praksi se realizuje postavljanjem određenog broj epoha (eng. *patience epochs*), tokom kojih model na validacionom skupu može stagnirati ili pogoršavati se, prije nego treniranje bude prekinuto. Ovim pristupom se smanjuju šanse za preuranjeni prekid treninga.

4.6.2 Normalizacija

Normalizacija se bavi optimizacijom samog procesa treniranja kako bi se postigla stabilnost i efikasnost u obučavanju neuralne mreže. Normalizacija po seriji (eng. *batch normalization*) i skaliranje na bazi minimuma i maksimuma (eng. *min-max scaling*) su najzastupljenije normalizacione tehnike, a isto tako su dio arhitekture umetača vodenog žiga opisanog u Sekciji 5.1, te će u nastavku biti nešto više riječi o njima.

Minimum-maksimum skaliranje je dominantno korišćena tehnika normalizacije u oblasti dubokog učenja prilikom pripreme podataka za trening, prvenstveno zbog jednostavnosti i efikasnosti. Ovom metodom se podaci transformišu tako da sve vrijednosti budu svedene na unaprijed definisani interval, obično između 0 i 1. Ovo se radi kako bi se izbjeglo da karakteristike sa većim numeričkim vrijednostima dominiraju nad onima sa manjim. Budući da pikseli digitalnih slika uzimaju vrijednosti od 0 do 255, skaliranje na opseg [0, 1] osigurava numeričku stabilnost optimizacionih algoritama poput Adama, dok proporcionalne razlike između vrijednosti piksela ostaju nepromijenjene. Matematički se definiše kao:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{40}$$

gdje je x' skalirana vrijednost, x originalna vrijednost, a x_{max} i x_{min} maksimalna, odnosno minimalna vrijednost u skupu podataka. Ukoliko se u podacima očekuje prisustvo većeg broja izuzetaka, treba izbjegavati upotrebu ove tehnike, jer će ta ekstremna vrijednost učestvovati u oblikovanju novog opsega.

Kako je ranije istaknuto, normalizacija po seriji doprinosi bržoj konvergenciji i ubrzava trening modela. Širu prepoznatljivost i primjenu ova tehnika je stekla kao dio ResNet arhitekture, jednog od najuticajnih i najpoznatijih radova u oblasti dubokih neuralnih mreža [61]. Za razliku od prethodne, gdje se podaci pripremaju prije pokretanja obuke, ova tehnika se koristi unutar modela. U inicijalnom radu gdje je ova metoda opisana [62], autori prednosti implementacije objašnjavaju smanjivanjem unutrašnjeg kovarijantnog pomjeranja (eng. *internal covariate shift*). Unutrašnje kovarijantno pomjeranje označava promjenu distribucije aktivacija slojeva unutar mreže tokom treninga, usljed stalnog ažuriranja parametara u prethodnim slojevima. Ovo otežava optimizaciju, jer model mora neprestano da se prilagođava novim, dinamičkim okolnostima pri učenju. Normalizacija po seriji se definiše sljedećom jednakošću:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \tag{41}$$

gdje je x_i originalna aktivacija ili vrijednost ulaza za i-ti uzorak unutar trenutnog skupa, μ_B srednja vrijednost svih aktivacija x_i, σ_B^2 varijansa svih aktivacija unutar trenutne serije i ε konstanta male vrijednosti kojom se spriječava dijeljenje sa nulom. Na ovaj način se srednja vrijednost i varijansa postavljaju na 0 i 1, što odgovara standardnoj Gausovoj raspodjeli. Konačna izlazna vrijednost nakon serijske normalizacije za i-ti uzorak je:

$$y_i = \gamma \hat{x}_i + \beta. \tag{42}$$

Parametri γ i β, koji označavaju skaliranje i pomak, omogućavaju mreži učenje optimalne distribucije nakon normalizacije. Iako je prethodno rečeno da osnovna normalizacije postavlja srednju vrijednost i varijansu aktivacija na 0 i 1, parametrima se modelu pružaju mogućnosti da prilagodi raspon i pomak normalizovanih vrijednosti prema potrebama specifičnog zadatka.

U [63] se tvrdi da prednosti ove regularizacione tehnike nisu vezane isključivo za suzbijanje kovarijatnog pomjeranja, već da normalizacija po seriji poboljšava pejzaž optimizacije (krive funkcije troška i gradijentnog spusta), čineći ga više glatkim i time olakšava učenje i ubrzava konvergenciju.

5 Predlog rješenja

U ovom poglavlju biće predstavljen predlog sistema za umetanje polu-krhkog vodenog žiga korištenjem konvolucionih neuralnih mreža. Sistem je testiran kroz performanse koje postiže pri umetanju vodenog žiga, ekstrakciji bitova na strani prijema, kao i pri dodavanju određenih šumova i smetnji kako bi se ispitala efikasnost polu-krhkog pristupa. U procesu istraživanja, korišteno je nekoliko različitih funkcija troška i njihove kombinacije s ciljem pronalaska optimalne funkcije koja daje najbolje rezultate za zadati problem. Istraživanje je u potpunosti sprovedeno koristeći *Google Collab*, popularnu platformu baziranu na oblaku koja omogućava korištenje GPU resursa besplatno. Ovo dolazi sa određenim ograničenjima, prvenstveno kada je riječ o dostupnosti resursa, koji su ograničeni, što utiče na veličinu modela i skupove podataka koji se mogu obraditi u određenom vremenskom roku.

Uzimajući u obzir vremenska i resursna ograničenja, izvorni *CelebA* skup podataka je prilagođen potrebama ovog rada [64]. Originalni skup uključuje preko 200 hiljada slika lica različitih poznatih ličnosti, zajedno s odgovarajućim atributima (npr. naočare, oblik kose, brada). Slike su raznovrsne po pitanju držanja, izraza lica i osvjetljenja, što skup čini bogatim i pogodnim za složenije zadatke učenja. Za proces treninga je odabrano 5 hiljada slika, dok je za validaciju i testiranje performansi modela izdvojeno po 500 slika. Sve slike su skalirane na dimenzije 64×64 piksela, čime se značajno smanjuju računski zahtjevi u odnosu na rad sa originalnim slikama većih dimenzija.

Arhitektura se sastoji od ukupno 477.491 parametara, od kojih mreža umetača obuhvata 268.571, a mreža detektora 208.920 parametara, pa bi ovaj sistem mogli opisati kao blago kompleksan. Kao optimizacioni algoritam tokom obuke koristi se Adam, sa konstantnom vrijednošću stope učenja od 0.0001 pri svakom pokretanju treninga.

Predloženi sistem se djelimično zasniva na istraživanju [65], u kojem se odvojene mreže umetača i detektora treniraju zajedno za robusno umetanje vodenog žiga u audio signal. Gradivni elementi arhitekture, umetač i detektor, kao i proces zajedničkog obučavanja tih mreža, će biti detaljno razmotreni u Sekcijama 5.1, 5.2 i 5.3, respektivno. Gausov šum, šum tipa so i biber, kao i Gausov filtar predstavljaju napade na sistem i obrađeni su u Sekciji 5.4.

5.1 Umetač vodenog žiga

Neuralna mreža umetača temelji se na poznatoj U-net arhitekturi [66], koja je inicijalno razvijena za biomedicinsku segmentaciju slika. Kasnije, uvidjevši prednosti

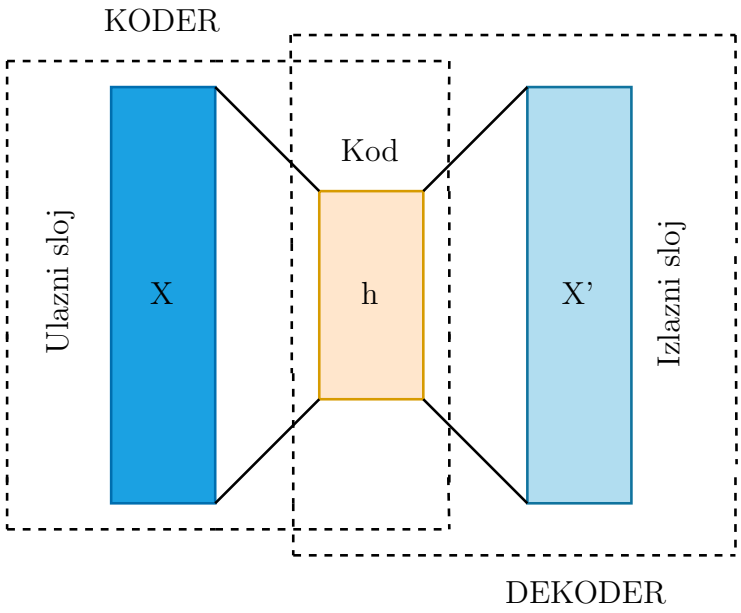
ove arhitekture, naučna zajednica je počela da je koristi u brojnim zadacima, šireći njenu primjenu na širok spektar oblasti. U-net se može posmatrati kao specifičan oblik autoenkodera. Autoenkoder je mreža koja uči da komprimuje podatke u manju, latentnu reprezentaciju i zatim rekonstruiše originalne informacije iz te reprezentacije. Sastoji se od kodera i dekodera (Slika 18). Koder smanjuje dimenzionalnost ulaznih podataka tako da sačuva najvažnije karakteristike ulaznog signala. Sažete informacije ulaznog signala se nazivaju latentnim vektorom ili kodom. S druge strane, dekoder prima latentni vektor i rekonstruiše podatke tako da dobijeni izlaz bude što vjerniji ulazu. U idealnom slučaju, za x na ulazu kodera, izlaz dekodera $\hat{x}$ bi trebao da bude takav da zadovoljava:

$$x = \hat{x}. \tag{43}$$

U praksi, međutim, nekoliko faktora otežava ostvarenje ove relacije. Prvo, proces kodovanja podrazumijeva redukciju dimenzionalnosti, pri čemu se određeni detalji iz ulaznih podataka gube, naročito ako latentni prostor nije dovoljno veliki da zadrži sve informacije. Drugo, dekoder često nije u mogućnosti da precizno rekonstruiše izgubljene informacije zbog ograničenja modela, aproksimacija korišćenih u optimizaciji i regularizacije koja se primjenjuje radi bolje generalizacije.

U-Net predstavlja specifičan tip autoenkodera jer zadovoljava sve osnovne kriterijume autoenkoderske arhitekture, uz nekoliko razlika. Prva od njih su preskačuće veze (eng. *skip connections*). Ideja potiče iz [61], dok se u U-netu primarno koristi zarad očuvanja bitnih karakteristika dajući slojevima dekodera pristup prostornim informacijama iz ranijih slojeva kodera, olakšavajući na taj način proces rekonstrukcije. Druga bitna razlika jeste namjena. Kao što je prethodno rečeno, kod klasičnih autoenkodera kompresija uslovljava pronalazak što kompaktnijeg i minimalnijeg latentnog prostora, dok cilj U-neta nije nužno smanjenje dimenzionalnosti do minimalne reprezentacije, već se fokusira na očuvanje prostornih informacija za preciznu segmentaciju.

Model umetača, prikazan na Slici 19, karakteriše simetrična koder-dekoder struktura u obliku latiničnog slova „U”, po čemu je ova arhitektura i dobila naziv. Takođe, primjetno je da se poruka, odnosno vodeni žig, umeće u latentnom prostoru, središnjem dijelu mreže između kodera i dekodera, koji se naziva usko grlo (eng. *bottleneck*). Paradoksalno, perfektnoj rekonstrukciji slike sa ulaza, u ovom slučaju, ne treba težiti i kao posljedicu bi imala neuspjeh čitavog sistema. Ne treba izgubiti iz vida da je i detektor, koji prima sliku sa vodenim žigom, takođe dio votermarking sistema. Ukoliko se na ulazu detektora pojavi slika identična onoj na ulazu kodera, to bi značilo da je umetač uklonio bitove vodenog žiga, čime bi strana detekcije, kao i čitav sistem, izgubili na smislu. Isto tako, loše sproveden proces umetanja, u



Slika 18: Blok šema autoenkodera

situaciji gdje je vodeni žig vidno primjetan, bio bi povoljan iz ugla detekcije i ubrzao konvergenciju te mreže. Uzimajući u obzir antagonističku prirodu obje mreže, očito je da umetač i detektor ne bi trebalo posmatrati kao odvojene entitete, već kao spregnute djelove unutar jedinstvenog okvira.

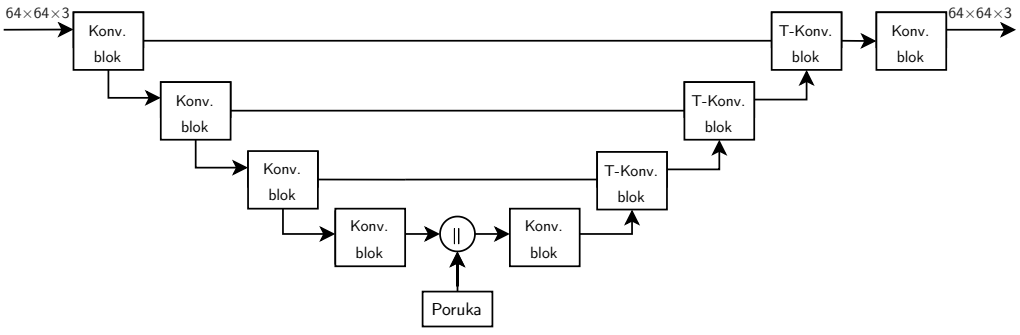
Na ulaz umetača dolazi slika dimenzija $64 \times 64 \times 3$. Prva dva broja označavaju visinu i širinu slike u pikselima, dok 3 predstavlja broj kanala. Slike su u izvornom domenu, dok je jedina predobrada podataka skaliranje na bazi minimuma i maksimuma, čija motivacija za uvođenjem je opisana u Poglavlju 4.6.2. Koder se sastoji od 4 decimaciona bloka, koji vrše sažimanje dimenzionalnosti do odgovarajućeg latentnog prostora u kojem će biti umetnut vodeni žig. Broj filtara u blokovima se progresivno povećava: prvi blok sadrži 8 filtara, drugi 16, treći 32, dok posljednji blok uključuje 64 filtra. Veličina koraka je postavljena na 2 u svim konvolucionim blokovima, izuzev prvog, gdje je veličina koraka 1. Veličina filtra je konzistentna kroz cijelu mrežu umetača i iznosi 5×5. Nad ulaznim podacima se u svakom bloku primijenjuje 2D konvolucija, potom normalizacija po seriji, i na kraju ReLU aktivaciona funkcija.

Dimenzije latentne reprezentacije, prije umetanja vodenog žiga, nakon 4 decimaciona bloka, su $8 \times 8 \times 64$. Vodeni žig je jednodimenzionalna osmobitna binarna sekvenca, koju nije moguće direktno dodati u latentni prostor, već mora biti preuređena tako da dimenziono odgovara tenzoru koji se dobija na izlazu četvrtog decimacionog bloka. U ovom rješenju to je realizovano ponavljanjem bitova vodenog žiga, čime se dodatno osigurava da bitovi umetnute poruke neće biti izgubljeni u

narednim slojevima. Vodeni žig se na latentnu reprezentaciju dodaje operacijom nadovezivanja (eng. *concatenation*). Spajanje se vrši duž treće, kanalne dimenzije, pa nakon umetanja tenzor poprima oblik $8 \times 8 \times 72$. Nakon toga, slijedi još jedan decimacioni blok sa 64 filtra i korakom 1, čija je prisutnost direktno povezana sa nenarušavanjem simetrije U-Net arhitekture, dok veličina koraka osigurava da neće doći do daljeg sažimanja dimenzija.

Dekoder se sastoji od 3 interpolaciona bloka koji vrše rekonstrukciju slike sa umetnutim žigom. Broj filtara opada kako se približavamo izlazu dekodera, odnosno umetača, počevši od 32, zatim 16, i na kraju 8. Veličina koraka u interpolacionim blokovima je konstantna i iznosi 2. Linije koje povezuju blokove kodera i dekodera na Slici 19 su prethodno pomenute preskačuće veze, kojima se izlaz iz pripadajućeg koderskog bloka nadovezuje na simetrični blok na dekoderskoj strani. Redosljed operacija je sličan onom u koderu, s tom razlikom da se umjesto standardne konvolucije koristi transponovana.

Izlaz posljednjeg bloka dekodera ima oblik $64 \times 64 \times 8$, što ne odgovara dimenzijama signala na ulazu u koder. Kako dimenzije na ulazu i izlazu umetača moraju biti iste, ovaj problem se prevazilazi umetanjem dodatnog decimacionog bloka sa 3 filtra i jediničnim korakom. Na ovaj način se broj kanala postavlja na 3, čime se postiže usaglašenost između oblika signala na ulazu i izlazu umetača. Kao aktivaciona funkcija se koristi sigmoid, kojom se izlazne vrijednosti umetača ograničavaju na opseg [0, 1], što je u skladu sa normalizovanim ulaznim slikama. Funkcije troška umetača korištene u istraživanju su srednja apsolutna greška, srednja kvadratna greška i njihove izvedenice. Odnos maksimalni signal-šum (eng. *Peak Signal-to-Noise Ratio* – PSNR) koristi se kao mjera za ocjenu kvaliteta umetanja vodenog žiga. PSNR je najčešće korišćena metrika za procjenu kvaliteta rekonstruisanih signala u odnosu na originalne verzije istih. Visoka vrijednost PSNR-a ukazuje na to da je



Slika 19: Struktura mreže umetača. Simbolom „||" je označena operacija nadovezivanja.

rekonstruisana verzija signala vrlo slična originalu, što znači da je degradacija signala minimalna. Vrijednosti se izražavaju u decibelima (dB). Definiše se kao logaritamski odnos između maksimalne moguće vrijednosti signala (snage) i srednje kvadratne greške između originalnog i rekonstruisanog signala:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right). \tag{44}$$

Visoka vrijednost PSNR-a ne mora nužno značiti dobar vizuelni kvalitet slike, jer metrika ne uzima u obzir subjektivnu percepciju ljudskog oka i njegovu osjetljivost na određene promjene nastale u slici. Iz ovog razloga se PSNR često kombinuje sa drugim mjerama koje uzimaju u obzir ljudsku percepciju vizuelnog sadržaja, i na taj način daje konačna procjena kvaliteta rekonstruisane slike. U predloženom sistemu, dodatna metrika za procjenu kvaliteta umetanja vodenog žiga, pored PSNR-a, je indeks strukturalne sličnosti (eng. *Structural Similarity Index* - SSIM). Ova metoda procjene kvaliteta je prvi put predložena u [67]. SSIM procjenjuje vizuelni uticaj promjena u osvjetljenju, konstrastu i strukturi. Matematički se zapisuje kao:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{45}$$

gdje su:

- μ_x i μ_y srednje vrijednosti piksela u prozorima x i y;
- σ_x^2 i σ_y^2 varijanse piksela u prozorima x i y;
- σ_{xy} kovarijansa piksela između prozora x i y;
- C_1 i C_2 konstante malih vrijednosti koje spriječavaju dijeljenje sa nulom.

SSIM kao rezultat daje vrijednosti u opsegu [-1, 1], pri čemu -1 znači potpuno različite slike, 0 označava izostanak strukturalne sličnost, dok 1 znači da je riječ o identičnim slikama. Najveći nedostatak SSIM-a je njegova kompleksnost računanja.

5.2 Detektor vodenog žiga

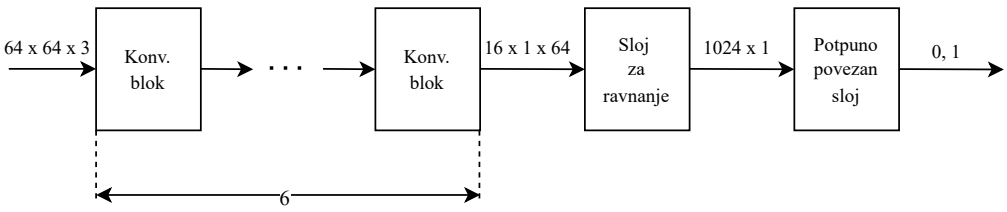
Neuralna mreža detektora za zadatak ima da prepozna i ekstrahuje binarnu sekvencu koja je prethodno dodata u sliku putem mreže umetača. Detekcija se vrši na nivou pojedinačnog bita, nakon čega se vrši rekonstrukcija originalne sekvence. Pošto se na izlazu mreže bitovi vodenog žiga sa određenom vjerovatnoćom klasifikuju kao 0 ili 1, zadatak se može uzeti kao klasifikacioni, te je stoga mreža koncipirana po

uzoru na neuralne mreže za klasifikaciju. Model se sastoji od 6 decimacionih blokova, sloja za ravnanje (eng. *flatten layer*) i potpuno povezanog sloja, kao što je prikazano na Slici 20.

Prvi decimacioni blok prima sliku sa vodenim žigom dimenzija $64 \times 64 \times 3$ i primijenjuje operaciju konvolucije sa 16 filtara veličine 5×5. Kao što je bio slučaj kod umetača, veličina filtra je konstantna i ostaje nepromijenjena kroz cijelu mrežu. Korakom veličine 2 se prostorna dimenzionalnost smanjuje za pola. Drugim blokom istih atributa se dobija tenzor dimenzija $16 \times 16 \times 16$.

Centralni dio mreže sastoji se od dva bloka sa po 32 filtra. Korak dimenzija (1, 2) označava pomjeranje prozora filtra za jedan piksel po visini i dva piksela po širini. Zadržavanje vrijednosti koraka iz prva dva bloka moglo bi ugroziti sposobnost detektora da precizno identifikuje bitove vodenog žiga zbog povećanog rizika od gubitka ključnih detalja. Nasuprot tome, upotreba jediničnog koraka smanjila bi rizik od gubitka informacija i preskakanja značajnih karakteristika, ali bi značajno povećala računske zahtjeve i usporila proces treniranja. Uzimajući prethodno rečeno u obzir, kombinovana vrijednost pomjeraja predstavlja kompromisno rješenje između očuvanja detalja i efikasnosti.

U posljednja dva bloka zadržava se kombinovana vrijednost koraka, dok se broj filtara povećava dvostruko. Takođe, u ovim, i svim prethodnim decimacionim blokovima se nakon svakog konvolucionog sloja sprovodi normalizacija po seriji, praćena ReLU aktivacionom funkcijom, slično postavci umetača. Sloj za ravnanje tenzor oblika $16 \times 1 \times 64$ pretvara u jednodimenzionalni vektor dimenzija 1024×1. Ovo služi kao priprema za potpuno povezani sloj koji daje konačni izlaz mreže. Broj čvorova potpuno povezanog sloja je jednak broju bitova žiga, dok se kao aktivaciona funkcija koristi sigmoid. Izbor sigmoida je uslovljen time da izlazi modela mogu biti direktno interpretirani kao vjerovatnoće da svaki bit pripada klasi 0 ili 1. Kao funkcija troška detektora uzeta je binarna unakrsna entropija, dok će se za procjenu robusnosti detekcije koristiti vjerovatnoća greške po bitu (eng. *Bit Error Rate* - BER), koja daje procenat pogrešno rekonstruisanih bitova vodenog žiga.



Slika 20: Struktura mreže detektora

5.3 Obučavanje sistema

Dizajn mreža za umetanje i detekciju vodenog žiga predstavlja bitan, ali samo još jedan korak u razvoju votermarking sistema. Iako važna, sama arhitektura nije dovoljna da osigura uspjeh sistema ukoliko procedura treninga nije pažljivo osmišljena i sprovedena tako da maksimizira performanse obje neuralne mreže. Kako je ranije istaknuto, suprotstavljeni ciljevi umetača i detektora dodatno komplikuju cijelokupan proces obuke. Odvojeni trening bi, međutim, predstavljao još veći problem. U odsustvu povratne sprege između dvije mreže, umetač je usmjeren ka postizanju što vjernije rekonstrukcije originalnog signala, pri čemu zanemaruje ograničenja detektora u prepoznavanju bitova vodenog žiga koji su potencijalno izgubljeni prilikom procesa rekonstrukcije, čime se ugrožava funkcionisanje kompletnog sistema. Kako bi se obezbijedila usklađenost, umetač i detektor se treniraju istovremeno.

Tokom procesa obuke, gradijenti iz funkcije troška detektora se računaju i propagiraju unazad, prolazeći kroz detektor do umetača, čime se omogućava „komunikacija" između dva modela. Na početku, mreža umetača pokazuje bolje performanse, dok vrijednost funkcije troška detektora ukazuje na nasumično pogađanje bitova žiga. Kako ovo znači konvergenciju samo jedne od mreža, potrebno je naći način kojim će se spriječiti preuranjeno opadanje krive funkcije troška umetača jer to znači sigurnu divergenciju mreže detektora. Uprošteno rečeno, u početnim fazama treninga neophodno je djelimično narušiti proces umetanja žiga kako bi se detektoru omogućilo vrijeme potrebno za prilagođavanje i postizanje stabilne konvergencije.

Balans između dvije mreže se postiže uvođenjem težinskih faktora koji se dinamički prilagođavaju tokom određenih faza treninga. Težinski faktori dodjeljuju odgovarajuće prioritete funkcijama troška umetača i detektora, omogućavajući kontrolu nad njihovim uticajem na ukupni proces učenja. Relacije (46) i (47) opisuju kako se vrijednosti težinskih faktora umetača i detektora mijenjaju u zavisnosti od trenutne epohe x. Tabelom 1 su date vrijednosti parametara i o njihovom izboru će biti više riječi u nastavku.

Parametri M, N i K predstavljaju granične epohe i mogu uzeti neke druge vrijednosti iz skupa prirodnih brojeva. Sa Λ i Δ su označene početne i krajnje vrijednosti težinskih faktora, dok je μ skalirajući koeficijent iz skupa realnih brojeva. Bitno je istaći da postavljene vrijednosti parametara nisu nužno optimalne, niti je moguće garantovati njihovu optimalnost, s obzirom na to da nisu rezultat primjene specifične matematičke formulacije. Vrijednosti su odabrane kroz iterativni proces treninga, uz praćenje obrazaca ponašanja krivih tokom različitih epoha. S obzirom na to da trening u prosjeku traje oko 6 sati, primjena ranog zaustavljanja zahtijeva pažljiv pristup, budući da može navesti na pogrešne zaključke. Manjkavosti u izboru

određenih parametara često postaju očigledne tek u kasnijim fazama obuke, kao što su nagli skokovi ili trend rasta krivih funkcija troška, koje ukazuju na pojavu pretreniranja.

Ako se vratimo na Tabelu 1, u skladu sa prethodnim razmatranjima, tokom prve 3 epohe značaj funkcije troška detektora je 3 puta veći u odnosu na funkciju troška umetača. Nakon treće epohe, ovaj odnos se postepeno preokreće, tako da sa početkom 21. epohe funkcija troška umetača dobija 3 puta veći značaj u odnosu na detektor. Ovaj odnos se zadržava do 100. epohe, odnosno do kraja obuke.

Na Slici 21 je jasno uočljiva potreba za uvođenjem opisanog mehanizma, posmatrajući grešku koju detektori prave u odnosu na grešku mreža umetača. Na primjeru srednje kvadratne greške se vidi da je, pri ovom pokretanju obuke, funkcija troška detektora na početku 7 puta veća u odnosu na trošak umetača, dok već sa prolaskom 15. epohe postiže stabilnu vrijednost koju zadržava do kraja treninga. Umetač, s druge strane, kreće sa mnogo nižom vrijednosti funkcije troška, ali, za razliku od one detektora, kriva troška umetača ima blaži nagib i trend opadanja traje do posljednje epohe. Različite funkcije troška, njihov izbor i uticaj na dobijene rezultate su tema ovog istraživanja i o njima će detaljnije biti riječi u Sekciji 6.

$$u(x) = \begin{cases} \Lambda_u, & x < M \\ \Lambda_u + (x-1)\cdot\mu_u, & M \leq x \leq N \\ \Delta_u, & N < x \leq K, \end{cases} \tag{46}$$

$$d(x) = \begin{cases} \Lambda_d, & x < M \\ \Lambda_d - (x-1)\cdot\mu_d, & M \leq x \leq N \\ \Delta_d, & N < x \leq K. \end{cases} \tag{47}$$

Tabela 1: Vrijednosti parametara korištenih u jednakostima (46) i (47).

Parametar	**Umetač**	**Detektor**
Λ	1.0	3.0
Δ	3.0	1.0
μ	0.1	
M	4	
N	20	
K	100	

5.4 Napadi na sistem

Do sada su analizirani umetač i detektor, uključujući proces zajedničke obuke ovih mreža. Da bi sve komponente opšte šeme umetanja vodenih žigova prikazane na Slici 1 bile u potpunosti pokrivene, potrebno je osvrnuti se i na prenosni kanal - segment kroz koji prolazi slika sa umetnutim žigom prije nego što dospije na ulaz detektora. Već je istaknuto da u prenosnom kanalu može doći do oštećenja slike, iz bilo kog razloga, što može dovesti do potpunog uklanjanja žiga ili do tolikog narušavanja kvaliteta slike da prisustvo žiga postaje sekundarno. Savremeni votermarking sistemi teže da unaprijed identifikuju i integrišu što veći broj potencijalnih napada tokom procesa obuke, kako bi se osigurala robusnost i efikasno neutralisanje ili ublažavanje posljedica tih prijetnji u realnim uslovima primjene. Sa druge strane, broj ovih napada je toliki da je teoretski nemoguće unaprijed predvidjeti sve vrste anomalija do kojih bi moglo doći. Kompromis se nalazi u dizajniranju sistema koji pokrivaju opseg sličnih, ili smetnji koje dijele skup određenih karakteristika. U pojedinim sistemima prioritet se daje otpornosti na napade putem filtriranja, dok se u drugim fokus stavlja na postizanje robusnosti prema različitim vrstama šumova.

Vrste napada korišćenih pri evaluaciji robusnosti vodenog žiga mogu varirati u zavisnosti od istraživačkog pristupa i unaprijed definisanih ciljeva istraživanja. U okviru ovog rada, robusnost detekcije biće ispitana korišćenjem Gausovog šuma, šuma tipa so i biber, kao i primjenom Gausovog filtra.

Gausov šum se modeluje kao slučajna promjenljiva $N(\mu, \sigma^2)$ koja prati Gausovu raspodjelu sa srednjom vrijednošću μ i varijansom σ^2. Njegova gustina vjerovatnoće je oblika:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{48}$$

Ako sliku na izlazu iz umetača označimo sa $W(x, y)$, onda dodavanje Gausovog šuma možemo izraziti kao:

$$W_{\text{noise}}(x, y) = W(x, y) + N(0, \sigma^2), \tag{49}$$

gdje je $N(0, \sigma^2)$ slučajni šum sa nultom srednjom vrijednošću i varijansom σ^2. Razmatrana vrijednost varijanse u predloženoj realizaciji je 0.01.

So i biber je impulsni šum koji se manifestuje nasumičnim crnim i bijelim pikselima rasutim po slici. Naziv potiče iz sličnosti sa zrncima soli (bijele tačke) i bibera (crne tačke). U slučaju oskudnog skupa podataka, može se koristiti i kao tehnike augmentacije podataka (Slika 17). Lokacija i intenzitet šuma su nasumični, i

šum zahvata određeni procenat ukupnih piksela slike. Najčešće nastaje zbog grešaka u prenosu podataka ili kvarova senzora u uređajima za snimanje. Prilikom testiranja robusnosti ovim šumom, vrijednost koja se postavlja je je procenat piksela p koji će biti zamijenjeni crnim ili bijelim pikselima. Za potrebe ovog rada, vrijednost p je postavljena na 0.01.

Gausov niskopropusni filtar je vrsta filtra koji se koristi za smanjenje visokofrekventnih komponenti signala, dok omogućava prolaz niskih frekvencija. Kako je ranije istaknuto, na visokim frekvencijama se nalaze detalji, ali i šumovi, dok niskofrekventne komponente predstavljaju „glatke“ dijelove slike. Iz ovog razloga se u oblasti obradi slike koristi za efekat zamagljivanja (eng. *blurring*). Matematički se opisuje kao:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{50}$$

gdje su x i y prostorne koordinate, a σ standardna devijacija. Prilikom realizacije Gausovog filtra, potrebno je odrediti veličinu filtra, kao i vrijednost standardne devijacije. U okviru ispitivanja, vrijednost standardne devijacije je postavljena na σ=0.5, dok je veličina filtra 5×5.

6 Rezultati

U ovom poglavlju izloženi su rezultati dobijeni prilikom implementacije i evaluacije sistema za umetanje vodenog žiga u digitalne slike. Cilj eksperimentalne evaluacije bio je procjena performansi sistema u pogledu:

- Efikasnosti umetanja i detekcije kroz analizu više funkcija troška umetača i stabilnosti sistema tokom treninga,
- Neprimjetnosti umetnutih vodenih žigova koristeći PSNR i SSIM,
- Otpornosti detekcije na određene napade primjenom BER-a.

U prvom dijelu Poglavlja 6.1 je pružen pregled korišćenih funkcija troška tokom procesa obuke sistema, uz analizu njihovih performansi i uticaja na postignute vrijednosti metrika kojima se ocjenjuje neprimjetnost umetanja. Drugi dio uključuje prikaz slika prije i poslije umetanja sa njihovim histogramima, kao i poređenje postignutih rezultata sa relevantnim istraživanjima iz ove oblasti. Tema Poglavlja 6.2 je robusnost umetnutog vodenog žiga, uz poređenje postignutih vrijednosti BER-a sa rezultatima iz navedenih istraživanja, kako u prisustvu šuma, tako i u idealnim uslovima bez smetnji.

6.1 Izbor funkcije troška i neprimjetnost umetanja

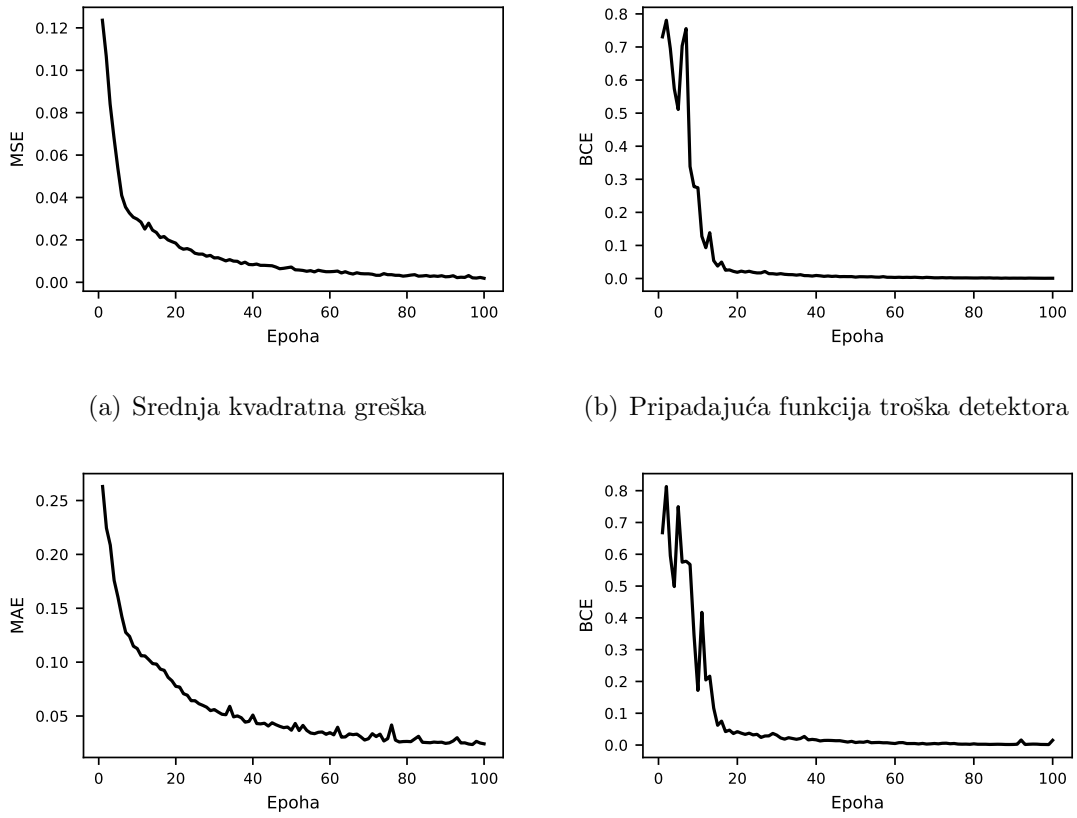
U Tabeli 2 dat je prikaz vrijednosti definisanih metrika performansi sistema u zavisnosti od korištene funkcije troška prilikom obuke. Najbolje performanse ostvarene su korištenjem RMSE, dok je MSE pokazao najslabije rezultate. Srednja apsolutna greška pokazuje performanse najbliže RMSE, dok kombinovana funkcija troška bazirana na dvije osnovne nije dala zadovoljavajuće rezultate. Postoje argumenti za i protiv kada se nastoje objasniti prethodne dvije rečenice. Kao glavnu manu MSE smo prethodno istakli osjetljivost na greške, odnosno na odstupanja u trening skupu podataka, kao što su šumovi i nesavršenosti u slici.

Tabela 2: Uticaj izbora funkcije troška na vrijednosti mjera performansi umetača

Funkcija troška umetača	**PSNR [dB]**	**SSIM**
MAE	30.51	0.9581
MSE	27.27	0.8766
RMSE	31.04	0.9623
$0.5 \times \text{MSE} + 0.5 \times \text{MAE}$	29.41	0.9059

Sa druge strane, ako na Slici 21 pogledamo trend krivih srednje kvadratne i srednje apsolutne greške, vidimo da kriva kvadratne greške pokazuje stabilniji trend opadanja, i na kraju, manju vrijednost greške po završetku 100. epohe. Primjetno je i da je na početku treninga greška umetača koji koristi MAE gotovo dvostruko veća u odnosu na MSE. Kada je riječ o uticaju na funkciju troška detektora, slično se može reći kao za grešku umetača. Kriva detektora sistema čiji umetač koristi MSE ima relativno stabiliniji pad, uz napomenu da oba detektora u početku uzimaju velike vrijednosti greške, da bi u kasnijim fazama treninga gravitirali nuli.

Slike prije i poslije umetanja vodenog žiga, zajedno sa njihovim histogramima, prikazane su na Slikama 22 i 23, respektivno. PSNR prve slike je nešto viši u poređenju sa najboljim rezultatom postignutim nakon treninga pomoću RMSE, dok druga slika ima neznatno nižu vrijednost od prijavljene. Ljudskom vizuelnom sistemu mnogo bliža metrika, SSIM, takođe ne prikazuje značajnije varijacije između slika prije i poslije umetanja, održavajući vrijednosti u rasponu od 0.95 do 0.96.



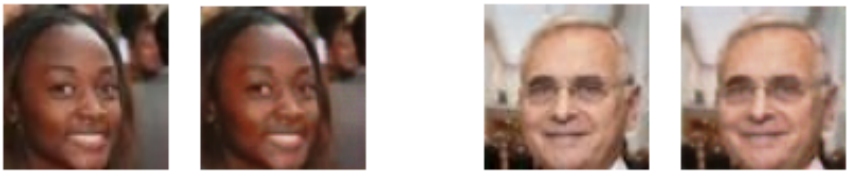
(a) Srednja kvadratna greška

(b) Pripadajuća funkcija troška detektora

(c) Srednja apsolutna greška

(d) Pripadajuća funkcija troška detektora

Slika 21: Uticaj različitih funkcija troška umetača na funkciju troška detektora



(a) PSNR = 31.25 dB, SSIM = 0.960 (b) PSNR = 30.92 dB, SSIM = 0.958

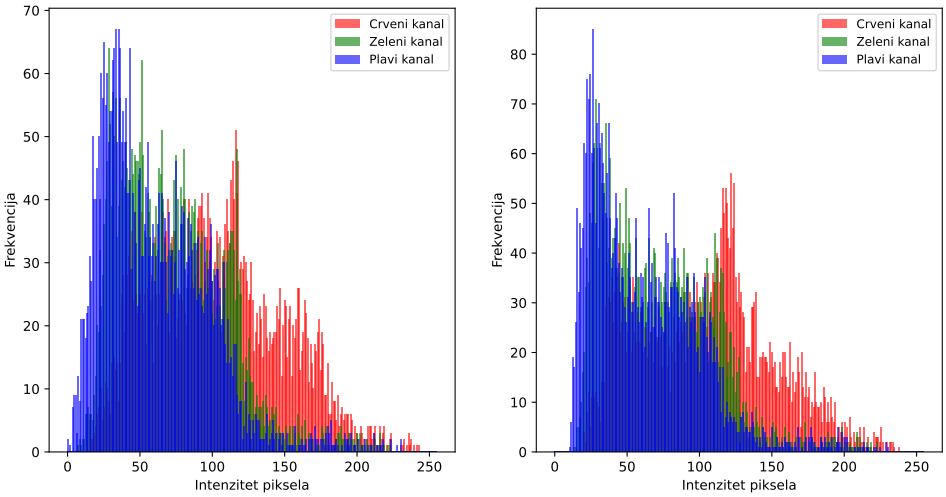
Slika 22: Uporedni prikaz slika prije i poslije umetanja vodenog žiga sa pripadajućim vrijednostima metrika za ocjenu kvaliteta

Histogrami su dodati kao jasniji prikaz na koji način umetnuti bitovi vodenog žiga utiču na originalnu sliku. Pojam histograma je uveden ranije prilikom pregleda jedne od tehnika votermarkinga u prostornom domenu (Sekcija 3.1). Oba histograma zadržavaju svoj osnovni oblik i konture, uz smanjenje visokih vrijednosti osvjetljenja i promjene u frekvenciji pojavljivanja srednjih vrijednosti.

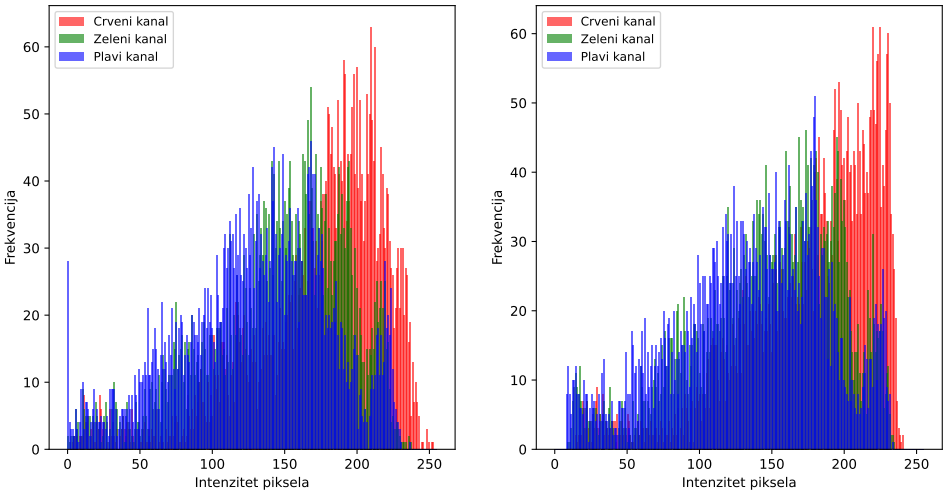
Najbolji rezultat PSNR, kao i SSIM metrike, postiže skalirana verzija kvadratne greške, RMSE, te će se rezultati postignuti korištenjem ove funkcije troška koristiti za poređenje sa referentnim radovima iz ove oblasti (Tabela 3). U poređenju sa drugim tehnikama, sistem ostvaruje bolju vrijednost PSNR samo u odnosu na [47]. Međutim, prijavljena vrijednost indeksa strukturne sličnosti premašuje sve tehnike, uključujući i one sa znatno većom ostvarenom vrijednošću PSNR.

Tabela 3: Poređenje dobijenih rezultata u pogledu kvaliteta umetanja sa relevantnim radovima

Tehnika	**PSNR [dB]**	**SSIM**
[37]	43.03	-
[68]	40.59	0.933
[46]	35.9	0.955
Predlog sistema	31.04	0.962
[47]	29.65	-



(a)



(b)

Slika 23: Oblici histograma prije i poslije umetanja vodenog žiga

6.2 Robusnost sistema

Robusnost označava sposobnost sistema da precizno identifikuje umetnute bitove vodenog žiga, kako u idealnim uslovima bez smetnji, tako i uz prisustvo šuma. Kao što je ranije naglašeno, kao napadi na sistem su uzeti Gausov šum, so i biber tip šuma i Gausov niskopropusni filtar. Predložena arhitektura postiže bolje [37], [46], [68], ili uporedive [47] rezultate u odnosu na odabrane radove. Primijećuje se da rad

Tabela 4: Poređenje dobijenih rezultata u pogledu robusnosti detekcije sa relevantnim radovima. Akronim BN označava bez napada, GŠ je Gausov šum, SB je so i biber tip šuma, a GF je Gausov filtar

Tehnika	**BER (%)**			
	BN	GŠ	SB	GF
[47]	0.000	0.000	0.000	0.000
Predlog sistema	0.000	0.002	0.001	0.001
[68]	0.000	0.001	0.002	0.004
[46]	0.000	0.110	0.056	0.000
[37]	0.000	0.114	0.066	0.000

sa najnižim vrijednostima PSNR-a daje najbolje rezultate u pogledu robusnosti, dok istraživanje sa najvećom ostvarenom vrijednošću maksimalni odnos signal-šum pravi najveću grešku u prisustvu šuma.

7 Zaključak

Sa rapidnim razvojem interneta i vještačke inteligencije, potreba za označavanjem vlasništva i zaštitom autorskih prava multimedijalnog sadržaja danas je važnija nego ikada. Umjetnici, stvaraoci i organizacije su konstantno na udaru zlonamjernih pojedinaca ili grupa koji imaju za cilj da prisvoje, mijenjaju ili nelegalno distribuiraju tuđi autorski rad. Veliki pomaci ostvareni u sferi vještačke inteligencije donijeli su nove metode zloupotrebe kojima je moguće, na najsofisticiraniji način, fabrikovati i falsifikovati digitalni sadržaj što može izazvati nesagledive posljedice po žrtvu - javnu ličnost, kompaniju ili instituciju. Zbog svega prethodno navedenog, potreba za razvijanjem i implementacijom različitih votermarking sistema, i primjenom vodenih žigova kao koncepta uopšte, danas je veća nego ikada.

Kreiranje sistema za umetanje vodenog žiga u digitalne slike predstavlja kompleksan zadatak koji zahtijeva temeljno planiranje, realizaciju i testiranje. U ovom radu predstavljen je sistem zasnovan na primjeni dubokog učenja, konkretno konvolucionih mreža, tehnike koja sa prolaskom godina i stalnim unaprijeđivanjem teži da istisne i odmjeni tradicionalne načine umetanja vodenog žiga.

Model se sastoji od dvije mreže međusobno suprotstavljenih ciljeva, umetača i detektora. Mreža umetača teži savršenoj rekonstrukciji slike, čime bi rad detektora bitova vodenog žiga bio onemogućen. Kompromis se postiže u zajedničkom treningu mreža uvođenjem težinskih faktora, pomoću kojih se sistem uči umetanju žiga na način koji ne ugrožava funkcionisanje niti jednog od dijelova sistema.

Uzimajući u obzir najčešće korištene metrike kada je ova oblast u pitanju, sistem postiže zadovoljavajuće rezultate po pitanju neprimjetnosti umetnutog žiga, kao i robusnosti prilikom detekcije bitova. Predložena arhitektura ima puno mjesta za poboljšanje, počevši sa povećanjem kapaciteta vodenog žiga. Trenutnih 8 bitova i 256 različitih poruka koje se mogu ugraditi predstavljaju dobru osnovu za dalju nadogradnju modela, međutim, u realnoj primjeni, ograničavajući su faktor za sistem. Ipak, kako je riječ o polu-krhkom pristupu, u kome je akcenat stavljen na ostvarivanju neprimjetnosti i robusnosti na benigne šumove, značajno povećanje kapaciteta bi dovelo do ugrožavanja polu-krhkog koncepta, te stoga treba postaviti prioritete u pogledu željenih karakteristika sistema.

Sa većom dostupnošću računarske snage, povećanje rezolucije ulaznih slika nad kojima se vrši trening i testiranje postaje tema razmatranja. Trenutnih 64×64 piksela je nedovoljno za širu primjenu sistema, uzimajući u obzir veličinu slika sa kojima se svakodnevno susrećemo. Šumovi simulirani u radu predstavljaju kap u moru smetnji na koje signal može naići prilikom prolaska kroz medijum. Veliki broj napada

koji nisu uzeti u obzir, poput filtriranja ili kompresija, mogu uzrokovati oštećenje ili uklanjanje bitova i ugroziti rad sistema. Sa druge strane, ukoliko se nastoji održati polu-krhki koncept, ne bi trebalo težiti potpunoj robusnosti na prethodno navedene smetnje, jer time pojam gubi na smislu. Na kraju, način odabira vrijednosti težinskih faktora kojima se reguliše zajednički trening umetača i detektora se mora istaći kao proces gdje postoji značajan prostor za napredak. Proces, kako je opisan, zahtijeva određeno vrijeme i nije podesan za precizno određivanje optimalnih vrijednosti parametara.

Literatura

[1] Directive - 2019/790 - EN - dsm - EUR-Lex — eur-lex.europa.eu. https://eur-lex.europa.eu/eli/dir/2019/790/oj.

[2] Zašto se internet pirateriji 'gleda kroz prste'. https://www.slobodnaevropa.org/a/internet-piraterija-torenti-toma-film-pirat/31519228.html.

[3] Deepfake video of Zelenskyy could be 'tip of the iceberg' in info war, experts warn. https://www.npr.org/2022/03/16/1087062648/deepfake-video-zelenskyy-experts-war-manipulation-ukraine-russia.

[4] Saraju P Mohanty. Digital watermarking: A tutorial review. *URL: http://www. csee. usf. edu/˜ smohanty/research/Reports/WMSurvey1999Mohanty. pdf*, 1999.

[5] Juergen Seitz. *Digital watermarking for digital media*. igi Global, 2005.

[6] Jiri Fridrich, Miroslav Goljan, and Arnold C Baldoza. New fragile authentication watermark for images. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 446–449. IEEE, 2000.

[7] C-T Li. Digital fragile watermarking scheme for authentication of jpeg images. *IEE proceedings-vision, image and signal processing*, 151(6):460–466, 2004.

[8] Abdulaziz Shehab, Mohamed Elhoseny, Khan Muhammad, Arun Kumar Sangaiah, Po Yang, Haojun Huang, and Guolin Hou. Secure and robust fragile watermarking scheme for medical images. *IEEE access*, 6:10269–10278, 2018.

[9] Eugene T Lin, Christine I Podilchuk, and Edward J Delp III. Detection of image alterations using semifragile watermarks. In *Security and Watermarking of Multimedia Contents II*, volume 3971, pages 152–163. SPIE, 2000.

[10] Ching-Yung Lin and Shih-Fu Chang. Semifragile watermarking for authenticating jpeg visual content. In *Security and watermarking of multimedia contents II*, volume 3971, pages 140–151. SPIE, 2000.

[11] Xiaojun Qi and Xing Xin. A quantization-based semi-fragile watermarking scheme for image content authentication. *Journal of Visual Communication and Image Representation*, 22(2):187–200, February 2011.

[12] C.I. Podilchuk and Wenjun Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications*, 16(4):525–539, May 1998.

[13] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.

[14] Igor Djurovic, Srdjan Stankovic, and Ioannis Pitas. Digital watermarking in the fractional fourier transformation domain. *Journal of Network and Computer Applications*, 24(2):167–173, 2001.

[15] Nikos Nikolaidis and Ioannis Pitas. Robust image watermarking in the spatial domain. *Signal processing*, 66(3):385–403, 1998.

[16] Srdjan Stankovic, Igor Djurovic, and Ioannis Pitas. Watermarking in the space/spatial-frequency domain using two-dimensional radon-wigner distribution. *IEEE transactions on image processing*, 10(4):650–658, 2001.

[17] Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking using multiresolution wavelet decomposition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 5, pages 2969–2972. IEEE, 1998.

[18] Chih-Chin Lai and Cheng-Chih Tsai. Digital image watermarking using discrete wavelet transform and singular value decomposition. *IEEE Transactions on instrumentation and measurement*, 59(11):3060–3063, 2010.

[19] Asifullah Khan, Syed Fahad Tahir, Abdul Majid, and Tae-Sun Choi. Machine learning based adaptive watermark decoding in view of anticipated attack. *Pattern Recognition*, 41(8):2594–2610, 2008.

[20] Asifullah Khan, Syed Fahad Tahir, and Tae-Sun Choi. Intelligent extraction of a digital watermark from a distorted image. *IEICE transactions on information and systems*, 91(7):2072–2075, 2008.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[22] Xin Zhong, Pei-Chi Huang, Spyridon Mastorakis, and Frank Y Shih. An automated and robust image watermarking scheme based on deep neural networks. *IEEE Transactions on Multimedia*, 23:1951–1961, 2020.

[23] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.

[24] Paarth Neekhara, Shehzeen Hussain, Xinqiao Zhang, Ke Huang, Julian McAuley, and Farinaz Koushanfar. Facesigns: semi-fragile neural watermarks for media authentication and countering deepfakes. *arXiv preprint arXiv:2204.01960*, 2022.

[25] Weiping Ding, Yurui Ming, Zehong Cao, and Chin-Teng Lin. A generalized deep neural network approach for digital watermarking analysis. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):613–627, 2021.

[26] Abdullah Bamatraf, Rosziati Ibrahim, and Mohd Najib B Mohd Salleh. Digital watermarking algorithm using lsb. In *2010 International Conference on Computer Applications and Industrial Electronics*, pages 155–159. IEEE, 2010.

[27] Shabir A Parah, Javaid A Sheikh, and GM Bhat. High capacity data embedding using joint intermediate significant bit (isb) and least significant bit (lsb) technique. *J Inf Eng Appl. ISSN (Paper)*, pages 2224–5782, 2012.

[28] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su. Reversible data hiding. *IEEE Transactions on circuits and systems for video technology*, 16(3):354–362, 2006.

[29] Walter Bender, Daniel Gruhl, Norishige Morimoto, and Anthony Lu. Techniques for data hiding. *IBM systems journal*, 35(3.4):313–336, 1996.

[30] Mahbuba Begum and Mohammad Shorif Uddin. Digital image watermarking techniques: a review. *Information*, 11(2):110, 2020.

[31] Igor Đurović. *Digitalna obrada slike*, Univerzitet Crne Gore, Elektrotehnički fakultet, Podgorica, 2006.

[32] Chi Kin Ho and Chang-Tsun Li. Semi-fragile watermarking scheme for authentication of jpeg images. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, volume 1, pages 7–11 Vol.1, 2004.

[33] W.C. Chu. Dct-based image watermarking using subsampling. *IEEE Transactions on Multimedia*, 5(1):34–38, 2003.

[34] Alessandro Piva, Mauro Barni, Franco Bartolini, and Vito Cappellini. Dct-based watermark recovering without resorting to the uncorrupted original image. In *Proceedings of international conference on image processing*, volume 1, pages 520–523. IEEE, 1997.

[35] Saraju P Mohanty, Kalpathi R Ramakrishnan, and Mohan S Kankanhalli. A dct domain visible watermarking technique for images. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, volume 2, pages 1029–1032. IEEE, 2000.

[36] Shinfeng D Lin and Chin-Feng Chen. A robust dct-based watermarking for copyright protection. *IEEE Transactions on Consumer Electronics*, 46(3):415–421, 2000.

[37] Soumitra Roy and Arup Kumar Pal. A blind dct based color watermarking algorithm for embedding multiple watermarks. *AEU-International Journal of Electronics and Communications*, 72:149–161, 2017.

[38] Vassilios Solachidis and Loannis Pitas. Circularly symmetric watermark embedding in 2-d dft domain. *IEEE transactions on image processing*, 10(11):1741–1753, 2001.

[39] Ante Poljicak, Lidija Mandic, and Darko Agic. Discrete fourier transform–based watermarking method with an optimal implementation radius. *Journal of Electronic Imaging*, 20(3):033008–033008, 2011.

[40] Matthieu Urvoy, Dalila Goudia, and Florent Autrusseau. Perceptual dft watermarking with improved detection and robustness to geometrical distortions. *IEEE Transactions on Information Forensics and Security*, 9(7):1108–1119, 2014.

[41] Discrete wavelet transform. https://www.st-andrews.ac.uk/~wjh/dataview/tutorials/dwt.html.

[42] The Uncertainty Principle (Stanford Encyclopedia of Philosophy). https://plato.stanford.edu/entries/qt-uncertainty/.

[43] Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. *Proceedings of the IEEE*, 87(7):1167–1180, 1999.

[44] Gwo-Jong Yu, Chun-Shien Lu, and Hong-Yuan Mark Liao. Mean-quantization-based fragile watermarking for image authentication. *Optical Engineering*, 40(7):1396–1408, 2001.

[45] Nikita Kashyap and GR Sinha. Image watermarking using 3-level discrete wavelet transform (dwt). *International Journal of Modern Education and Computer Science*, 4(3):50, 2012.

[46] Xiao-bing Kang, Fan Zhao, Guang-feng Lin, and Ya-jun Chen. A novel hybrid of dct and svd in dwt domain for robust and invisible blind image watermarking with optimal embedding strength. *Multimedia Tools and Applications*, 77:13197–13224, 2018.

[47] Amit Kumar Singh. Improved hybrid algorithm for robust and imperceptible multiple watermarking using digital images. *Multimedia Tools and Applications*, 76:8881–8900, 2017.

[48] Md Saiful Islam and Ui Pil Chong. A digital image watermarking algorithm based on dwt dct and svd. *International Journal of Computer and Communication Engineering*, 3(5):356, 2014.

[49] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[50] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[51] Ljubisa Stankovic. *Digital Signal Processing with Selected Topics*. 11 2015.

[52] Marko M Dabović and Igor I Tartalja. Duboke konvolucijske neuronske mreže–koncepti i aktuelna istraživanja. *Proceedings of the Zbornik 61. Konferencije za Elektroniku, Telekomunikacije, Računarstvo, Automatiku i Nuklearnu Tehniku, ETRAN 2017*, pages 1–1, 2017.

[53] What is cross-entropy loss function? https://www.geeksforgeeks.org/what-is-cross-entropy-loss-function/.

[54] 3Blue1Brown - Backpropagation calculus. https://www.3blue1brown.com/lessons/backpropagation-calculus.

[55] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[56] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/k2). In *Dokl. Akad. Nauk. SSSR*, volume 269, page 543, 1983.

[57] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[58] Tijmen Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical report*, 2017.

[59] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[60] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[62] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[63] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[64] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[65] Kosta Pavlović, Slavko Kovačević, Igor Djurović, and Adam Wojciechowski. Robust speech watermarking by a jointly trained embedder and detector using a dnn. *Digital Signal Processing*, 122:103381, April 2022.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[67] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[68] Khalid M Hosny, Mohamed M Darwish, Kenli Li, and Ahmad Salah. Parallel multi-core cpu and gpu for fast and robust medical image watermarking. *IEEE Access*, 6:77212–77225, 2018.