

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Ivan Ilić, Karlo Koledić, Marin Drabić, Ante Šućur,
Tea Šarić, Viktorija Sajković, Lea Patekar, Marija Klaić

Procjena zubnog statusa pomoću dubokog učenja

Zagreb, 2018.

Ovaj rad izrađen je u Zavodu za elektroničke sustave i obradbu informacija na Fakultetu elektrotehnike i računarstva u suradnji sa studentima Stomatološkog fakulteta u Zavodu za dentalnu antropologiju. Rad je izrađen pod vodstvom izv. prof. dr. sc. Marka Subašića s Fakulteta elektrotehnike i računarstva i izv. prof. dr. sc. Marina Vodanovića sa Stomatološkog fakulteta te je predan na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2017./2018.

Računalno zahtjevni eksperimenti tijekom izrade ovog rada provedeni su pomoću TitanXp GPU jedinica koje je donirala tvrtka Nvidia.

SADRŽAJ

1. Uvod.....	1
2. Svrha, ciljevi i hipoteza istraživanja.....	3
2.1. Svrha istraživanja.....	3
2.2. Ciljevi istraživanja.....	3
2.3. Hipoteza istraživanja.....	3
3. Materijali i metode.....	4
3.1. Odabir ortopantomograma.....	4
3.1.1. Obrada ortopantomograma.....	4
3.1.2. Postupak obrade ortopantomograma.....	5
3.1.3. Koraci obrade ortopantomograma.....	6
3.2. Duboke neuronske mreže.....	9
3.2.1. Arhitektura neuronske mreže.....	9
3.2.2. Prednosti i mane.....	10
3.2.3. Duboke neuronske mreže.....	11
3.2.4. Konvolucijske mreže.....	11
3.2.5. Umjetna neuronska mreža.....	13
3.2.5.1. Regresija i klasifikacija.....	13
3.2.5.2. Regresija.....	13
3.2.5.3. Klasifikacija.....	17
3.2.5.4. Umjetne neuronske mreže za regresiju.....	20
3.2.5.5. Umjetne neuronske mreže za klasifikaciju.....	21
3.2.6. Duboke neuronske mreže za detekciju i klasifikaciju.....	22
3.2.7. Faster R-CNN.....	24
3.2.8. YOLO.....	30
4. Rezultati.....	35
4.1. Procjena dobi.....	35
4.2. Procjena spola.....	36
4.3. Detekcija zuba.....	38
4.3.1. Rezultati detekcije pomoću mreže Faster RCNN.....	40
4.3.2. Rezultati detekcije pomoću mreže YOLO.....	41
5. Rasprava.....	43
5.1. Primjeri praktične primjene.....	44
5.1.1. „Ljudi bez identiteta“.....	44
5.1.2. Utvrđivanje identiteta žrtava nesreća.....	44
5.1.3. Procjena dobi napuštene djece.....	45
6. Zaključak.....	46
LITERATURA.....	47
Sažetak.....	50
Summary.....	51

1. Uvod

Zubni status označava stanje svakog pojedinog zuba gornje i donje čeljusti u usnoj šupljini pojedinca. Prilikom registracije zubnog statusa bilježi se broj prisutnih zuba, broj izvađenih zuba, broj zuba s karijesom, s ispunima te broj zuba opskrbljenih protetskim nadomjescima (krunice, mostovi, zubne proteze). Osim toga, zubni status uključuje registraciju svih ostalih obilježja karakterističnih za stanje pojedinog zuba, odnosno stomatognatog sustava u cjelini. U svakodnevnoj praksi, stomatolog raznim dijagnostičkim metodama određuje zubni status. Osim intraoralnim pregledom, koristi se i radiološkim pretragama, koje uključuju rendgenska snimanja (RTG), kompjutoriziranu tomografiju (CT), Cone Beam kompjuteriziranu tomografiju (CBCT) i slično.

Procjena dentalne dobi je postupak koji se rabi u dječjoj stomatologiji i ortodontici u kliničke svrhe te u forenzičnoj stomatologiji pri identifikaciji pojedinca i izradi dentalnog profila [1]. Dentalnim se profiliranjem obično može osigurati informacija o životnoj dobi, spolu, socijalnom statusu, podrijetlu, a ponekad i podaci o zanimanju, prehranbenim navikama, sistemskim bolestima i bolestima zuba. Poznavanjem antropoloških obilježja zuba forenzični stomatolog/dentalni antropolog može procijeniti etničku pripadnost na temelju izgleda i veličine kostiju kraniofacijalnog skeleta te izgleda i veličine zuba [2]. Svrha procjene dobi na temelju ljudskih ostataka je utvrđivanje biološkog profila umrle osobe. Dentalna identifikacija jedinstven je način identifikacije pri avionskim nesrećama kada zbog visoke temperature stradavaju sva tkiva, a opstaju zubi. Njezino značenje dolazi do izražaja i prilikom drugih velikih prometnih katastrofa gdje stradava velika skupina ljudi [3]. Uz to, potreba za dentalnom identifikacijom visoka je i u ratnim i poslijeratnim situacijama.

Određivanje dobi neidentificiranih ljudskih ostataka pomoću zuba temelji se na poznavanju bioloških promjena koje obilježavaju rast i razvoj zuba. Kronologija tih promjena ovisi o mnogim čimbenicima: općem tjelesnom razvoju, genetskom utjecaju, konstituciji, endokrinom statusu, prehrani, rasnoj pripadnosti. Stoga se može odrediti samo približna dob. Nakon završenog rasta i razvoja svih zuba određivanje dobi je teže, a temelji se na jedva vidljivim, ali trajnim promjenama u strukturi tvrdih zubnih tkiva. Najčešće korištene metode za procjenu dentalne dobi utemeljene su na analizi rendgenskih snimaka na kojima je moguće vidjeti različite faze u razvoju zuba, ali i histološke promjene na zubima već završenog razvoja. Ubrzo nakon otkrića x-zraka stomatološke rendgenske snimke postale su nezamjenjiv i trajan podatak koji doktorima dentalne medicine daju uvid u stanje zubala te koštanih dijelova obiju čeljusti [4].

Određivanje spola također je moguće na temelju ortopantomograma. Skelet odrasle žene i muškarca razlikuje se po obliku i veličini. Morfološke razlike u građi kostura između spolova počinju se razvijati već prije rođenja, a spolni dimorfizam povećava se tijekom djetinjstva i adolescencije. Nakon puberteta, otprilike oko osamnaeste godine života, spolne razlike u kosturu toliko su izražene da je moguće odrediti spol s točnošću većom od 90%. Ženski kostur je manjih dimenzija i nježnije građe, dok je muški veći i robusniji. Naravno, postoje individualne varijabilnosti, ali i varijabilnosti unutar samih populacija, pa se i to treba uzeti u

obzir prilikom istrage [5]. Prema tome, kosti lubanje kod žena su manje i nježnije, a cijelo lice je više okruglo, dok su kod muškaraca kosti veće i lice je četvrtastog oblika. Glavna prednost određivanja zubnog statusa, dobi i spola pomoću ortopantomograma je neinvazivnost, što znači da je primjenjiv kod živih, ali i preminulih osoba. Pouzdanost, preciznost i točnost rendgenskih metoda za procjenu dentalne dobi ovisi, između ostaloga, o iskustvu stručnjaka koji vrši procjenu te o stupnju ponovljivosti očitavanja rendgenskih snimaka. Od presudne je važnosti da se pri ponovljenim očitavanjima istih snimaka dobiju isti rezultati. Nažalost, tu se zbog ljudskog čimbenika nerijetko događaju pogreške [6]. Korištenjem metoda dubokog učenja, konkretnije umjetnih neuronskih mreža, problem određivanja zubnog statusa te procjene spola i godina formaliziramo na matematičkoj razini. Cijeli proces postaje kvantitativno mjerljiv, osigurava se konzistentnost i preciznost rezultata. Također, otklanja se potreba za visokom naobrazbom, iskustvom te subjektivnom procjenom pojedinca. Modele umjetnih neuronskih mreža moguće je koristiti bez posebne opreme i znanja, lako se unapređuju, a sâm proces dobivanja rezultata uz priloženi ortopantomogram traje svega nekoliko sekundi.

2. Svrha, ciljevi i hipoteza istraživanja

2.1. Svrha istraživanja

Svrha ovog istraživanja je ispitati mogućnosti primjene metoda dubokog učenja za potrebe računalnog uzimanja zubnog statusa, procjene spola i dobi pojedinca na temelju ortopantomograma (rendgenska snimka kompletne gornje i donje čeljusti s okolnim anatomskim strukturama koja daje opću informaciju o kostima glave, sinusima, čeljustima, temporomandibularnim zglobovima i zubima). Kako bi to bilo moguće, osmislit će se i razviti u praksi primjenjiva metoda temeljena na dubokom učenju koja će moći samostalno određivati dentalnu dob i spol pojedinca.

2.2. Ciljevi istraživanja

Cilj istraživanja je razviti metode temeljene na dubokom učenju koje će omogućiti računalnu identifikaciju zuba iz ortopantomograma kao sastavnog dijela uzimanja zubnog statusa te metode za procjenu spola i dobi pojedinca. U ovom radu objasniti će se koncept umjetnih neuronskih mreža te će se prezentirati ukupno četiri korištena modela. Ako je suditi po dosadašnjim rezultatima ostvarenim korištenjem metoda dubokog učenja u raznim područjima znanosti, mogu se očekivati vrlo dobri rezultati i u ovom području.

Ispitat će se mogućnosti i načini primjene metoda dubokog učenja za procjenu zubnog statusa, dobi i spola, stoga su specifični ciljevi ovog istraživanja:

- Identificirati prednosti i nedostatke primjene dubokog učenja za procjenu dentalne dobi i spola;
- Utvrditi teorijske i tehnološke pretpostavke potrebne kako bi se duboko učenje primijenilo za procjenu dentalne dobi i spola;
- Razviti i izraditi računalni model temeljen na dubokom učenju koji će moći samostalno odrediti dentalnu dob i spol pojedinca;
- Dokazati da je primjenom trenutno dostupnih tehnologija na području dubokog učenja moguće procijeniti dentalnu dob i spol uz prihvatljivu razinu preciznosti, pouzdanosti i ponovljivosti.

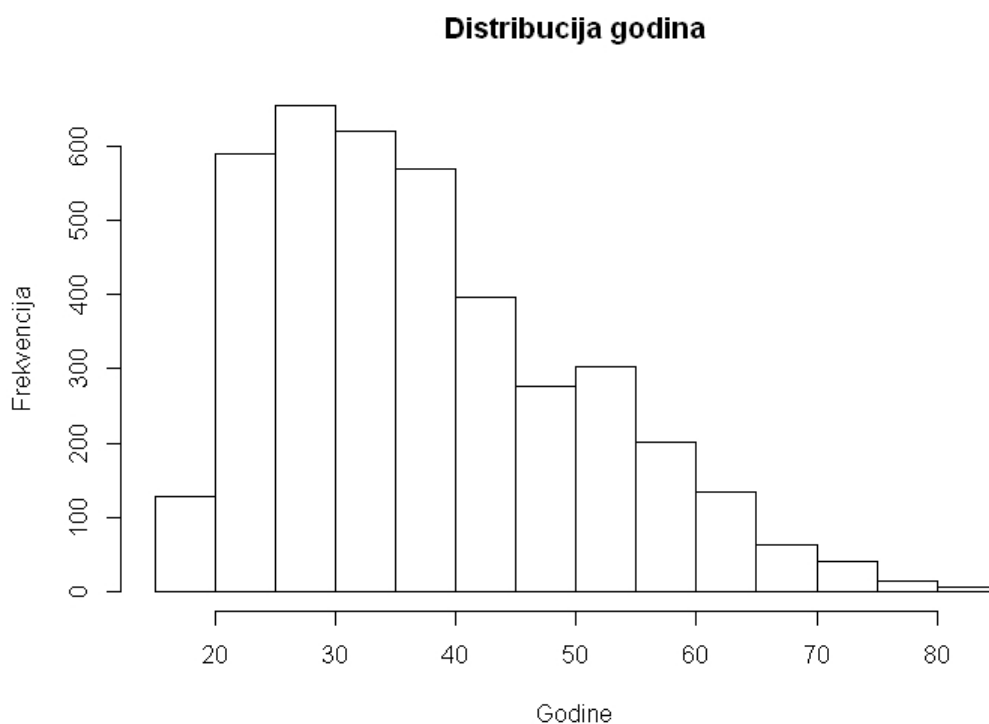
2.3. Hipoteza istraživanja

Primjenom trenutno dostupnih tehnologija na području dubokog učenja moguće je procijeniti dentalnu dob, status te spol pojedinca uz prihvatljivu razinu preciznosti, pouzdanosti i ponovljivosti.

3. Materijali i metode

3.1. Odabir ortopantomograma

Za istraživanje je korišteno 4000 digitalnih ortopantomograma poznate dobi i spola iz zbirke Zavoda za dentalnu antropologiju Stomatološkog fakulteta Sveučilišta u Zagrebu. Ortopantomogrami su prikupljeni na Zavodu tijekom više godina, primjereno su pohranjeni i koriste se u znanstveno-istraživačke svrhe. S ortopantomograma su uklonjeni podaci o identitetu. Za potrebe ovog istraživanja nije bilo snimanja novih ortopantomograma.



Slika 2.1 – Distribucija ortopantomograma po godinama

3.1.1. Obrada ortopantomograma

Obradu su proveli članovi tima, studenti Stomatološkog fakulteta u Zagrebu, koji su stručni u tome, kako bi baza označenih slika bila što pouzdanija. Korišteni ortopantomogrami obrađivani su pomoću besplatnog računalnog programa LabelImg [7]. To je grafički alat za označavanje slika napisan u programskom jeziku Python. Oznake se spremaju kao XML datoteke u PASCAL VOC formatu za daljnje korištenje u neuronskim mrežama.

3.1.2. Postupak obrade ortopantomograma

Obrada je započeta unosom digitalnog oblika ortopantomograma u navedeni računalni program. Stvaranjem pravokutnika oko svakog pojedinog zuba, cilj je bio obuhvatiti cijeli opseg pojedinog zuba, uključujući krunu i korijen. Na mjestima izvađenog i nenadomještenog zuba otprilike je postavljen pravokutnik u okvirima u kojima bi taj zub postojao. Zube smo označavali dualnim ili dvobrojnim sustavom koji je predložila FDI (Federation Dentaire Internationale) u Bukureštu 1970. godine, a koji su prihvatile Svjetska zdravstvena organizacija (WHO) i International Association of Dental Research (IADR). Svaki zub označava se dvjema brojkama: prva označava kvadrant, a druga broj. Kvadranti u trajnoj denticiji: 1, 2, 3, 4. Kvadranti u mliječnoj denticiji: 5, 6, 7, 8. Što se tiče izgovora, svaka znamenka oznake izgovara se pojedinačno. Npr. za zub s oznakom 11 kažemo „zub jedan jedan“, a ne „jedanaest“; oznaka 11 odnosi se na trajni gornji desni prvi sjekutić.

TRAJNI ZUBI																
gore desno								gore lijevo								
18	17	16	15	14	13	12	11		21	22	23	24	25	26	27	28
48	47	46	45	44	43	42	41		31	32	33	34	35	36	37	38
dolje desno								dolje lijevo								

Slika 2.2 – Prikaz dvobrojnog (dualnog) označavanja trajnih zubi

Kako bismo što preciznije odredili status zuba, osim dualnog sustava označavanja dogovoreno je šest oznaka velikim tiskanim slovima, pri čemu svaka od njih ima svoje značenje za određivanje zubnog statusa. Te oznake navedene su u tablici. Ovisno o njegovu stanju, postoji mogućnost da pojedini zub ima više oznaka. Uz svaku oznaku stoji objašnjenje i primjer. Uz oznaku jednog zuba može biti pridruženo više dodatnih oznaka. Između oznaka ne stavlja se razmak. Npr. 26CE – označava trajni gornji lijevi prvi kutnjak s endodontskim punjenjem i krunicom.

Tablica 3.1 – Objašnjenje oznaka zuba gornje čeljusti

Oznaka	Značenje	Oznaka	Značenje
11	Trajni gornji desni središnji sjekutić	21	Trajni gornji lijevi središnji sjekutić
12	Trajni gornji desni bočni sjekutić	22	Trajni gornji lijevi bočni sjekutić
13	Trajni gornji desni očnjak	23	Trajni gornji lijevi očnjak
14	Gornji desni prvi pretkutnjak	24	Gornji lijevi prvi pretkutnjak
15	Gornji desni drugi pretkutnjak	25	Gornji lijevi drugi pretkutnjak
16	Trajni gornji desni prvi kutnjak	26	Trajni gornji lijevi prvi kutnjak
17	Trajni gornji desni drugi kutnjak	27	Trajni gornji lijevi drugi kutnjak
18	Gornji desni treći kutnjak	28	Gornji lijevi treći kutnjak

Tablica 3.2 – Objašnjenje oznaka zuba donje čeljusti

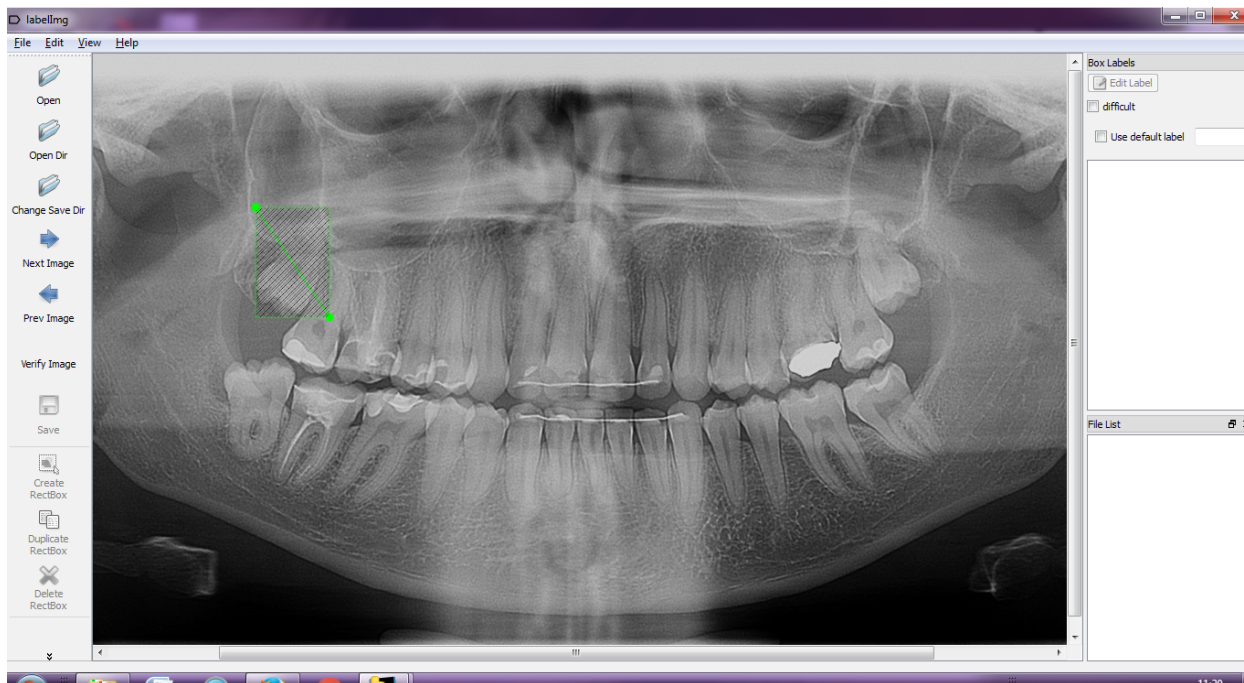
Oznaka	Značenje	Oznaka	Značenje
31	Trajni donji lijevi središnji sjekutić	41	Trajni donji desni središnji sjekutić
32	Trajni donji lijevi bočni sjekutić	42	Trajni donji desni bočni sjekutić
33	Trajni donji lijevi očnjak	43	Trajni donji desni očnjak
34	Donji lijevi prvi pretkutnjak	44	Donji desni prvi pretkutnjak
35	Donji lijevi drugi pretkutnjak	45	Donji desni drugi pretkutnjak
36	Trajni donji lijevi prvi kutnjak	46	Trajni donji desni prvi kutnjak
37	Trajni donji lijevi drugi kutnjak	47	Trajni donji desni drugi kutnjak
38	Donji lijevi treći kutnjak	48	Donji desni treći kutnjak

3.1.3. Koraci obrade ortopantomograma

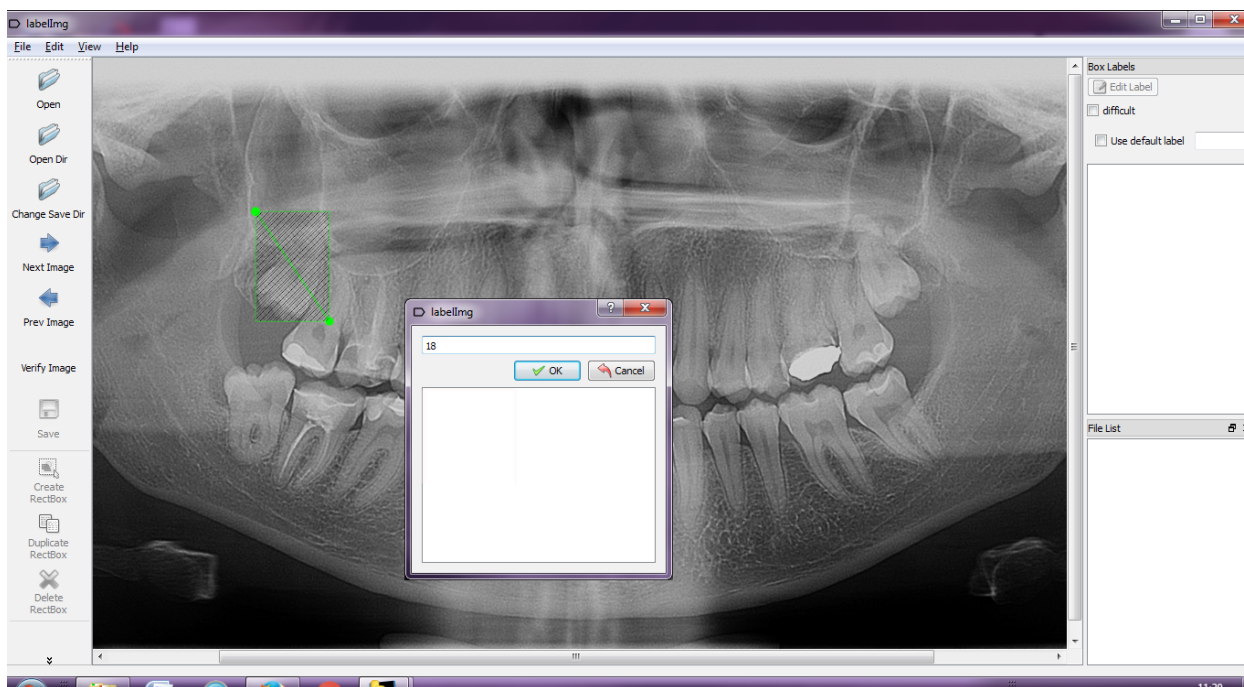
Sljedećim slikama prikazani su koraci označavanja zubi.



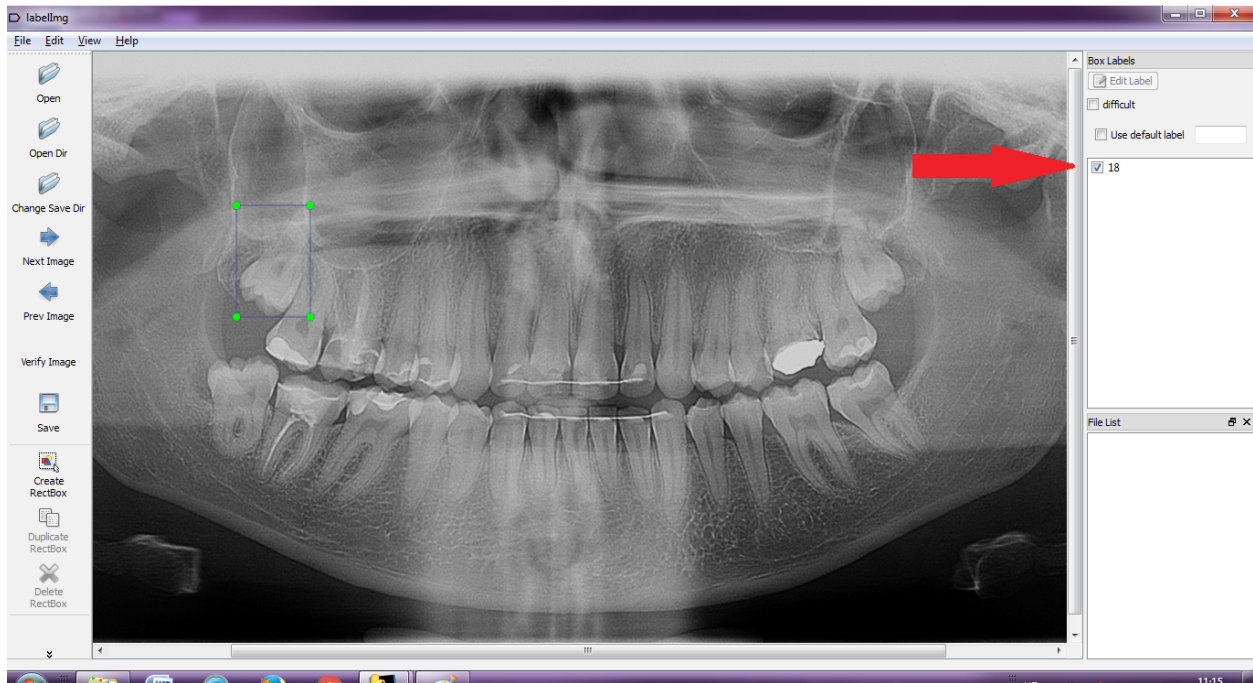
Slika 2.3 – Unošenje ortopantomograma u računalni program LabelImg



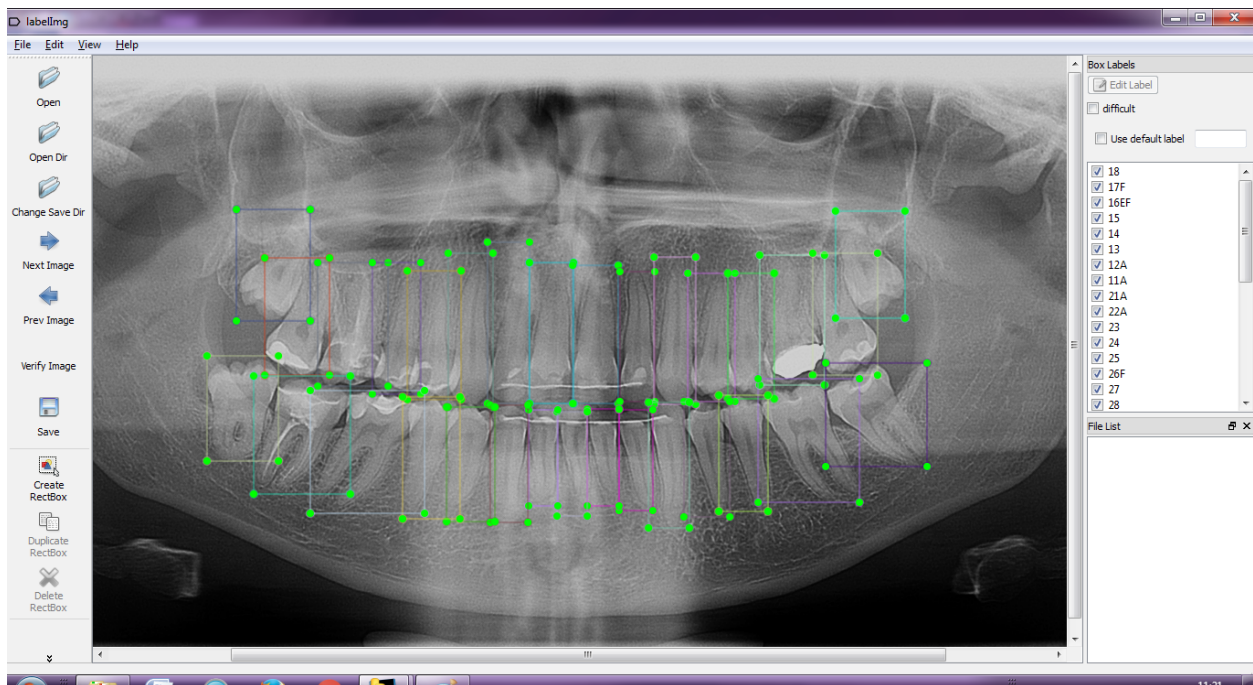
Slika 2.4 – Postavljanje pravokutnika oko granica zuba



Slika 2.5 – Označavanje zuba dvobrojnim sustavom



Slika 2.6 – Prikaz označenog zuba unutar pravokutnika, 18



Slika 2.7 – Prikaz svih označenih zuba na ortopantomogramu, uz pripadajuće oznake zubnog statusa

3.2. Duboke neuronske mreže

Strojno učenje predstavlja područje računalne znanosti koje se bavi proučavanjem algoritama koji omogućavaju rješavanje problema kroz učenje na temelju podataka [8]. Jedan od najpopularnijih pristupa strojnog učenja su umjetne neuronske mreže.

Umjetna neuronska mreža može se opisati kao model zaključivanja koji funkcionira slično kao i ljudski mozak. Mozak se sastoji od približno 10 milijardi neurona i 60 bilijuna veza među njima, a istovremenim korištenjem više neurona mozak obavlja svoje funkcije brže od najbržih svjetskih računala. Najbitnije svojstvo neuronskih mreža je svojstvo plastičnosti. Na svaki simulacijski uzorak događaju se dugotrajne promjene u snazi veza među neuronima, stvaraju se veze s drugim neuronima, a ponekad se i cijele kolekcije neurona sele s jednog mjesta na drugo. Ljudski mozak može se definirati kao vrlo složen, nelinearan i paralelan sustav za obradu informacija. Podaci koji dolaze do mozga pohranjuju se i obrađuju u neuronskim mrežama istovremeno kroz cijelu mrežu, a ne na pojedinim lokacijama, tj. podaci i njihova obrada su globalni, a ne lokalni. Zbog svojstva plastičnosti, veze koje dovode do „točnog odgovora“ jačaju, a ostale slabe. Rezultat svojstva plastičnosti je sposobnost učenja kroz iskustvo.

Učenje je sposobnost koja razlikuje živa bića od stvari, a omogućiti računalu da uči je velik korak naprijed u području umjetne inteligencije.

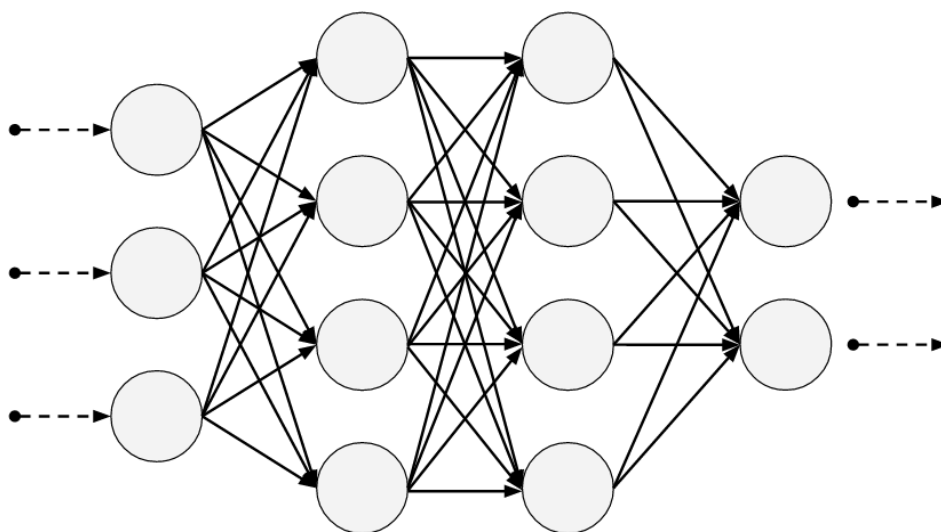
Prvi model za umjetne neuronske mreže stvorili su Warren McCulloch i Walter Pitts 1943. [9] godine, a istraživanje se temeljilo na dvama pristupima. Prvi pristup temeljio se na istraživanju bioloških procesa u mozgu, dok se drugi temeljio na primjenjivanju neuronskih mreža u području umjetne inteligencije. Nemogućnost rješavanja „isključivo-ili“, kao i računala koja nisu imala dovoljno jake procesore za upravljanje velikim neuronskim mrežama doveli su do stagniranja istraživanja neuronskih mreža. Ključni trenutak u istraživanju neuronskih mreža bio je algoritam *backpropagation* koji je osmislio Paul Werbos 1975. [9] godine. *Backpropagation* je riješio „isključivo-ili“ problem i općenito ubrzao treniranje višeslojnih mreža. Funkcioniranje neuronskih mreža nakon stvaranja *backpropagation* algoritma opet postaje tema mnogih istraživanja.

Razlika među mrežama je u arhitekturi, načinu učenja i signalima. Arhitektura se dijeli na mreže s vezama isključivo prema naprijed i na mreže s povratnim vezama. Kod mreža s povratnim vezama, veze su zatvorene na način da se rezultat analize vraća na početak i uvježbavanje ponavlja s ciljem minimiziranja pogreške. Prema načinu učenja, mreže se dijele na nadgledane, pojačane (podaci se pribavljaju interaktivno iz okoline) te nenadgledane. Nadgledane neuronske mreže rade s označenim tipom podataka (engl. *labeled*) i moguće je procijeniti točnost, dok nenadgledane rade s neoznačenim tipom podataka.

3.2.1. Arhitektura neuronske mreže

Umjetnu neuronsku mrežu čini hijerarhija slojeva, a neuroni u mreži raspoređeni su uzduž tih slojeva. Ulazne i izlazne slojeve čine neuroni povezani s vanjskom okolinom. Cilj je

uskladiti ponašanje mreže s vanjskom okolinom. Najjednostavnija umjetna neuronska mreža sastoji se od ulaznog, izlaznog i barem jednog skrivenog sloja. Signali se prenose unaprijed na „sloj po sloj“ osnovi.



Slika 3.1 – Primjer arhitekture neuronske mreže [10]

Svaki sloj u neuronskoj mreži ima specifičnu funkciju. Ulazni sloj redistribuira zaprimljene signale iz vanjskog svijeta na sve neurone u skrivenom sloju te nikad ne obrađuje podatke, dok izlazni sloj prihvaća uzorke iz skrivenih slojeva i uspostavlja izlazni uzorak cijele mreže. Neuroni koji pripadaju skrivenom sloju otkrivaju značajke koje pripadaju ulaznim uzorcima, a te se značajke koriste pri određivanju izlazne sheme u izlaznom sloju.

Najpopularnija metoda učenja umjetnih neuronskih mreža je upravo *backpropagation* algoritam. Za treniranje korištenjem *backpropagation* algoritma računa se pogreška između dobivenih i očekivanih vrijednosti. Koristeći izračunatu pogrešku, težine na čvoru se povećavaju množenjem pogreške globalnim skalarom. Složeni dio učenja s praćenjem unaprijed jest procijeniti koji ulaz je razlog najlošije procjene i kako promijeniti taj element kako bi se ispravila pogreška. Tijekom procesa treniranja primjenjuje se prosljeđivanje unaprijed, a izlaz svakog neurona računa se postepeno za svaki sloj. Razlika između očekivanog i dobivenog izlaza u završnom sloju vraća se u protekle čvorove, uglavnom modificirana derivacijom prijenosne funkcije, te se prilagođavaju težine. Ovaj proces izvršava se za svaki sloj.

3.2.2. Prednosti i mane

Neuronske mreže su pristup rješavanju problema koji će uvijek rezultirati rješenjem, ali kvaliteta rješenja ovisi o mnogo čimbenika. Na temu umjetnih neuronskih mreža izvršeno je iznimno mnogo akademskih istraživanja kojima je cilj unaprijediti kvalitetu njihova korištenja. Njihovoj popularnosti potpomogla je jednostavnost koncepta i korištenja, a dosad su stvorene brojne implementacije i okviri za njihovo korištenje. Iz tih razloga već se jako dugo uspješno

koriste u industriji. Međutim, neuronske mreže su u vrlo malo slučajeva optimalno rješenje za određeni problem. Gotovo uvijek postoje alternative koje su brže, jednostavnije za treniranje i daju bolja rješenja. Što je veći broj neurona, teže ih je trenirati i potrebno je sve više memorije.

Način na koji neuronska mreža rješava problem je za inženjera „crna kutija“, tj. nitko ne može procijeniti kako dolazi do rješenja, pa je u slučaju krivog rada vrlo teško pronaći pogrešku. Stoga ne treba za svaki problem odmah primijeniti ovakav pristup rješenju jer u mnogo slučajeva postoji bolje i optimalnije rješenje. Međutim, za probleme koji su iznimno složeni i za koje ne postoje kvalitetna alternativna rješenja neuronske mreže su najbolji pristup.

3.2.3. Duboke neuronske mreže

Kroz povijest razvijanja neuronskih mreža najveći je problem bio manjak memorije i nemogućnost obrade velikih količina podataka. Stoga su mreže uvijek bile ograničene na najviše četiri skrivena sloja (vrlo sporo). U posljednjih desetak godina pojavom jačih procesora, računala i sve većih količina podataka kreiraju se mreže s mnogo većim brojem skrivenih slojeva. Takve se mreže nazivaju „duboke neuronske mreže“. Dodatnu prepreku za duboke neuronske mreže predstavljala su teoretska ograničenja za učenje dubokih slojeva, no istraživači su u posljednjih desetak godina predložili neka uspješna rješenja koja su otvorila brojne mogućnosti primjene.

3.2.4. Konvolucijske mreže

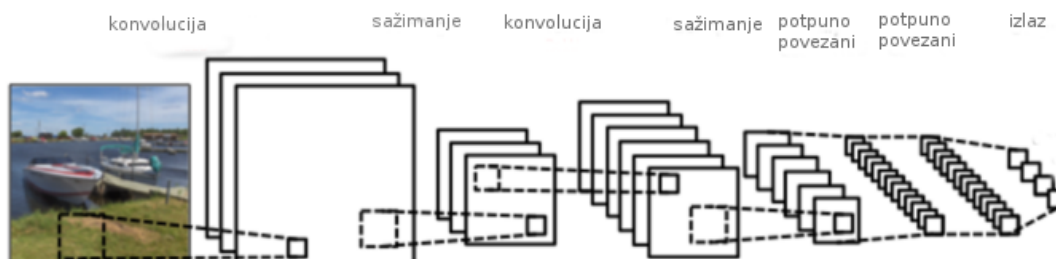
Trenutno vrlo popularan tip dubokih neuronskih mreža su konvolucijske neuronske mreže (engl. *convolutional neural networks*). One se vrlo često koriste u problemima vezanima uz analizu slike, a uspješne su i u brojnim drugim područjima. Glavna prednost konvolucijskih mreža jest njihova sposobnost da samostalno nauče detektirati važne uzorke u slici, za razliku od umjetnih neuronskih mreža gdje programer mora odlučiti koji su važni uzorci. To je iznimno korisno u prepoznavanju objekata jer je teško odrediti koji uzorci su zapravo važni za izvršavanje zadatka. Memorijski su vrlo učinkovite zato što je smanjen broj težina i parametara, što omogućuje univerzalnu iskoristivost. Zbog pomicanja filtera po slici moguće je detektirati objekt bez obzira na njegovu lokaciju.

Dosad su dizajnirane mnoge arhitekture konvolucijske neuronske mreže [11]. Najstarija arhitektura konvolucijske mreže je LeNet-5 (1998.), a koristile su je banke za prepoznavanje potpisa. Ulaz su bile male slike veličine 32x32 piksela zato što nisu zahtijevale veliku količinu memorije.

AlexNet (2012.) je mreža sa sličnom arhitekturom kao i LeNet-5, samo s većim brojem slojeva, filtera i nanizanim konvolucijskim slojevima. GoogleNet/Inception (2014.) je mreža čija je preciznost bila gotovo jednaka ljudskoj preciznosti. Uvelike je smanjila broj parametara u odnosu na AlexNet arhitekturu. VGGNet (2014.) je najpopularnija arhitektura za detekciju uzoraka u slici.

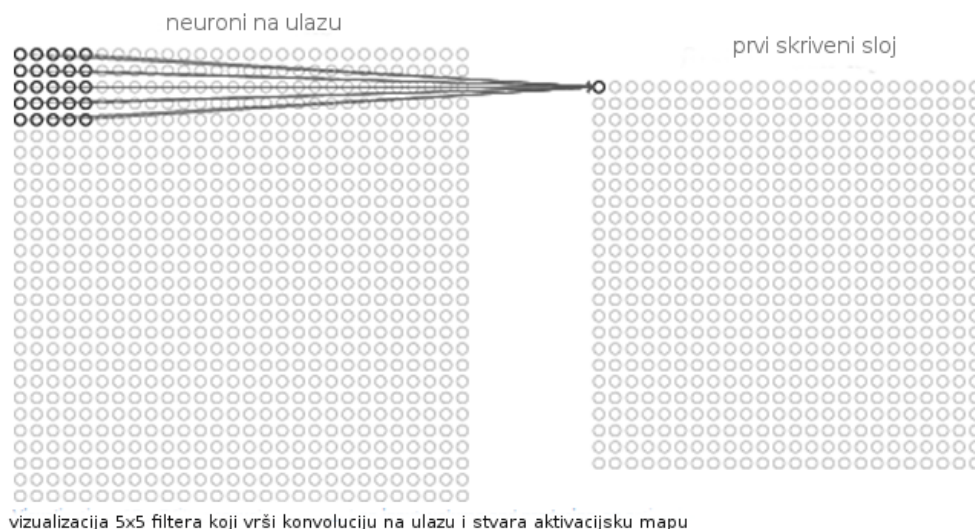
Struktura konvolucijskih mreža sastoji se od ulaznog, skrivenog i izlaznog sloja. Za razliku od običnih neuronskih mreža, skriveni sloj mora sadržavati odgovarajuće slojeve, a to su:

- Konvolucijski slojevi;
- Slojevi sažimanja;
- Potpuno povezani slojevi;
- Normalizacijski slojevi.



Slika 3.2 – Primjer arhitekture konvolucijske mreže

Konvolucijski slojevi primjenjuju operaciju konvolucije nad ulazom te prenose rezultat operacije do sljedećeg sloja. Konvolucija zapravo predstavlja reakciju neurona na ulaz, a svaki neuron obrađuje podatke samo za njegovo receptivno polje. Konvolucijski sloj sastoji se od filtera čije je težine potrebno naučiti kako bi se poboljšala uspješnost mreže.



Slika 3.3 – Obavljanje konvolucije na ulazu

Rezultat konvolucije je dvodimenzionalna aktivacijska mapa i ona predstavlja odziv filtera na svakoj lokaciji. Cilj je postići da se filter aktivira kada prepozna slikovna svojstva s kojima se već susreo. Npr. to mogu biti rubovi, kutovi i slično, a što je mreža dublja, svojstva postaju sve složenija i mreža može početi prepoznavati objekte, životinje i slično.

3.2.5. Umjetna neuronska mreža

3.2.5.1. Regresija i klasifikacija

Općenito govoreći, neuronske mreže koriste se u svrhu grupiranja kroz nenadzirano učenje, klasifikacije putem nadziranog učenja ili regresije. Drugim riječima, one pomažu pri grupiranju neoznačenih podataka, kategoriziranju označenih podataka ili predviđanju numeričkih vrijednosti.

U ovome radu koristili smo postupak klasifikacije i postupak regresije. Stoga slijede njihovi kratki opisi.

Zadatak klasifikacije je aproksimirati funkciju mapiranja f od ulaznih varijabli X prema diskretnim izlaznim varijablama y koje opisuju pripadnost zadanim klasama, tj. svaki ulazni primjer svrstava se u određenu klasu. Najčešće korišteni postupak klasifikacije jest logistička regresija.

Za razliku od klasifikacije, zadatak regresije je aproksimirati funkciju mapiranja f od ulaznih varijabli X prema kontinuiranoj izlaznoj varijabli y . Za svaki ulazni primjer izračunava se izlaz koji je stvarna vrijednost, poput cijelog broja ili broja s pomičnim zarezom. Najčešće korišteni postupci regresije su linearna regresija i polinomijalna regresija.

3.2.5.2. Regresija

Za linearno odvojive podatke koristi se postupak linearne regresije čiji je model 2D pravac, a za nelinearno odvojive podatke postupak polinomijalne regresije čiji je model prikazan polinomom stupnja 2 ili više.

Modeli regresije također se dijele ovisno o broju ulaznih te o broju izlaznih varijabli, odnosno značajki. Univarijatna ili jednostavna regresija sastoji se od jedne ulazne varijable, a multivarijatna ili višestruka od više ulaznih varijabli. Zatim, jednoizlazna regresija na izlazu daje jednu varijablu, a višeizlazna regresija kao izlaz daje više varijabli.

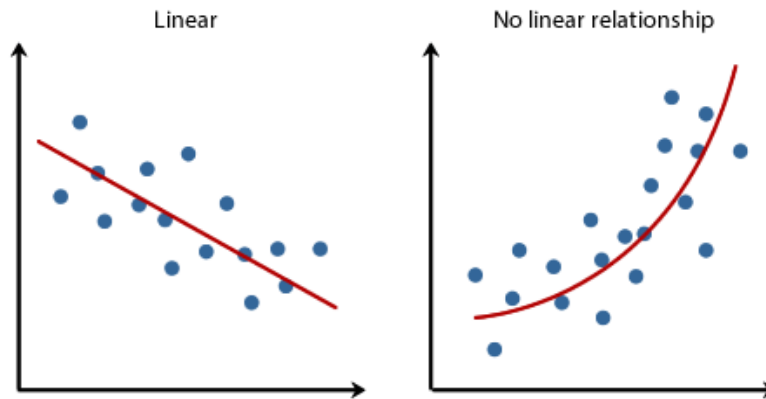
Svaki algoritam strojnog učenja sastoji se od modela, funkcije gubitka i optimizacijskog postupka.

Model

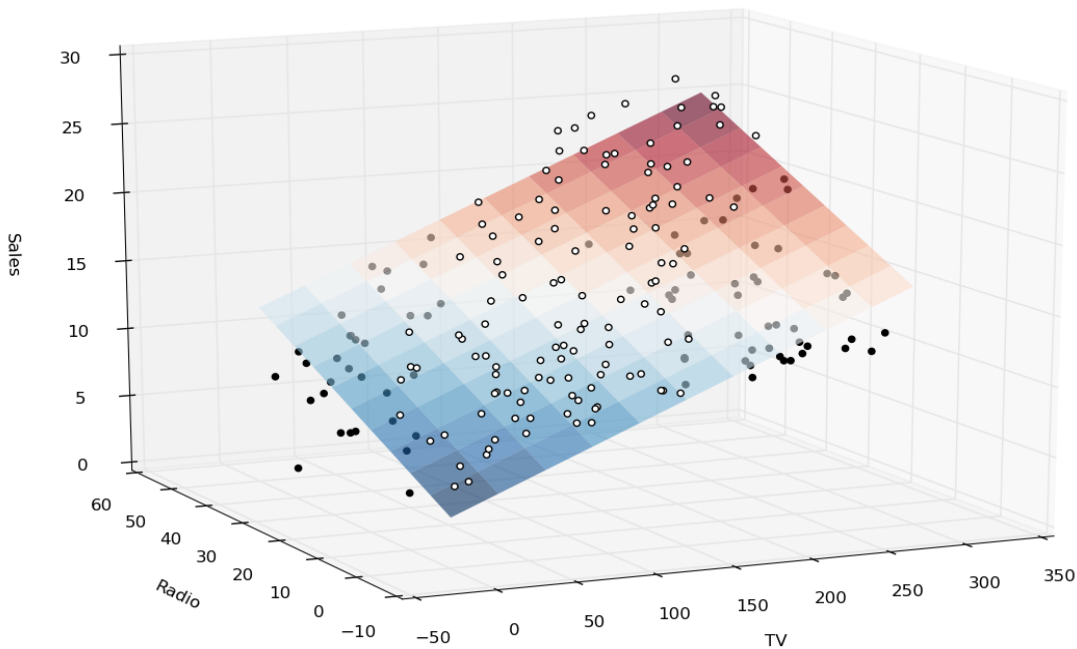
Kod modela regresije, hipoteza h je funkcija parametara $w = (w_0, \dots, w_n)$, a x je skup ulaznih varijabli, odnosno značajki. Postupak učenja regresijskog modela označava procjenu vrijednosti parametara w te se orijentira na minimizaciju ukupne pogreške modela.

Primjeri modela regresije:

1. Univarijatna linearna regresija: $h(x \vee w) = w_0 + w_1 * x_1$
2. Multivarijatna linearna regresija: $h(x \vee w) = w_0 + w_1 * x_1 + w_2 * x_2 + \dots + w_n * x_n$
3. Univarijatna polinomijalna regresija: $h(x \vee w) = w_0 + w_1 * x_1 + w_2 * x_2 + \dots + w_d * x_d (n=1)$
4. Multivarijatna polinomijalna regresija:
 $h(x \vee w) = w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_1 * x_2 + w_4 * x_{12} + w_5 * x_{22} (n=2, d=2)$



Slika 3.4 – Prikaz modela linearne regresije i polinomijalne regresije s dvjema ulaznim varijablama [12]



Slika 3.5 – Prikaz modela multivarijatne regresije s trima ulaznim varijablama [13]

Funkcija gubitka

Postoje razne funkcije gubitaka koje se odabiru ovisno o tome odgovaraju li rješavanju problematike samog postupka te one mjere pogrešku na jednome primjeru. Empirijska pogreška je zbroj gubitaka svih primjera i njome se iskazuje netočnost hipoteze h na skupu podataka. Najčešće se za problem regresije odabire kvadratna pogreška kao funkcija gubitka, a za empirijsku pogrešku srednja kvadratna pogreška.

Kvadratna pogreška kao funkcija gubitka:

$$L(y, h(\mathbf{x}^{(i)}|\theta)) = (h(\mathbf{x}^{(i)}|\theta) - y^{(i)})^2$$

Srednja kvadratna pogreška kao empirijska pogreška:

$$E(h|D) = \mathbb{E}_D[L] = \frac{1}{N} \sum_{i=1}^N L(y^{(i)}, h(\mathbf{x}^{(i)}))$$

gdje $x^{(i)}$ označava primjer, $y^{(i)}$ označava točan izlaz primjera $x^{(i)}$, h označava hipotezu naučenog modela na primjeru $x^{(i)}$, a N označava ukupan broj svih primjera [14].

Optimizacijski postupak

Kao optimizacijski postupak u radu smo koristili gradijentni spust koji je iterativno optimizacijski algoritam za pronalaženje minimuma neke funkcije. Često je korišten s linearnom regresijom, a u praksi je koristan kada imamo veliki skup podataka ili u broju primjera ili broju varijabli koje možda ne bi mogle stati u memoriju.

On funkcionira tako što model daje predviđanja o podacima treniranja i koristi pogrešku na predviđanjima za ažuriranje modela na način da se smanji ukupna pogreška modela. Cilj algoritma je pronaći parametre modela koji minimiziraju pogrešku modela na skupu za treniranje. To čini tako što mijenja model koji ga pomiče uz gradijent ili nagib pogreške prema minimalnoj vrijednosti pogreške. Zbog toga se algoritam naziva „gradijentni spust“ [15]. Stopa učenja koristi se kao faktor skale i parametri se ažuriraju u smjeru minimizacije pogreške.

Gradijentni spust može varirati u smislu veličine skupa primjera koji se koriste za izračunavanje pogreške i zatim za ažuriranje modela. Razlikujemo grupni (engl. *batch*) gradijentni spust, stohastički gradijentni spust i mini-batch gradijentni spust.

Stohastički gradijentni spust

Stohastički gradijentni spust (skraćeno SGD) je varijanta koja izračunava pogrešku modela i ažurira model za svaki primjer u skupu podataka za treniranje.

Prednosti su česta ažuriranja koja odmah daju uvid u izvedbu modela i brzinu poboljšavanja, povećana frekvencija ažuriranja modela može rezultirati bržim učenjem, a

šumovit proces ažuriranja može dopustiti modelu da izbjegne lokalne minimume, npr. preranu konvergenciju.

Nedostaci su sljedeći: proces je računalno skuplji od ostalih zbog prečestih ažuriranja modela; šumovit proces učenja niz gradijent pogreške može otežati algoritmu da se smjesti na pogrešci minimalnoj za model.

Batch gradijentni spust

Batch gradijentni spust je varijanta koja izračunava pogrešku modela za svaki primjer u skupu podataka treniranja, ali samo ažurira model nakon što se svi primjeri za treniranje procjene.

Jedan ciklus kroz cijeli skup podataka za treniranje naziva se epoha treniranja. Zato se kaže da batch gradijentni spust obavlja ažuriranje modela na kraju svake epohe treniranja.

Prednosti su što je računalno učinkovitiji od stohastičkog postupka, smanjena frekvencija ažuriranja rezultira stabilnijim gradijentom pogreške i može rezultirati stabilnijom konvergencijom.

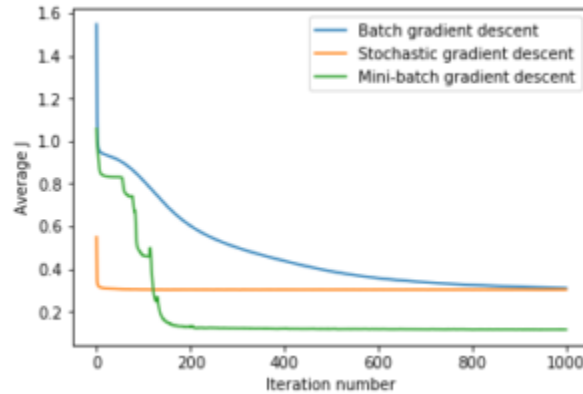
Nedostaci su što stabilniji gradijent pogreške može rezultirati preranom konvergencijom modela s manje optimalnim skupom parametara te ažuriranja modela mogu postati vrlo spora za velike skupove podataka.

Mini-batch gradijentni spust

Mini-batch gradijentni spust je varijanta koja razdvaja skup podataka treniranja u male serije koje se koriste za izračunavanje pogreške modela i ažuriranje parametara modela. Mini-batch nastoji pronaći ravnotežu između robusnosti stohastičkog gradijentnog spusta i učinkovitosti batch gradijentnog spusta. Najčešće je korištena varijanta gradijentnog spusta u području dubokog učenja te smo nju koristili u radu.

Prednosti su što je frekvencija ažuriranja modela veća nego kod batch gradijentnog spusta, što omogućuje bržu konvergenciju izbjegavajući lokalne optimume, a računski je učinkovitiji od stohastičkog gradijentnog spusta.

Nedostaci su što zahtijeva konfiguraciju dodatnog hiperparametra „mini-batch veličina“ za algoritam učenja.



Slika 3.6 – Pregled krivulja i efikasnosti verzija gradijentnog spusta [16]

3.2.5.3. Klasifikacija

Za binarne klasifikacijske probleme koristi se logistička regresija, a za probleme s više od dvije klase koristi se višeklasna logistička regresija.

Logistička regresija

Logistička regresija ili binomna logistička regresija je jedan od najpopularnijih algoritama strojnog učenja za binarnu klasifikaciju zato što je riječ o jednostavnom algoritmu koji dobro funkcionira na širokom rasponu problema.

Ime je dobila po logističkoj funkciji, odnosno sigmoidalnoj funkciji koja se koristi u srži metode logističke regresije. To je krivulja u obliku slova S koja može uzeti bilo koji realni broj i transformirati ga u vrijednost između 0 i 1, ali nikada točno u te vrijednosti [17].

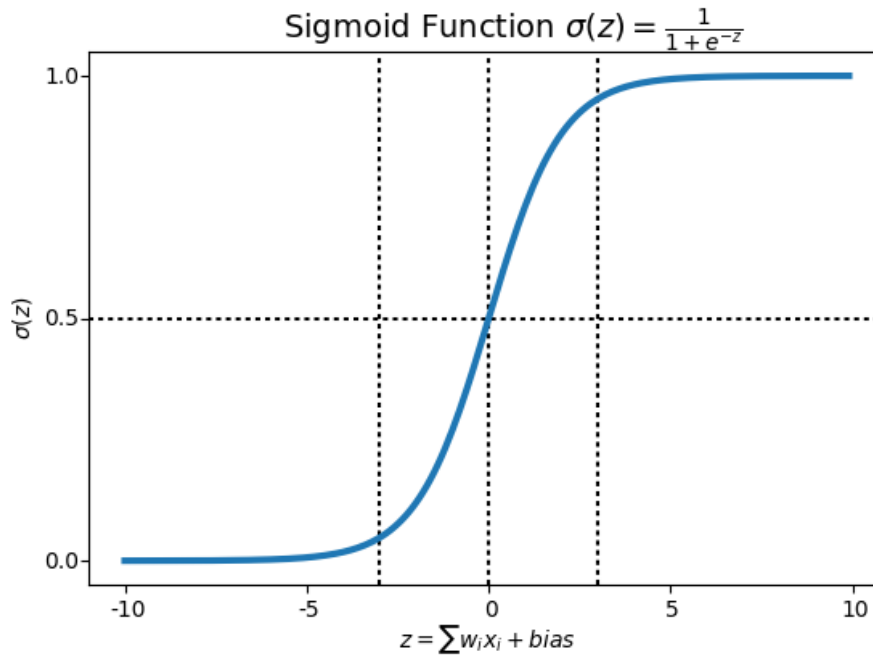
Definicija logističke (sigmoidalne) funkcije:

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)}$$

Model logističke regresije:

$$h(\mathbf{x}|\mathbf{w}) = \sigma(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})) = \frac{1}{1 + \exp(-\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}))}$$

gdje \mathbf{x} označava skup ulaznih varijabli, \mathbf{w} skup parametara modela, h hipotezu naučenog modela, a $\Phi(\mathbf{x})$ označava skup baznih nelinearnih funkcija ulaznih varijabli [18].



Slika 3.7 – Prikaz logističke funkcije [19]

Na slici 3.7 prikazan je graf logističke funkcije. Logistička regresija modelira vjerojatnost zadane klase, odnosno uzima vrijednosti ulaza i daje predviđanje kao vjerojatnost pripadanja zadanoj klasi. Ako je vjerojatnost veća od 0.5, možemo uzeti izlaz kao predviđanje za zadanu klasu (npr. klasa 0), inače je riječ o predviđanju za drugu klasu (npr. klasa 1).

Parametri w se procjenjuju iz podataka i to se vrši pomoću procjene maksimalne log-izglednosti. Najbolji parametri rezultirali bi modelom koji bi predvidio vrijednost vrlo blizu 1 za zadanu klasu i vrijednost vrlo blizu 0 za drugu klasu.

Empirijsku pogrešku definiramo kao negativnu log-izglednost i nazivamo je „pogreška unakrsne entropije“.

$$E(\mathbf{w}|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \left(-y^{(i)} \ln h(\mathbf{x}^{(i)}|\mathbf{w}) - (1 - y^{(i)}) \ln (1 - h(\mathbf{x}^{(i)}|\mathbf{w})) \right)$$

Postupak optimizacije temelji se na minimizaciji empirijske pogreške i za tu problematiku mogu se iskoristiti verzije gradijentnog spusta.

Višeklasna logistička regresija

Višeklasna ili multinomijalna logistička regresija je jednostavno proširenje binomne logističke regresije, a koristi se kada imamo više od dvije klase.

Svaka izlazna varijabla y sadrži *dummy* vrijednosti, odnosno primjenjeno je *one-hot* kodiranje, što znači da je izlazna varijabla proširena na vektor duljine broja klasa i svaka vrijednost sadrži 1 ili 0, ovisno o tome pripada li izlazna varijabla nekoj klasi ili ne. Na mjestu pripadne klase izlazne varijable vrijednost će biti 1, a na ostalima 0.

Zatim se procjenjuje odvojeni model binarne logističke regresije za svaku od tih *dummy* varijabli. Rezultat su modeli binarne logističke regresije kojih ima koliko i broj klasa umanjениh za 1, izračun za svaku klasu, ali bez izračuna za zadanu klasu. Svaki model govori o utjecaju procjenitelja na vjerojatnost uspjeha u toj klasi u usporedbi sa zadanom klasom [20].

Za funkciju modela umjesto sigmoidalne funkcije koristimo softmax funkciju koja je vrlo prigodna jer na izlazu daje distribuciju vjerojatnosti svake klase.

$$h_k(\mathbf{x}; \mathbf{W}) = \frac{\exp(\mathbf{w}_k^T \phi(\mathbf{x}))}{\sum_j \exp(\mathbf{w}_j^T \phi(\mathbf{x}))}$$

Za procjenu parametara w koristimo maksimalnu log-izglednost, a empirijsku pogrešku definiramo kao negativnu log-izglednost i nastojimo je minimizirati.

Pogreška unakrsne entropije:

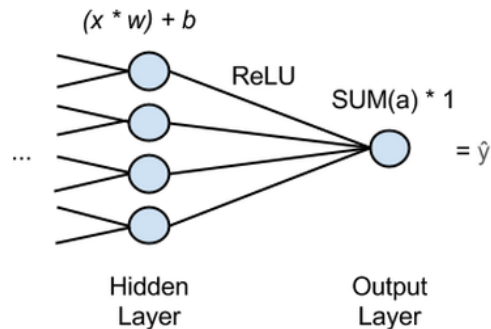
$$E(\mathbf{W}|\mathcal{D}) = - \sum_{i=1}^N \sum_{k=1}^K y_k^{(i)} \ln h_k(\mathbf{x}^{(i)}; \mathbf{W})$$

gdje $\mathbf{x}^{(i)}$ označava primjer, $y_k^{(i)}$ označava točan izlaz primjera $\mathbf{x}^{(i)}$, h_k označava hipotezu naučenog modela na primjeru $\mathbf{x}^{(i)}$ za klasu k , \mathbf{W} matrica parametara modela, h hipotezu naučenog modela, N označava ukupan broj svih primjera, a K broj klasa.

Postupak optimizacije temelji se na minimizaciji empirijske pogreške i za tu problematiku mogu se iskoristiti verzije gradijentnog spusta.

3.2.5.4. Umjetne neuronske mreže za regresiju

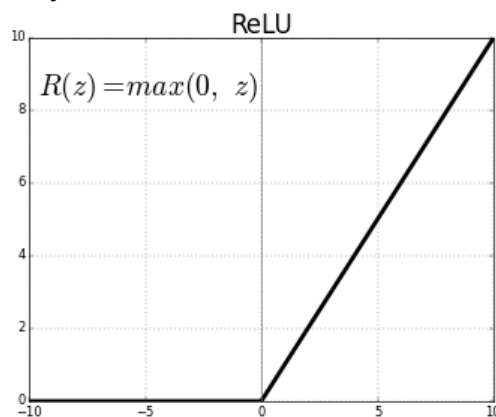
Umjetne neuronske mreže mogu se vrlo lako iskoristiti za regresiju, odnosno za procjenu realnog broja.



Slika 3.8 – Prikaz umjetne neuronske mreže koristeći regresiju [21]

Na slici 3.8 x označava ulaz, odnosno značajke koje se prenose naprijed iz prethodnog sloja mreže. Mnogi x -evi bit će ubačeni u svaki neuron zadnjeg skrivenog sloja i svaki x bit će pomnožen s odgovarajućim parametrom w .

Sumi produkata dodana je pristranost (engl. *bias*) i cijeli izračun je ubačen u aktivacijsku funkciju. U ovom slučaju je korištena ReLU aktivacijska funkcija koja se obično koristi kod regresije. Definirana je kao $f(x) = \max(0, x)$ i vrlo je korisna jer se neće zasititi na plitkim gradijentima kao što se zasite sigmoidalne aktivacijske funkcije. Za svaki neuron u zadnjem skrivenom sloju izračunava se rezultat ReLU funkcije primijenjene na taj neuron, a ti se rezultati zbrajaju i prenose u neuron izlaznog sloja čiji izlaz daje sumu rezultata ReLU funkcija. Na slici 3.9 prikazan je graf ReLU funkcije.



Slika 3.9 – Izgled grafa ReLU funkcije [22]

Neuronska mreža koja obavlja regresiju imat će jedan izlazni neuron, a taj neuron će samo pomnožiti zbroj aktivacija prethodnog sloja za 1, odnosno samo će prenijeti taj zbroj. Rezultat će biti procjenitelj \hat{y} koji procjenjuje mrežu.

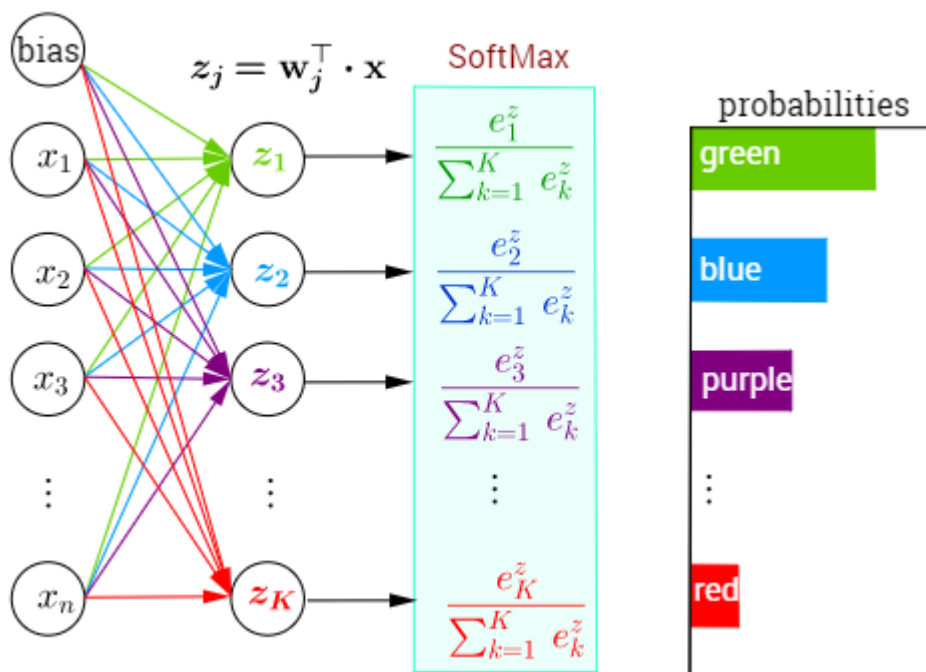
Da bi se obavio *backpropagation* i učila mreža, procjenitelj \hat{y} se uspoređuje s točnim izlazom y i tada se namještaju parametri mreže w tako da ukupna pogreška mreže bude minimalna. Najčešće korištena funkcija gubitka jest srednja kvadratna pogreška.

Backpropagation je algoritam koji podrazumijeva da učenje mreže uključuje određivanje razlike između stvarnog i željenog izlaza, tj. izračun pogreške se vraća unatrag kroz mrežu, s ciljem postizanja najuspješnijeg učenja. Takva je pogreška izračunata za svaki neuron te je upotrijebljena za prilagodbu težinskih parametara. To je ponovljeno toliko puta dok pojedinačna ili ukupna pogreška nije niža od neke granične vrijednosti. Nakon dosezanja te granice završeno je učenje mreže i ona može biti primijenjena za obradu novih ulaza. Taj je algoritam danas najpopularniji postupak koji se primjenjuje u neuronskim mrežama [23].

Na taj način možemo iskoristiti neuronsku mrežu da bismo dobili funkciju koja uzima proizvoljan broj ulaznih varijabli x s izlaznom varijablom y koju pokušavamo predvidjeti. U ovom radu pokušali smo procijeniti starost osobe na temelju ortopantomograma, odnosno procijeniti starost osobe u danima.

3.2.5.5. Umjetne neuronske mreže za klasifikaciju

Umjetne neuronske mreže također se mogu iskoristiti za klasifikaciju, odnosno za procjenu pripadnosti ulaznog uzorka nekoj klasi.



Slika 3.10 – Prikaz umjetne neuronske mreže korištenjem klasifikacije [24]

Na slici 3.10 (x_1, \dots, x_n) označava ulaz, odnosno značajke koje se prenose naprijed iz prethodnog sloja mreže.

Predzadnji sloj s izlazima (z_1, \dots, z_K) sastoji se od broja neurona koliki je broj klasa i svaki neuron zastupa jednu klasu. Za svaki neuron se izračunava suma produkata svih izlaza prethodnog sloja s pripadnim parametrima w te se u sumu dodaje pristranost (engl. *bias*).

Kao zadnji sloj korištena je softmax aktivacijska funkcija koja daje distribuciju vjerojatnosti svake klase. Drugim riječima, svaki izlaz u izlaznom sloju govori o utjecaju procjenitelja na vjerojatnost uspjeha u toj klasi u usporedbi sa zadanom klasom. Softmax funkcija je korištena kao aktivacijska funkcija kod višeklasne logističke regresije, odnosno kod problematike klasifikacije s više od dvije klase.

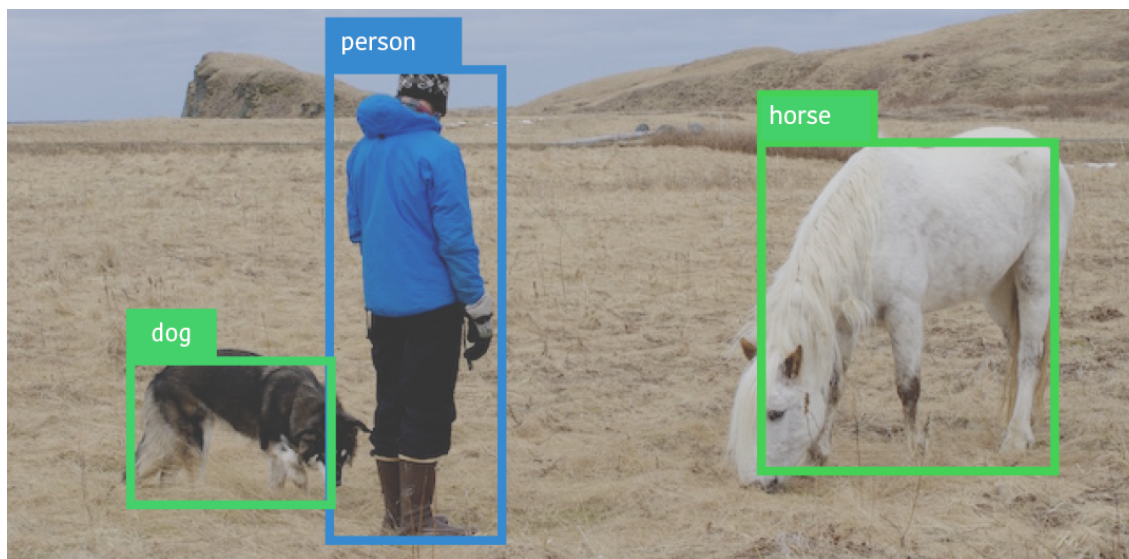
Da bi se obavio *backpropagation* i učila mreža, jednakost klase uspoređuje se s najvećom vjerojatnosti na izlazu softmax funkcije s točnim izlazom y . Zatim se namještaju parametri mreže tako da ukupna pogreška mreže bude minimalna. Najčešće korištena funkcija gubitka je negativna log-izglednost.

Na ovaj način možemo iskoristiti neuronsku mrežu da bismo svaki primjer razvrstali u predviđenu klasu. U ovom radu pokušali smo odrediti spol na temelju ortopantomograma, odnosno provesti klasifikaciju na muški i ženski spol.

3.2.6. Duboke neuronske mreže za detekciju i klasifikaciju

Detekcija objekata je zadatak iz područja računalnog vida kojemu je cilj pronalazak instanci objekata određene klase na digitalnim slikama i videozapisima. Detekcija objekata objedinjuje dva problema računalnog vida koji dobivaju poseban značaj jednom kada ih kombiniramo. Prvi problem nazivamo lokalizacijom, tj. određivanjem konkretne i precizne lokacije objekta na slici. Drugi problem je klasifikacija, a cilj je odrediti semantički značaj objekta (je li locirani objekt automobil, avion, osoba itd.). Sustav za detekciju objekata na slici će locirati sve željene objekte te odrediti njihove pripadne klase.

Ovakvi sustavi imaju vrlo široku primjenu. Imaju važnu ulogu u područjima poput detekcije ljudskog lica, detekcije pješaka i automobila kod autonomnih vozila, u industrijskim sustavima te brojnim drugim područjima.



Slika 3.11 – Primjer detekcije objekata¹

Sustav za detekciju objekata možemo promatrati i izvesti kao dvije zasebne komponente spojene u jednu. Prva komponenta rješava problem lokalizacije, tj. ima ulogu pronalaženja i predlaganja mogućih lokacija objekata. Druga komponenta rješava problem klasifikacije, odnosno na temelju predložene lokacije određuje o kojem se točno objektu radi. Za svaki objekt na slici računaju se koordinate, pripadna klasa te vjerojatnost koja opisuje sigurnost detekcije.

Poseban naglasak stavlja se na problem lokalizacije objekta na digitalnoj slici ili videozapisu. Problem se popularno naziva „predlaganje regija“ (engl. *region proposal*), a neuronsku mrežu koja rješava takav problem nazivamo neuronskom mrežom za predlaganje regija (engl. *region proposal network*).

S obzirom na taj problem, algoritme detekcije temeljene na dubokim konvolucijskim neuronskim mrežama možemo podijeliti u dvije skupine:

1. Algoritmi temeljeni na predlaganju regija (engl. *region based algorithms*), poput Region-based Convolutional Neural Network (skraćeno R-CNN), Fast Region-based Convolutional Neural Network R-CNN (skraćeno Fast R-CNN) te Faster Region-based Convolutional Neural Network (skraćeno Faster R-CNN);
2. Algoritmi temeljeni na regresiji lokacije (engl. *regression based algorithms*), poput You Only Look Once (skraćeno YOLO) i Single Shot Multibox Detector (skraćeno SSD).

U ovom radu obuhvatit će se obje skupine algoritama. Kao primjer algoritama temeljenih na predlaganju regija objasniti će se metoda Faster R-CNN, a kao primjer algoritama temeljenih na regresiji lokacije objasniti će se metoda YOLO.

¹ <http://mobileml.com/images/posts/yolo-objects.png>

Od velike nam je važnosti da naš sustav za detekciju objekata možemo trenirati s kraja na kraj (engl. *end-to-end training*). Trenirati duboku neuronsku mrežu s kraja na kraj znači optimizirati težine simultano, u jednom hodu, uzimajući u obzir piksele na ulazu neuronske mreže te njihovo direktno mapiranje na izlazu. Neuronska mreža koja nema mogućnost treniranja s kraja na kraj zahtijevat će nekoliko zasebnih postupaka treniranja prije nego što dobijemo konačan sustav.

Duboka konvolucijska neuronska mreža temeljena na predlaganju regija kao ulaz prima sliku (bilo koje veličine) te na izlazu daje listu prijedloga lokacija mogućih objekata, a svaki prijedlog sadrži vjerojatnost da se baš u tom području nalazi neki objekt (engl. *objectness score*). Za generiranje prijedloga slika prvo prolazi kroz konvolucijske slojeve, a zatim se koristi metoda pomičnog prozora nad obilježjima konvolucijskih slojeva (engl. *feature maps*). Neuronska mreža za predlaganje regija može se trenirati s kraja na kraj korištenjem *backpropagation* algoritma uz bilo koji optimizacijski algoritam (najčešće gradijentni spust).

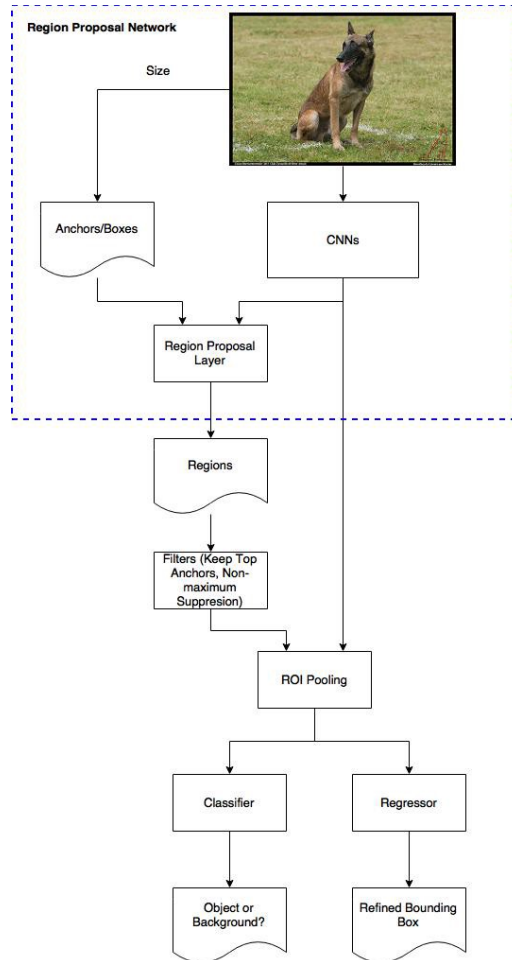
Duboka konvolucijska neuronska mreža temeljena na regresiji kao ulaz također prima cijelu sliku (bilo koje veličine) te na izlazu daje listu prijedloga lokacija mogućih objekata, a također sadrži vjerojatnost da se baš u tom području nalazi neki objekt (engl. *objectness score*). Slika se dijeli na ćelije te svaku ćeliju promatramo kao zasebnu sliku u kojoj tražimo objekte. Za traženje granica objekata koristi se metoda regresije. Neuronska mreža temeljena na regresiji lokacije također se može trenirati s kraja na kraj korištenjem *backpropagation* algoritma uz bilo koji optimizacijski algoritam (najčešće gradijentni spust).

3.2.7. Faster R-CNN

Faster R-CNN (*Faster Regional Convolutional Neural Network*) [25] je treća u nizu R-CNN arhitektura koje koriste konvolucijske neuronske mreže kako bi lokalizirale i klasificirale objekte na slikama. Za razliku od svojih prethodnika, Faster R-CNN koristi mapu konvolucijskih obilježja koju generira zadnji konvolucijski sloj. Posljedica toga je znatno brže generiranje prijedloga, jer se ne koriste dodatni resursi za selektivno pretraživanje [26].

Arhitektura

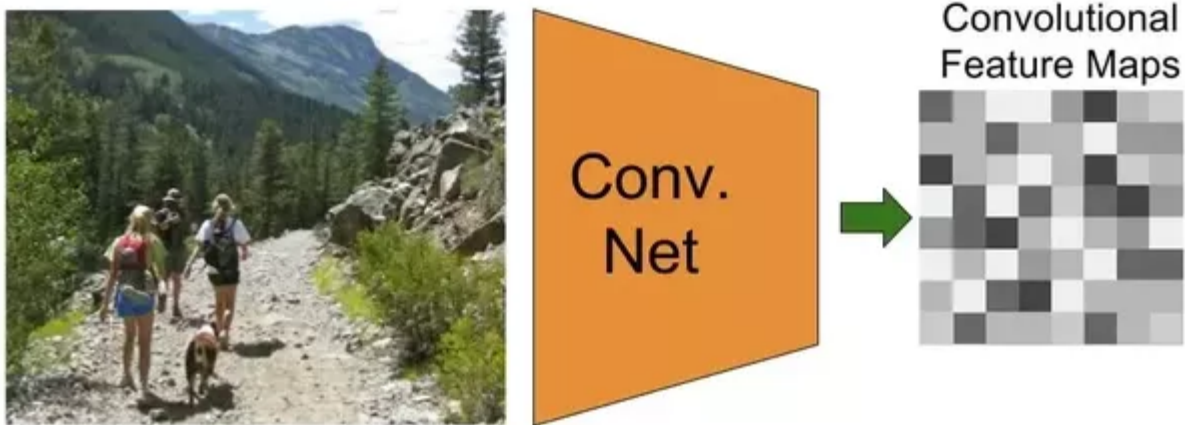
Arhitekturu generalno možemo podijeliti na konvolucijsku mrežu za predviđanje regija te Fast R-CNN detektor koji koristi predviđene regije kako bi detektirao te klasificirao objekte. Moglo bi se reći da mreža za predviđanje regija govori Fast R-CNN modulu gdje gledati. Stoga je Faster R-CNN zapravo kombinacija mreže za predviđanje regija i svoga prethodnika, Fast R-CNN-a [27].



Slika 3.12 – Faster R-CNN arhitektura

Region proposal network

Cilj RPN-a je predvidjeti područje slike na kojemu je velika mogućnost nalaska objekta. Ona kao ulaz prima sliku bilo koje veličine te generira pravokutna područja i vjerojatnost nalaženja objekta u toj regiji (engl. *objectness score*) za svaki od njih. Slika se pri ulazu u mrežu prvo obrađuje u konvolucijskoj mreži. Ta mreža je već trenirana na raznim objektima pa su njezine težine već inicijalizirane. Izlaz konvolucijske mreže je niz mapa konvolucijskih obilježja (engl. *convolutional feature map*).



Slika 3.13 – Izlaz inicijalne konvolucijske mreže²

Sljedeći korak je generiranje područja pomoću informacija iz konvolucijske mape. Broj piksela konvolucijske mape je za r puta manji od ulazne slike. Varijabla r predstavlja omjer uzorkovanja (engl. *subsampling ratio*) te određuje širinu i visinu mape. Ona nije fiksno određena, već je karakteristika mreže i ovisi o broju konvolucijskih i *pooling* slojeva. Tako je, recimo, za mrežu VGG [28] omjer uzorkovanja jednak 16.

Zbog te karakteristike možemo definirati „predloške“ (engl. *anchors*) [29] na svakoj poziciji mape, što odgovara svakom r -tom pikselu na originalnoj slici. Na svakoj poziciji generira se 9 *anchora*, kombinirajući tri različite veličine i tri razmjera proporcija (engl. *aspect ratio*). Iako se tim postupkom dolazi do velikih količina područja koja se moraju obraditi, štedi se na resursima jer se u potpunosti koriste konvolucijske mreže, bez ikakvih dodatnih postupaka koji troše resurse. Također, implementacija smanjuje složenost uklanjajući *anchore* koji nisu u potpunosti unutar slike.

U svakom se ciklusu treniranja RPN-a priprema $W \times H \times 9 - X$ *anchora*, gdje su W i H širina i visina konvolucijske mape, dok je X broj *anchora* koji nisu u potpunosti unutar slike.

Za svaki *anchor* računa se „presjek kroz unija“ (engl. *Intersection-over-Union*, skraćeno *IoU*) [30] kao omjer presjeka *anchora* i manualno zadanog područja u kojemu se nalazi objekt te unije tih dvaju područja. *Anchorima* s najvećim *IoU* -om za pojedino označeno područje te onima s *IoU* -om većim od 0,7 dodjeljujemo pozitivnu oznaku. *Anchorima* s *IoU* -om manjim od 0,3 određujemo negativnu oznaku. Ostali *anchori* nemaju utjecaja na treniranje mreže jer se smatraju nepouzdanima. Za svaku seriju treniranja nasumično se odabire 256 *anchora* uz uvjet da mreža po mogućnosti odabire jednak broj pozitivnih i negativnih *anchora*.

Anchori se često preklapaju zbog veličina uglavnom većih od međusobnih udaljenosti najbližih *anchora*. Posljedica toga je i neželjeno preklapanje u prijedlozima. Kako bismo izbjegli

² <https://qph.fs.quoracdn.net/main-qimg-3fa20d817a121d31c641978114ff6f5d>

preklapanje prijedloga, suvišne prijedloge uklanjamo algoritmom *Non-Maximum Supression* (skraćeno NMS) [31]. Algoritam filtrira prijedloge koristeći predefiniranu *IoU* donju granicu.

Funkcija gubitka RPN-a računa se kao linearna kombinacija gubitka regresije te gubitka klasifikacije. Regresija u ovom slučaju predviđa koordinate, dok klasifikacija određuje sadržava li *anchor* objekt. Gubitak klasifikacije određen je metodom binarne unakrsne entropije (engl. *binary cross entropy*).

Za razliku od klasifikacijskog dijela, regresijski gubitak određen je samo pozitivnim *anchorima* te se računa pomoću jednostavnog L1 gubitka (engl. *L1 loss*) [32]. Ukupan gubitak je tada linearna kombinacija s težinskim faktorom λ jednakim 10 uz regresijski gubitak. S tim definicijama možemo zapisati:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

Pojašnjena varijabli formula:

- i – indeks *anchora* u seriji
- p_i – vjerojatnost da *anchor* sadrži objekt
- p_i^* – 1 za pozitivan *anchor* te 0 za negativan
- t_i – četverodimenzijalni vektor koji reprezentira predviđene koordinate
- t_i^* – vektor s koordinatama najbližeg manualno označenog prozora

Važno je napomenuti da L_{cls} označava klasifikacijski gubitak, L_{reg} regresijski, a p_i^* ispred regresijskog gubitka znači da funkcija uzima u obzir samo pozitivne *anchore*.

Koristeći ove podatke, algoritam propagacije pogreške unazad smanjuje pogrešku te optimizira mrežu. RPN se u praksi može koristiti i kao zasebna mreža za klasificiranje jedne vrste objekata na slikama gdje nam „objectness score“ može poslužiti kao vjerojatnost pronalaska objekta te klase.

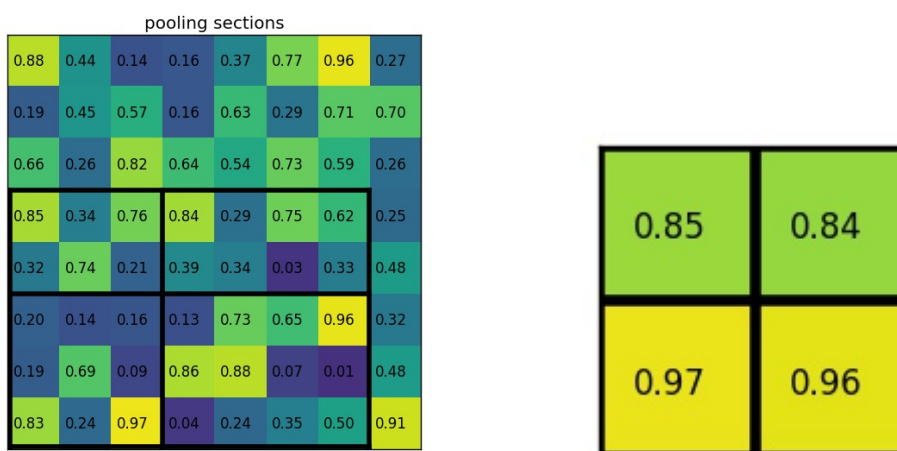
U našem slučaju RPN se koristi kako bi predložio mreži za klasifikaciju gdje je najveća mogućnost pronalaska objekta u slici. Budući da predloške različitih veličina želimo predati konvolucijskoj mreži fiksne veličine, moramo ih skalirati bez prevelikog gubljenja informacije. Tehnika kojom se to postiže je „udruživanje po interesnim regijama“ (engl. *Region of Interest pooling*, skraćeno *ROI pooling*) [33].

Region of Interest (RoI) pooling

RoI pooling je tehnika kojom skaliramo originalne prijedloge na prijedloge jednakih veličina. Time omogućujemo korištenje fiksne mreže za klasifikaciju te treniranje mreže s kraja na kraj. Eksperimentalno je dokazano da se takvim treniranjem postižu 25-50% brže treniranje od treniranja obaju modula zasebno. *RoI pooling* je prvi put uveden u mreži Fast R-CNN te joj je omogućio puno bržu klasifikaciju s obzirom na originalni R-CNN. Faster R-CNN koristi identično rješenje kao i njegov prethodnik.

RoI pooling sloj kao ulaz prima mape konvolucijskih obilježja generirane u početnoj konvolucijskoj mreži te prijedloge koje je generirao RPN. Iz mape se uzimaju pozicije koje je RPN označio kao područja s velikim izgledom postojanja objekta. Dobiveni su pravokutnici koji se dijele na fiksni broj dijelova.

U slučaju Faster R-CNN-a, prijedlozi se dijele na po mogućnosti 49 jednakih dijelova uz 7 podjela širine te 7 podjela visine. Svaki od tih 49 dijelova sadrži više vrijednosti iz originalne mape. Nad tim vrijednostima koristi se maksimalno sažimanje (engl. *max pooling*). *Max pooling* zapravo odabire najveću vrijednost iz skupa vrijednosti te generira novu mapu veličine 7×7 za svaki prijedlog. Te mape, sada jednakih veličina, propagiraju se dalje u sljedeći modul mreže, Fast R-CNN detektor.



Slika 3.14 – *RoI pooling* za regiju označenu na lijevoj slici³ uz prozor 2×2

Fast R-CNN detektor

Drugi modul Faster R-CNN arhitekture čini Fast R-CNN detektor. Fast R-CNN je poboljšana implementacija originalne *Region-based convolutional neural network* (skraćeno R-CNN). Nedostatak Fast R-CNN-a je velika potrošnja resursa na generiranje prijedloga metodom selektivnog pretraživanja. Faster R-CNN to rješava efikasnom RPN mrežom koristeći već

³ https://upload.wikimedia.org/wikipedia/commons/d/dc/RoI_pooling_animated.gif

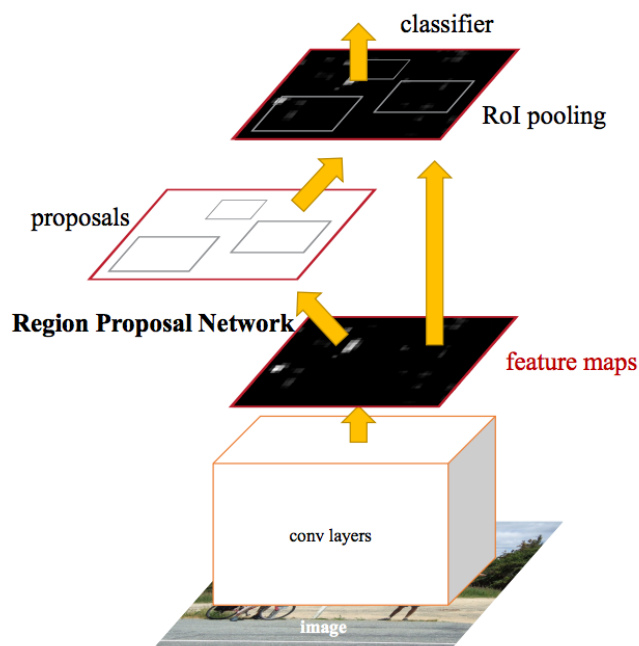
spomenute *anchors*. Budući da je generiranje prijedloga riješeno na učinkovitiji način, možemo iskoristiti preostali dio Fast R-CNN mreže kako bismo klasificirali i podesili predviđene regije.

Stoga se uloga drugog modula može podijeliti na dva dijela:

- Klasificiranje prijedloga u jednu od ponuđenih klasa ili klasu pozadina za loše prijedloge;
- Podešavanje graničnog okvira prijedloga s obzirom na predviđenu klasu.

Može se zaključiti da i drugi modul, kao i RPN, ima klasifikacijski i regresijski gubitak. Konačni dio mreže su dva potpuno spojena sloja (engl. *fully connected layer*) koji generiraju klasu prijedloga te koordinate njegovog graničnog okvira. Treniranje drugog modula stoga je vrlo slično treniranju RPN-a. Ukupan gubitak je kombinacija klasifikacijskog *multiclass cross entropy* gubitka i regresijskog *smooth L1* gubitka. Ovdje se kao jedna od ponuđenih klasa određuju prijedlozi s *IoU* većim od 0,5, dok se kao pozadina određuju prijedlozi s *IoU* između 0 i 0,5.

Za razliku od prvog modula, prijedloge koji nemaju presjek s manualno označenim objektom ignoriramo. Pretpostavlja se da je RPN ispravno generirao prijedloge, pa se pokušava usredotočiti na rješavanje težih primjera. Svaka serija treniranja sadrži 64 prijedloga od kojih su do 25% pozitivno klasificirani. Ukupan gubitak drugog modula zatim se izračuna kao linearna kombinacija regresijskog i klasifikacijskog gubitka.



Slika 3.15 – Zadnji modul prima podatke iz RPN-a te originalne mape obilježja⁴

⁴ https://cdn-images-1.medium.com/max/1600/0*_nNI03ESXm2P6YXO.

3.2.8. YOLO

YOLO je jedan od trenutno najmodernijih modela dubokih konvolucijskih neuronskih mreža za detekciju objekata. Arhitektura YOLO neuronske mreže razvijena je na Sveučilištu u Washingtonu [34] s ciljem da se poboljša vremenska i prostorna složenost postojećih algoritama za detekciju. Autori su odlučili dizajnirati sustav koji spaja probleme lokalizacije i klasifikacije u jedinstvenu duboku konvolucijsku neuronsku mrežu. Naziv YOLO proizašao je upravo iz činjenice da ovaj model u jednom prolazu obavlja zadatke lokalizacije i klasifikacije (engl. *You Only Look Once*).

Dosadašnji modeli za detekciju objekata prenamjenjivali su klasifikacijske neuronske mreže za rješavanje problema lokalizacije. Kako bi detektirali objekt, takvi sustavi uzastopno bi pokretali klasifikator na raznim lokacijama unutar slike dok ne pronađu objekt. Moderniji sustavi poput Faster RCNN koriste RPN za lokalizaciju objekata. Iako je daleko napredniji, Faster RCNN je sustav s vrlo složenom arhitekturom koju je teško i sporo optimizirati s obzirom na to da svaku komponentu moramo trenirati zasebno.

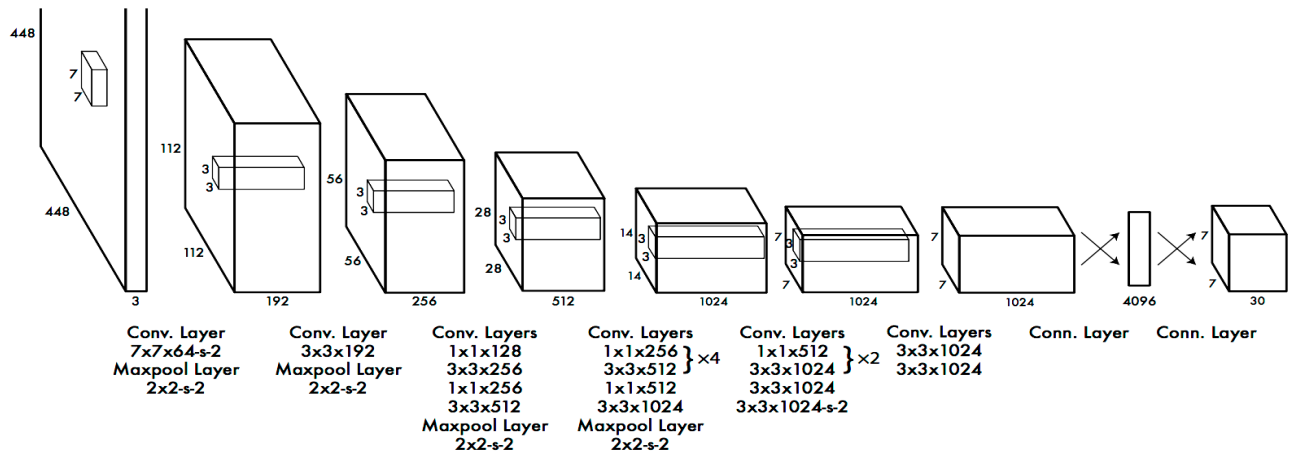
Unificirana arhitektura modela YOLO donosi ekstremno ubrzanje. Bazični YOLO model obrađuje oko 45 slika po sekundi (engl. *frames per second*) na Titan X grafičkom procesoru. Malo manje složena verzija YOLO mreže nazivana Fast YOLO obrađuje oko 155 slika po sekundi, pri čemu je prosječna preciznost i dalje oko dva puta veća od ostalih algoritama za detekciju. Kod ovakve arhitekture više nemamo dvije funkcije gubitka, već samo jednu, što znači da višestruko povećavamo brzinu optimiranja.

Druga velika prednost YOLO modela je da neuronska mreža „vidi“ širi kontekst slike, tj. implicitno se ugrađuju podaci o okolini objekta. To je zbog činjenice da YOLO vidi cijelu sliku tijekom treniranja, a ne samo određene dijelove koje predloži neka druga mreža za lokalizaciju, kao što je slučaj kod Faster RCNN-a.

Arhitektura

Inspiracija za arhitekturu ove neuronske mreže je model za klasifikaciju naziva GoogLeNet [35]. Neuronska mreža se sastoji od 24 konvolucijska sloja te 2 potpuno povezana sloja. Ranije spomenuta neuronska mreža manje složenosti pod nazivom Fast YOLO sastoji se od 9 konvolucijskih slojeva s manjim brojem filtera u slojevima.

Prvih 20 slojeva modela YOLO bavi se ekstrakcijom podataka sa slike, a dobiveni oblik podataka nazivamo „mape konvolucijskih značajki“. Konvolucijske mape sadrže najbitnije informacije sa slike. Daljnji slojevi (4 konvolucijska i 2 potpuno povezana) bave se lokalizacijom (regresijski problem određivanja okvira) te računanjem vjerojatnosti svake pojedine klase. Dimenzija izlaza bit će $S \times S \times (B * 5 + C)$. Na slici 3.16 prikazana je arhitektura YOLO modela.



Slika 3.16 – Arhitektura YOLO modela⁵

Princip rada

Slijedi opis funkcioniranja YOLO modela.

Sliku dijelimo na mrežu veličine $S \times S$. Ako se centar objekta nalazi u ćeliji X , onda kažemo da je ćelija X odgovorna za detekciju toga objekta. Svaka ćelija „predviđa“ B okvira i za svaki okvir računamo vjerojatnost da se unutar okvira nalazi objekt (engl. *objectness score*). Izračunata vjerojatnost govori nam koliko je model siguran da se unutar okvira nalazi neki objekt te koliko je siguran u točnost granica.

Ta vjerojatnost definira se kao: $P(\text{object}) * IoU_{pred}^{truth}$

Cilj je dobiti model neuronske mreže koji daje vjerojatnost što bližu vrijednosti IoU_{pred}^{truth} ako se unutar okvira stvarno nalazi objekt, odnosno vjerojatnost koja teži prema nuli ako objekta nema.

Svaki okvir (engl. *bounding box*) sadržavat će pet podataka:

1. x – položaj centra objekta na osi apscisa
2. y – položaj centra objekta na osi ordinata
3. *širina* – širina objekta relativno stvarnoj širini slike
4. *visina* – visina objekta relativno stvarnoj visini slike
5. *vjerojatnost* – vjerojatnost da se u okviru nalazi objekt

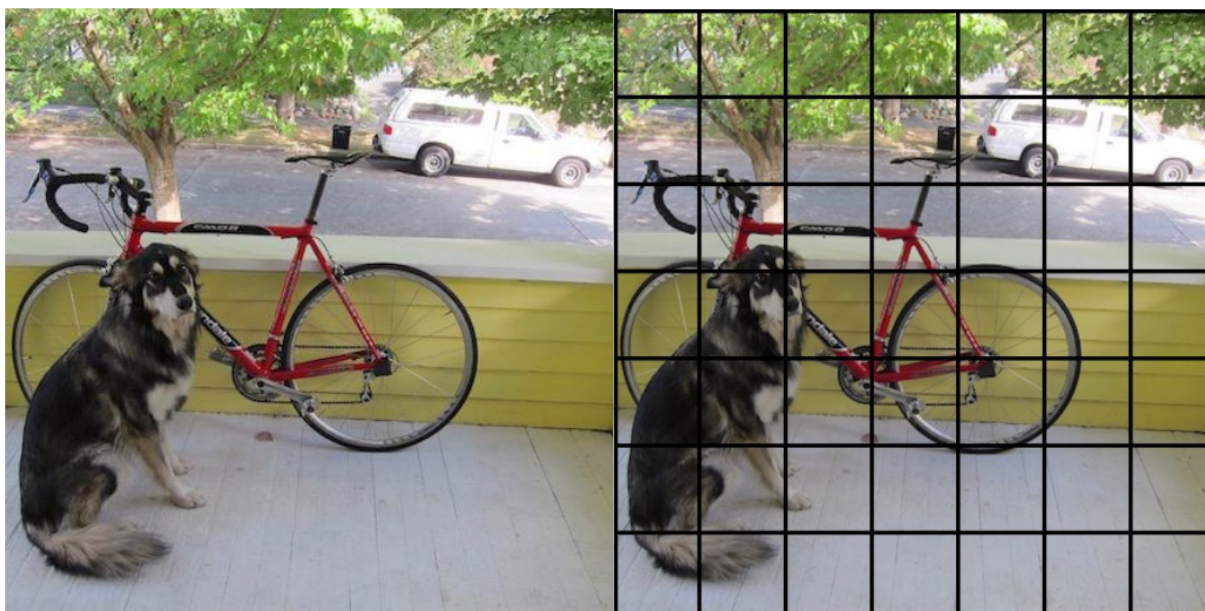
Sljedeći korak je računanje uvjetnih vjerojatnosti pojedine klase unutar svake ćelije. Svaka ćelija računa C vjerojatnosti, tj. za svaku klasu po jednu vjerojatnost, bez obzira na broj okvira B . Ta vjerojatnost definirana je kao $Pr(Class_i \vee Object)$.

⁵ https://cdn-images-1.medium.com/max/1600/1*ZbmrSjQJW-Lp72C5KoTnzUg.jpeg.

Jednom kada imamo uvjetne vjerojatnosti pojedinih klasa za svaku ćeliju, množimo svaku od dobivenih uvjetnih vjerojatnosti s IoU pojedinog okvira unutar ćelije. Formula koju koristimo za izračun: $Pr(Class_i \vee Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth}$.

Odbacujemo sve okvire kod kojih je rezultat množenja s uvjetnom vjerojatnosti klase bio manji od 0,3. Također, kako postoji mogućnost pojave više okvira koji detektiraju jedan te isti objekt, koristi se algoritam za odstranjivanje duplikata koji mjeri preklapanje okvira koji predstavljaju istu klasu (engl. *non-maximum suppression algorithm*, skraćeno NMS).

Na sljedećem primjeru⁶ pokazat ćemo prethodno opisane postupke.

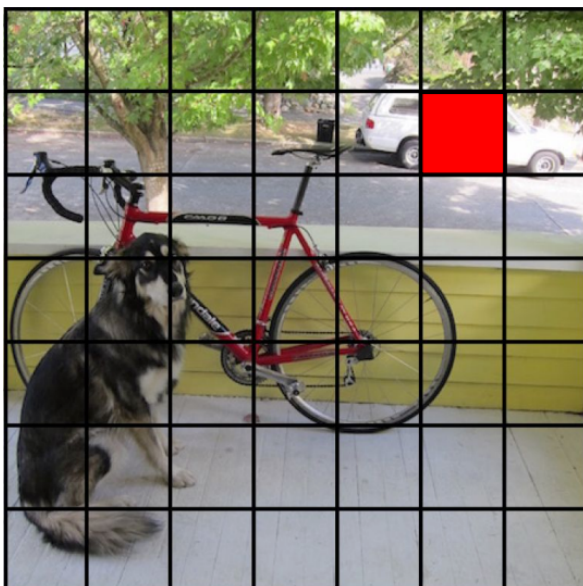


Slika 3.17 – Slika prije detekcije

Slika 3.18 – Prikaz podjele slike na ćelije

Na lijevoj slici nalaze se pas, bicikl te automobil u pozadini. Želimo iskoristiti YOLO model kao sustav detekcije nad navedenim objektima. Prvi korak je podjela slike na ćelije. U ovom slučaju sliku dijelimo na mrežu dimenzija 7×7 .

⁶ Sve slike preuzete su s <https://pjreddie.com/darknet/yolov2/>.

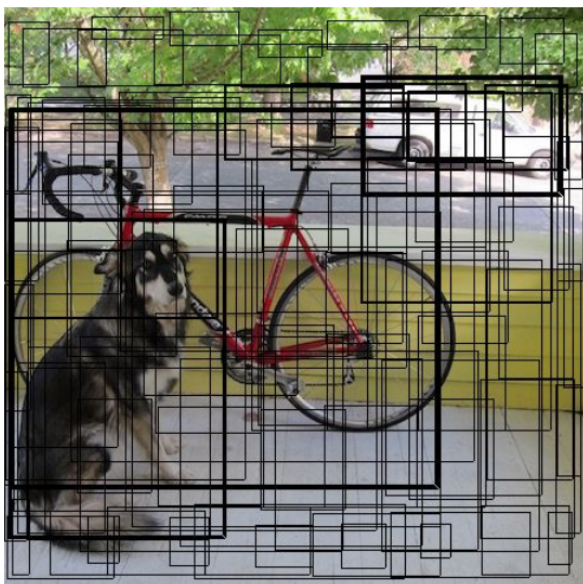


Slika 3.19 – Prikaz ćelije u kojoj tražimo objekte

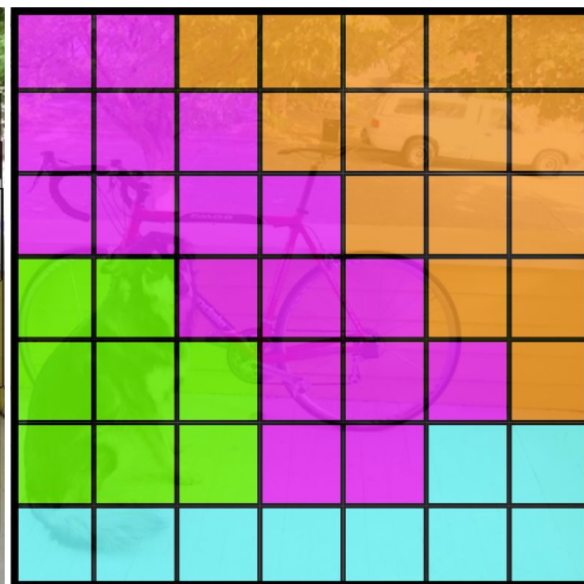


Slika 3.20 – Rezultat pretrage dvaju okvira

Za svaku ćeliju predviđamo dva okvira te računamo vjerojatnost da se u ćeliji nalazi objekt. Na lijevoj slici crvenom bojom prikazana je ćelija koja je trenutno u fokusu. Na desnoj slici prikazani su okviri koje je sustav detekcije predvidio za dotičnu ćeliju.

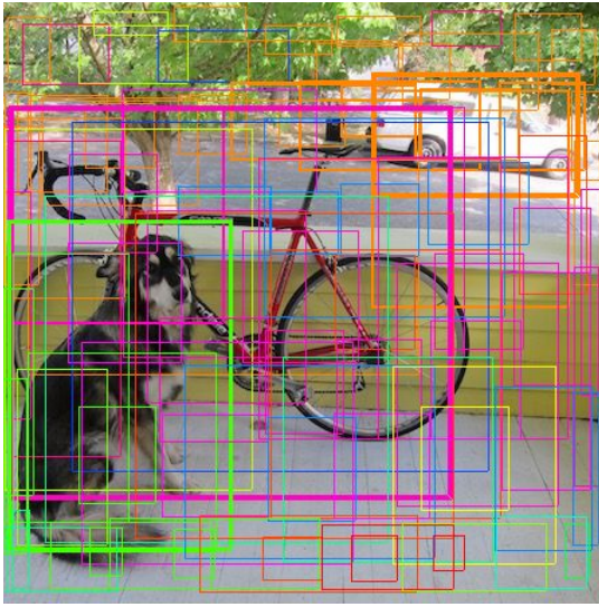


Slika 3.21 – Prikaz svih predviđenih okvira

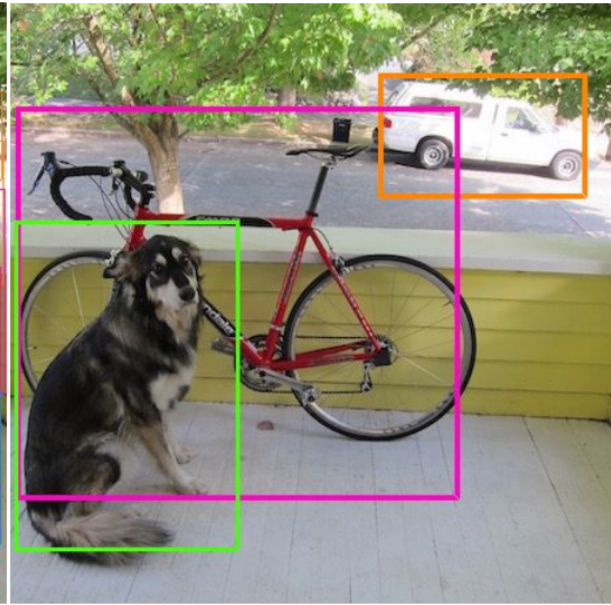


Slika 3.22 – Mapa vjerojatnosti klasa

Na lijevoj slici prikazano je stanje nakon obilaska svih ćelija. Za svaku ćeliju generirali smo 2 okvira, što ukupno daje 98 okvira. Desna slika prikazuje rezultat računanja uvjetnih vjerojatnosti za svaku ćeliju, pri čemu klasa s najvećom vjerojatnošću reprezentira ćeliju. Prikaz na desnoj slici još nazivamo mapom vjerojatnosti klasa (engl. *class probability map*).



Slika 3.23 – Prikaz potencijalnih objekata



Slika 3.24 – Rezultat nakon primjene NMS-a

Lijeva slika prikazuje stanje nakon množenja uvjetne vjerojatnosti pojedine klase unutar ćelije s IoU pojedinog okvira unutar te iste ćelije. Dosad smo znali samo da okvir predstavlja objekt neke klase, ali ne i koje. Nakon množenja s uvjetnim vjerojatnostima klasa znamo o kojoj se točno klasi radi. Filtriramo rezultate manje od 0,3 te odstranjujemo duplikate korištenjem NMS algoritma. Rezultat je prikazan na desnoj slici.

4. Rezultati

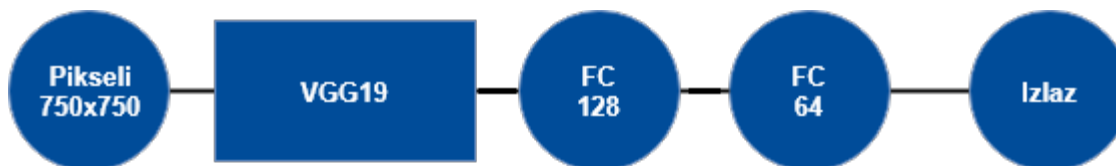
Naša implementacija u pozadini koristi Tensorflow biblioteku [36]. Tensorflow je jedna od najpopularnijih te najrazvijenijih *open-source* programskih biblioteka za rad s visoko performansnim računanjem. Tensorflow je razvio Google Brain tim za internu upotrebu, a 2015. godine biblioteka je objavljena za javno korištenje. Tensorflow je napisan u programskim jezicima Python i C++ te koristi CUDA [37] sučelje za paralelno programiranje. Podržava paralelan rad na NVIDIA grafičkim procesorima i iz tog razloga se uvelike koristi u svrhe strojnog učenja. Temelji se na paradigmi protoka podataka (engl. *dataflow*) koja omogućuje mnogo bržu obradu podataka nego konvencionalna paradigma upravljačkog toka (engl. *controlflow*) korištena na procesorima.

Koristili smo i biblioteku Keras [38] koja nudi jednostavnije sučelje za programiranje dubokih neuronskih mreža, dizajniranu s idejom brzih eksperimenata. Keras u pozadini radi s prethodno spomenutom Tensorflow bibliotekom.

Također, koristili smo i platformu Docker [39] za izolaciju poslova treniranja od OS-a te za osiguravanje reproducibilnih uvjeta. Korišten je službeni Docker image koji podržava GPU implementaciju Tensorflow biblioteke. Svi modeli neuronskih mreža trenirani su na serveru s dvama NVIDIA Titan X grafičkim procesorima koji su pribavljeni putem donacija u NVIDIA GPU Grant programu.

4.1. Procjena dobi

Za problem procjene dentalne dobi dizajnirali smo duboku konvolucijsku neuronsku mrežu kao spoj već postojeće VGG19 mreže [40] i potpuno povezanih slojeva koje smo sami dodali. Problem rješavamo metodom regresije. Mreža na ulazu prima sliku dimenzija 750×750 piksela.



Slika 4.1 – Prikaz arhitekture

Na prethodnoj slici prikazana je arhitektura neuronske mreže za procjenu dobi. Ulaz za piksele spojen je na VGG19 neuronsku mrežu za ekstrakciju konvolucijskih značajki. Izlaz VGG19 mreže spajamo na dva potpuno povezana spoja, sa 128 i 64 neurona, respektivno. Koristimo ReLu aktivacijsku funkciju na izlaznom sloju.

Zbog zakrivljenosti distribucije ulaznog skupa slika odlučili smo trenirati mrežu na slikama čiji je podatak o godinama u intervalu [20, 50]. Nakon filtriranja slika ostalo nam je 3148 slika koje smo dalje koristili za treniranje modela. Također, kako bismo spriječili eventualnu stagnaciju optimizacije, koristimo mehanizam za reduciranje stope učenja s faktorom smanjenja u iznosu 0,8.

Neki od korištenih hiperparametara su:

- Konstanta učenja: 0.001
- Optimizator: Adam s parametrima $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=1e-08$
- Broj epoha: 100
- Batch size: 10

Nakon treniranja testirali smo model nad testnim skupom od 39 slika čiji je podatak o godinama u intervalu [20, 50] te smo dobili rezultate s prosječnom apsolutnom pogreškom od 4,9 godina. Nastavljamo raditi na ovom problemu jer smatramo da je moguće ostvariti mnogo bolje rezultate. Do trenutne stagnacije dolazi zbog male baze slika i nedovoljnih resursa memorije za treniranje.



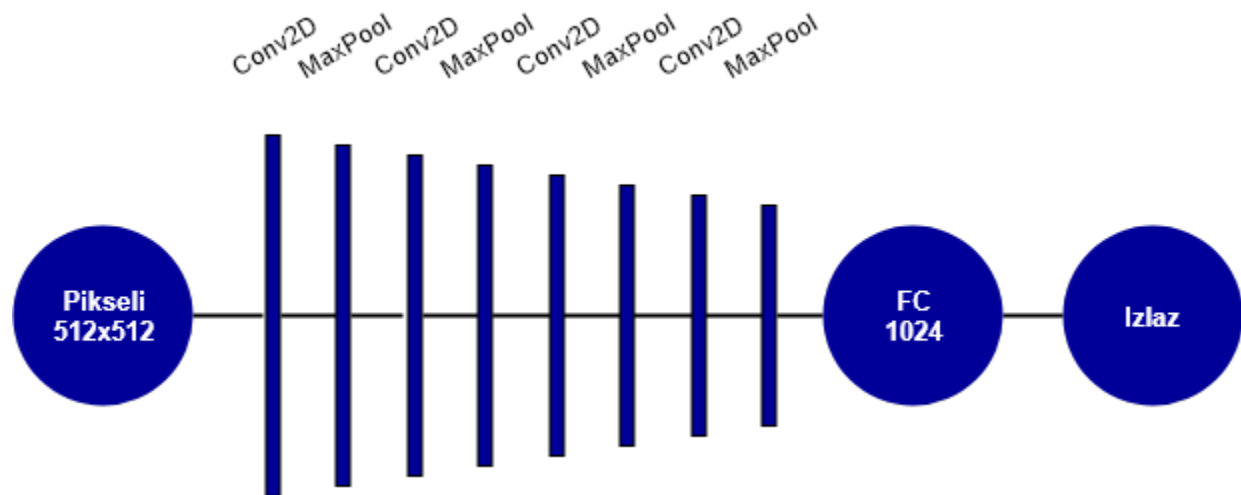
Slika 4.2 – Primjer slike s koje je vrlo teško utvrditi starost osobe

Slika 4.2 primjer je ortopantomograma za koji mreža radi iznadprosječne pogreške pri procjeni dobi. Vidljivo je vrlo loše dentalno stanje osobe pa je razumljivo zašto se takve pogreške događaju.

4.2. Procjena spola

Procjena spola je još jedan primjer klasifikacije. Dizajnirali smo vlastitu duboku konvolucijsku neuronsku mrežu za rješavanje ovog problema. Jedini ulaz koji nam je potreban

je slika veličine 512×512 . Referentne podatke o spolu kodiramo kao polje s jednim aktivnim elementom (engl. *one-hot*).



Slika 4.3 – Arhitektura neuronske mreže za procjenu spola

Na slici 4.3 prikazana je arhitektura korištene neuronske mreže. Neuronska mreža sastoji se od četiriju konvolucijskih slojeva. Svaki konvolucijski sloj praćen je slojem za sažimanje. Nakon posljednjeg sloja za sažimanje nalazi se potpuno povezani sloj s 1024 neurona te još jedan potpuno povezani izlazni sloj s dvama neuronima (po jedan za svaki spol). Aktivacijska funkcija na izlazu je softmax. Mrežu smo trenirali na 4000 slika bez dodatnog argumentiranja.

Neki od korištenih hiperparametara su:

- Konstanta učenja: 0.0001
- Optimizator: Adam
- Broj epoha: 100
- Batch size: 50

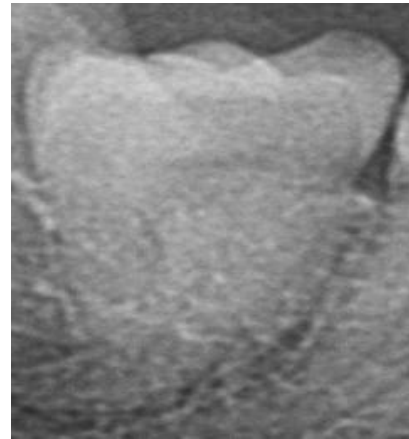
Nakon treniranja testirali smo model nad testnim skupom od 104 slike te smo imali svega tri pogrešne procjene, što rezultira točnošću od 97,11%.

Mrežu za procjenu spola zatim smo trenirali na slikama pojedinačnih zuba. Usporedbom rezultata između dvaju modela možemo provjeriti saznaje li mreža informacije o spolu iz slike cijele čeljusti ili su joj dovoljne informacije o korijenu pojedinog zuba. Slike pojedinih zuba možemo manualno generirati ili koristiti Faster R-CNN model opisan u nastavku.

S obzirom na to da je ulaz mreže fiksne veličine, potrebno je promijeniti veličinu slika pojedinih zuba na istu veličinu. To postizemo rastezanjem ili skupljanjem slike na prosječnu vrijednost širine i visine ili rezanjem svih slika na maksimalne dimenzije za pojedini zub. Prvim načinom gubimo na informaciji zbog nepoštivanja originalnog omjera visine i širine slike, dok drugim načinom dobivamo slike koje uglavnom sadrže dio susjednog zuba.



Slika 4.4 – Slika zuba 48 generirana pomoću maksimalnih koordinata



Slika 4.5 – Slika zuba 48 generirana promjenom veličine na prosječne vrijednosti

Mreža je trenirana na 1700 slika zuba klase 48, 44 te 41 koristeći oba načina rezanja slike, čime se dobije 6 treniranih modela spremnih za evaluaciju. Najbolje rezultate dobili smo za zub klase 48 koristeći rezanje slike pomoću maksimalnih dimenzija te točnost takve evaluacije iznosi 55,37%, što je relativno loš rezultat i malo je bolje od bacanja novčića. S obzirom na visoku točnost procjene spola iz cijelog ortopantomograma, može se zaključiti da sami zubi u sebi ne nose korisne informacije za procjenu spola, već da te informacije dolaze iz strukture čeljusti.

4.3. Detekcija zuba


Da bismo ocijenili naš model, moramo odabrati evaluacijsku metriku. Metrika će bilježiti razne statističke podatke našeg modela te će nam dati informaciju o tome koliko je model točan. S obzirom na to da u strojnom učenju postoje razni modeli, vrlo je važno odabrati prikladnu metriku. Također, važno je da metrika bude objektivna kako bismo mogli usporediti performanse više različitih modela koji imaju isti zadatak. U ovom slučaju radi se o modelima detekcije objekata. Konkretno, to su modeli YOLO i Faster RCNN. U ovom radu odlučili smo se za evaluacijsku metriku prosječne srednje preciznosti (engl. *mean average precision*, skraćeno mAP).

Metrika mAP je standardni postupak za ocjenu performansi sustava za detekciju objekata. Svaka metrika, pa tako i ova, mora imati dostupne podatke za usporedbu s dobivenim rezultatima. Te podatke nazivamo „referentni podaci“ (engl. *ground-truth*). Kod problema detekcije objekata referentni podaci uključuju koordinate svakog objekta na slici te pripadnu klasu korespondentnog objekta.

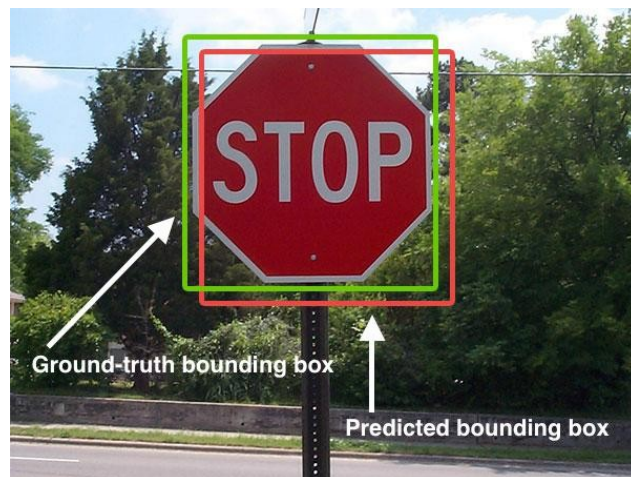
Metrika mAP računa se prema sljedećem principu:

1. Odaberemo klasu za koju želimo utvrditi točnost detekcije.
2. Inicijaliziramo cjelobrojne varijable TP (engl. *true positive*), FP (engl. *false positive*) te FN (engl. *false negative*) na nulu.
3. Idemo redno po slikama i tražimo objekte odabrane klase.
4. Kada nađemo objekt pripadajuće klase, pokrenemo sustav za detekciju objekata.
5. Pregledavamo podatke o dobivenim rezultatima i tražimo taj isti objekt.
6. Ako smo objekt pronašli među rezultatima, računamo ono što se zove „presjek kroz unija“ (engl. *intersection over union*, skraćeno IoU). IoU nam govori u kojoj se mjeri pronađeni objekt i stvarni objekt poklapaju, kao što je prikazano na slici 1. Ako je broj IoU veći od 0,5, tada rezultat smatramo i bilježimo kao točnu detekciju te povećavamo varijablu TP za jedan. Ako objekt nismo pronašli ili je broj IoU manji od 0,5, onda kažemo da se pojavila pogreška tipa 2, tj. objekt postoji na slici, ali ga sustav za detekciju nije pronašao. U tom slučaju povećavamo varijablu FN za jedan.
7. Ako postoji objekt među rezultatima kojeg nema na originalnoj slici, kažemo da se dogodila pogreška tipa 1. Povećavamo varijablu FP za onaj broj koliko postoji objekata među rezultatima kojih nema na stvarnoj slici.
8. Na temelju rezultata nakon obilaska svih slika provodi se postupak crtanja Precision-Recall krivulje, u čije detalje ne ulazimo. Važno je napomenuti da prema [41] integral krivulje Precision-Recall odgovara prosječnoj točnosti detekcije dotične klase.
9. Postupak ponavljamo za sve klase i uzimamo prosjek integrala kao mjeru mAP.

Za sâm izračun koristili smo gotovu biblioteku [42] pisanu u Pythonu, za koju je bilo potrebno pripremiti referentne oznake i dobivene rezultate.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Slika 4.6 – Računanje IoU⁷



Slika 4.7 – Usporedba rezultata i referentnih

podataka⁸

⁷https://www.pyimagesearch.com/wp-content/uploads/2016/09/iou_equation.png.

⁸https://upload.wikimedia.org/wikipedia/commons/2/2d/Intersection_over_Union_-_object_detection_bounding_boxes.jpg.

4.3.1. Rezultati detekcije pomoću mreže Faster RCNN

Za označavanje zuba Faster R-CNN mrežom koristili smo službenu implementaciju u Tensorflow Object Detection API-u [43] koji nam omogućuje treniranje raznih mreža, među kojima je i Faster R-CNN.

Za treniranje smo koristili 1000 ortopantomograma s pripadnim XML zapisima za oznake pojedinih zuba. Odlučili smo iskoristiti pretreniranu ResNet101 mrežu [44] za inicijalnu konvolucijsku mrežu čiji je cilj generiranje već opisanih mapa obilježja (engl. *feature map*).

Mrežu smo trenirali uz sljedeće parametre koji su zapisani u konfiguracijskoj datoteci:

- Broj klasa – 33
- Batch size – 20
- Konstanta učenja – 0,0001
- Broj iteracija – 80000

Ostali parametri postavljeni su na preporučene vrijednosti iz originalnog rada o Faster R-CNN mreži. Mrežu smo evaluirali na 100 slika koje nismo koristili pri treniranju.

Rezultati su prikazani u tablici 4.1.

Tablica 4.1 – Prikaz rezultata Faster RCNN modela

klasa	AP	klasa	AP	klasa	AP	klasa	AP
11	94,88%	21	92,44%	31	88,55%	41	84,10%
12	90,06%	22	93,04%	32	85,19%	42	85,00%
13	87,96%	23	86,04%	33	78,42%	43	85,81%
14	84,48%	24	82,93%	34	81,13%	44	75,994%
15	82,01%	25	83,71%	35	78,90%	45	82,48%
16	76,23%	26	69,61%	36	71,45%	46	68,04%
17	75,05%	27	71,40%	37	68,37%	47	72,58%
18	69,38%	28	70,83%	38	70,50%	48	71,52%

Mjera mAP za sve klase pri evaluaciji je 77,88%. Može se zaključiti da mreža generalno bolje prepoznaje zube u sredini ortopanske slike. Tako, na primjer, od 99 zuba klase 11 mreža korektno označi 94 zuba. S druge strane, za 84 zuba klase 48 mreža korektno označi 72 zuba. Mreža je trenirana na relativno malom broju slika te se očekuje da će rezultati biti bolji jednom kada se ulazni skup ortopantomograma proširi.

4.3.2. Rezultati detekcije pomoću mreže YOLO

Korištena implementacija YOLO modela preuzeta je s [45] te je modificirana za rad na našoj serverskoj konfiguraciji.

Naš YOLO model trenirali smo na serverskoj konfiguraciji s dvama NVIDIA Titan X grafičkim procesorima. Treniranje je trajalo 30 sati. Naš ulazni skup slika sačinjavalo je 2000 dentalnih radiografskih snimki s pripadnim anotacijama.

Neki od korištenih hiperparametara:

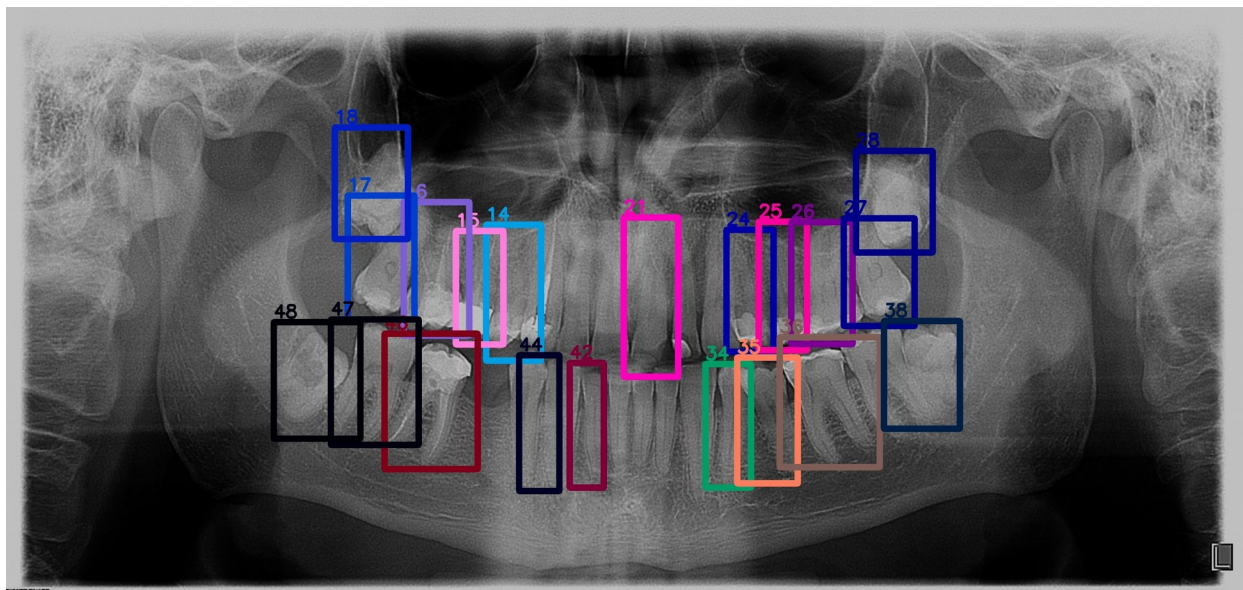
- Batch size – 16
- Konstanta učenja – 0,001
- Broj epoha – 2000

Dobiveni rezultati prikazani su u tablici 4.2.

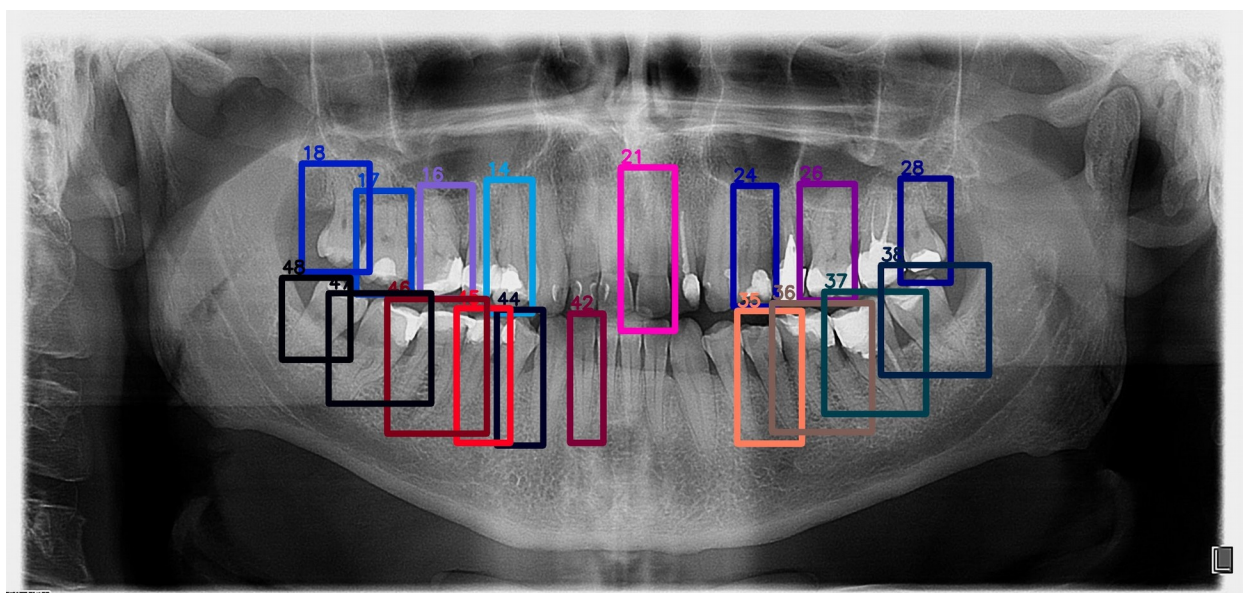
Tablica 4.2 – Prikaz rezultata YOLO modela

klasa	AP	klasa	AP	klasa	AP	klasa	AP
11	85,20%	21	24,47%	31	0,56%	41	13,82%
12	27,45%	22	33,12%	32	8,89%	42	60,48%
13	13,60%	23	18,44%	33	10,21%	43	5,99%
14	85,85%	24	75,65%	34	48,21%	44	53,48%
15	33,24%	25	53,14%	35	86,61%	45	69,68%
16	81,67%	26	77,42%	36	82,19%	46	89,14%
17	70,99%	27	69,86%	37	75,31%	47	71,08%
18	100,00%	28	96,67%	38	100,00%	48	92,89%

Prosječna srednja točnost, tj. mAP, je 55,01%. Slijedi nekoliko ilustracija rezultata.



Slika 4.8 – Primjer detekcije



Slika 4.9 – Primjer detekcije

Dobiveni rezultati ucrtani su na referentne slike korištenjem OpenCV biblioteke [46]. Svaki pravokutnik također ima oznaku klase, tj. zuba koji predstavlja. Vizualnom inspekcijom vidljivo je da YOLO model postiže veću točnost pri detekciji zuba vanjskog dijela slike, a manju točnost pri detekciji prema sredini slike. To podupiru i izmjereni podaci mAP u tablici 4.2. Uz pristup većem broju slika naš model postizao bi bolje rezultate.

5. Rasprava

Uzimanje zubnog statusa i procjena dentalne dobi je postupak koji se vrši u mnogim granama stomatologije (ortodonciji, dječjoj stomatologiji, dentalnoj antropologiji). Može se vršiti na živim osobama, ali i na mrtvima, u sklopu stomatološkog područja koje nazivamo forenzička stomatologija. S obzirom na to da zubi ostaju dugo sačuvani nakon smrti, mnogo puta je zubalo jedino dostupno antropološko obilježje za identifikaciju žrtava avionskih nesreća, prirodnih katastrofa, ali i drugih situacija u kojima nije poznat identitet preminule osobe. Tako su nakon rata forenzični stomatolozi sudjelovali i pri otkrivanju žrtava iz masovnih grobnica na temelju njihovih postmortalnih ostataka. S druge strane, pojavom migrantskog vala određivanje starosti na živim osobama sve je potrebnije zbog ljudi koji dolaze bez ikakvih dokumenata i nemoguće ih je registrirati pri dolasku ili pokrenuti proces dobivanja azila. Nadalje, određivanje starosti na živim osobama potrebno je i da bi se odredila dob napuštene djece i ljudi koji prikrivaju svoju dob zbog raznih sudskih procesa.

Metode određivanja dentalne dobi dijelimo na neinvazivne i invazivne. Invazivne metode podrazumijevaju vađenje zuba te mikroskopsko određivanje promjena na vanjskim i unutarnjim dijelovima zuba. Najpoznatija metoda je ona po Gustafsonu koja koristi 6 parametara za određivanje dentalne dobi. To su izraženost atricije zuba, resorpcija alveolne kosti, nakupljenost sekundarnog dentina, nakupljenost acelularnog cementa, translucencija korijena zuba i resorpcija korijena [47]. Iz etičkih razloga, invazivne metode nisu prikladne za uporabu kod živih osoba, već samo u forenzičkoj stomatologiji. S druge strane, neinvazivne metode temelje se na radiološkim snimkama, najčešće ortopantomogramima, iz kojih se dobivaju informacije o zamecima zuba, stadijima mineralizacije i erupcije krune u oralnu šupljinu. Jedna od najčešće korištenih je metoda po Demirijanu gdje se na ortopantomogramu koristi 7 zuba lijeve strane mandibule, od središnjeg sjekutića do drugog kutnjaka. Svaki se zub obilježi jednim od 8 opisanih stadija mineralizacije, od A do H. Oznake se zatim pretvore u bodove prema vrijednosnoj tablici i zajedno daju zrelosne bodove na temelju kojih se zatim može odrediti dob pojedinca [48].

Cilj ovog istraživanja bio je razviti metodu temeljenu na dubokom učenju koja bi iz ortopantomograma mogla odrediti zubni status te procijeniti dob i spol osobe. Razvijene su četiri metode temeljene na dubokim neuronskim mrežama koje su ispitivanjem dale vrlo dobre rezultate. Temeljem dobivenih rezultata dokazana je moguća primjena razvijenih metoda u procjeni spola, dobi i određivanju zubnog statusa te samim time one dobivaju na važnosti i potencijalnoj primjenjivosti u budućnosti. Jedna od prednosti navedenih metoda jest da pripadaju u neinvazivne tehnike, što povećava opseg njihovih mogućih primjena. S obzirom na to da je identifikacija preminulih osoba često dug i težak posao (posebice kod masovnih stradavanja), ove metode bi računalnim uzimanjem zubnog statusa, procjene dobi i spola znatno olakšale i ubrzale taj proces čineći ga znatno jeftinijim. Osim toga, smanjila bi se mogućnost ljudskih pogrešaka, posebno onih koje bi bile izazvane visokom razinom emocionalnog stresa koji se javlja kod forenzičnih stomatologa koji su mu intenzivno izloženi

tijekom identifikacije žrtava masovnih stradavanja u kratkom vremenskom razdoblju. Isto tako, kod živih osoba određivanje zubnog statusa i dobi bilo bi jednostavnije te bi se time svakodnevni rad stomatologa unaprijedio. Nadalje, prednost ovih metoda je i njihova jednostavna uporaba, bez potrebnog predznanja, posebne obuke ili opreme. Mogući nedostatak ranije navedenih metoda je trenutačna nemogućnost prepoznavanja krunica, implantata, mostova, ispuna itd. prilikom određivanja zubnog statusa. To je moguće riješiti daljnjim usavršavanjem modela uz korištenje veće baze ortopantomograma.

5.1. Primjeri praktične primjene

Primjena istraživanja može ići u dvama smjerovima: automatizirano uzimanje zubnog statusa na temelju računalne analize ortopantomograma u svrhu dentalne identifikacije u individualnim i masovnim stradavanjima (prometne nesreće – npr. zrakoplovne, pomorske; prirodne katastrofe – npr. tsunami, potresi; teroristički napadi, ratovi i dr.). Kod masovnih stradavanja automatizirano uzimanje zubnog statusa bilo bi posebno važno jer može značajno ubrzati otkrivanje identiteta stradale osobe, posebno ako postoje primjereni prijesmrtni podaci (zubni karton, ortopantomogram i dr.). S druge strane, primjena koja se nadovezuje na ranije opisanu je određivanje dentalne dobi, što također uključuje prethodno opisanu praktičnu upotrebu.

5.1.1. „Ljudi bez identiteta“

Prema procjeni UN-a, danas u Republici Hrvatskoj živi 3000 ljudi bez državljanstva i osobnih dokumenata, koji u formalno-pravnom smislu ne postoje. Oni nisu uvedeni u bilo kakve popise ili evidencije te o njima nitko ništa ne zna. Na svijetu je čak 12 milijuna tzv. „ljudi bez identiteta“. Ovom brzom i jednostavnom metodom pružila bi se mogućnost uvida u njihovu dob, što bi već bilo od velikog značaja. Najvažniji podatak koji se može približno odrediti ovom metodom jest punoljetnost, odnosno maloljetnost neidentificirane osobe. Kronologija razvoja zuba manje je promjenjiva nego u razvoju kostiju pa je zato preciznija za identifikaciju dobi.

5.1.2. Utvrđivanje identiteta žrtava nesreća

Utvrđivanje identiteta pomoću dentalnih parametara ima vrijednost otisaka prstiju živih ljudi. To vrlo važno svojstvo denticije pronašlo je primjenu u nesrećama u kojima je čitavo ljudsko tijelo uništeno dugotrajnom izlaganju vodi, karbonizirano utjecajem vatre i slično. Zubi su otporni na visoke temperature, mnogo više od kostiju, jer caklina počinje dekrystalizirati tek pri izloženosti temperaturi od 1300°C. Tako je nerijetko zubno tkivo jedini izvor informacija o unesrećenima [49].

Denticija je apsolutno individualna osobitost pojedinca te ne postoji identičan nalaz dviju osoba. Nakon utvrđivanja dobi unesrećenih, postupak povezivanja identiteta i postmortalnih ostataka znatno je jednostavniji.

Nadalje, ako je stanje zubnih tkiva rendgenski zabilježeno za života, kompjutorski je mnogo lakše i brže pronaći podudarnosti s postmortalnom snimkom denticije. Tako utvrđivanje identiteta može biti olakšano uz pomoć umjetne inteligencije [50].

5.1.3. Procjena dobi napuštene djece

U nerazvijenim zemljama, gdje vladaju loše ekonomske i socijalne prilike, sve je više napuštene djece koja nemaju dokumente i čija starost nije poznata. Primjena navedene metode omogućila bi brži i jednostavniji uvid u njihovu dob.

Iako je druga po redu najveća ekonomska sila na svijetu, broj siročadi u Republici Kini kreće se od 600 000 pa do milijun. Unatoč svojem radu, Kina nije osmislila prihvatljivu mrežu socijalne podrške za one kojima je ona najpotrebnija. Kako kineski zakon o posvajanju nalaže da nije dozvoljeno posvojiti siročad nakon što napuni 14 godina, primjena ove metode dobiva na važnosti.

6. Zaključak

Nakon provođenja niza eksperimenata ustanovili smo da je korištenjem trenutno dostupnih tehnologija metode dubokog učenja itekako moguće primijeniti na rješavanje problema određivanja zubnog statusa, procjene godina i spola osobe.

Pokazalo se da je problem procjene spola, od triju ranije navedenih, računalu najlakši. Korištenjem relativno jednostavne konvolucijske neuronske mreže postižu se iznimno dobri rezultati s točnošću od 97,11%.

Sljedeći u nizu je problem određivanja godina s prosječnom apsolutnom pogreškom od 4,9 godina. Iako je ovo vrlo dobar rezultat, smatramo da postoji mogućnost za napredak. Jedan od problema konvolucijskih neuronskih mreža na koji smo i sami naišli prilikom procjene godina je potreba za većom količinom podataka za treniranje. Iz tog razloga mrežu za procjenu godina morali smo ograničiti na interval od 20 do 50 godina, jer izvan tih granica nismo imali dovoljan broj ortopantomograma na kojima bi neuronska mreža mogla „učiti“.

Naposlijetku, problem određivanja zubnog statusa korištenjem YOLO i Faster RCNN modela također se pokazao vrlo uspješnim. Tako YOLO ostvaruje mAP rezultate od 55,01%, dok superiorniji Faster RCNN ostvaruje 77,88%. Potencijalni problem prilikom određivanja zubnog statusa je, kao i ranije, nedovoljan broj ortopantomograma na kojima bi mreža mogla učiti o kronicama, implantatima, mostovima itd., s obzirom na to da ti slučajevi nisu česti.

Spomenute probleme moguće je riješiti većom bazom ortopantomograma.

LITERATURA

- [1] Brkić H., Kaić Z., Keros J., Šoljan M., Turković K. et al. *Forenzična stomatologija*. Zagreb, Školska knjiga. 2000. Str. 44-51.
- [2] Brkić, H. *Procjena dentalne dobi u forenzičnim postupcima*, Sonda, 2015., www.sonda.sfzg.hr.
- [3] Brkić H., Škavić J., Strinović D. *Postmortalna identifikacija tijela postignuta statusom zubala*, Acta Stomatol. Croat. 1994; 28: 231-236.
- [4] Brkić H., Kaić Z., Keros J., Šoljan M., Turković K. et al. *Forenzična stomatologija*. Zagreb, Školska knjiga. 2000. Str. 44-51.
- [5] Rajić Šikanjić P., Janković I., M.O. *Metode određivanja spola na ljudskom skeletnom materijalu*. 2003.
- [6] Brkić H., Kaić Z., Keros J., Šoljan M., Turković K. et al. *Forenzična stomatologija*. Zagreb, Školska knjiga. 2000. Str. 44-51.
- [7] Labellmg. <https://github.com/tzutalin/labellmg>.
- [8] Machine learning. https://en.wikipedia.org/wiki/Machine_learning.
- [9] Artificial neural network. https://en.wikipedia.org/wiki/Artificial_neural_network.
- [10] Gibson A., Patterson J., Deep Learning <https://www.safaribooksonline.com/library/view/deep-learning/9781491924570/ch04.html>.
- [11] CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more.. https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5.
- [12] Linear Regression Analysis in SPSS Statistics - Procedure, Assumptions and Reporting the Output. <https://statistics.laerd.com/spss-tutorials/linear-regression-using-spss-statistics.php>.
- [13] Multivariate regression. <https://stackoverflow.com/questions/30336138/how-to-plot-a-multivariate-function-in-python>.
- [14] Šnajder J. (2015). Bilježnica 5: Regresija. <http://nbviewer.jupyter.org/github/jsnajder/StrojnoUcenje/blob/master/notebooks/SU-2015-5-Regresija.ipynb>
- [15] Brownlee J. (2017). A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size. <https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/>.
- [16] Thomas A. (2017). Stochastic Gradient Descent - Mini-Batch and More - Adventures in Machine Learning. <http://adventuresinmachinelearning.com/stochastic-gradient-descent>.
- [17] Brownlee J. (2016). Logistic Regression for Machine Learning. <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>.
- [18] Šnajder J. (2015). Bilježnica 7: Logistička regresija. <http://nbviewer.jupyter.org/github/jsnajder/StrojnoUcenje/blob/master/notebooks/SU-2015-7-LogistickaRegresija.ipynb>.
- [19] Kang N. (2017.) Multi-Layer Neural Networks with Sigmoid Function – Deep Learning for Rookies (2). <https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f>.
- [20] Multinomial Logistic Regression: Definition and Examples.

- <http://www.statisticshowto.com/multinomial-logistic-regression/>.
- [21] Nicholson C., Gibson A., Patterson J. (2017.) Using Neural Networks With Regression – Deeplearning4j: Open-Source, Distributed Deep Learning for the JVM. <https://deeplearning4j.org/logistic-regression.html>.
- [22] Sharma S. (2017). Activation Functions: Neural Networks – Towards Data Science. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [23] Malvić, M., Vrbanac, B. (2009.) Geomatematički Pojmovnik. <http://e.math.hr/vol23/Malvic>.
- [24] Dahal, P. (2017.) Classification and Loss Evaluation – Softmax and Cross Entropy Loss. <http://deepnotes.io/softmax-crossentropy>.
- [25] Ren S., He K., Girshick R., Sun J. (2016.) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. <https://arxiv.org/pdf/1506.01497.pdf>.
- [26] Uijlings J.R.R., van de Sande K.E.A., Gevers T. Smeulders A.W.M. (2013.) Selective Search for Object Recognition. <https://koen.me/research/pub/uijlings-ijcv2013-draft.pdf>.
- [27] Girshick R. (2015.) Fast R-CNN. <https://arxiv.org/pdf/1504.08083.pdf>.
- [28] Das S. (2017.) CNNs Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more. https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5.
- [29] Gao H. (2017.) Faster R-CNN Explained. <https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8>.
- [30] Rosebrock A. (2016.) Intersection over Union (IoU) for object detection. <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>.
- [31] Hosang J., Benenson R. (2017.) Learning non-maximum suppression. <https://arxiv.org/pdf/1705.02950.pdf>.
- [32] Differences between L1 and L2 as Loss Function and Regularization. <http://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/>.
- [33] Grel T. (2017.) Region of interest pooling explained. <https://blog.deepsense.ai/region-of-interest-pooling-explained/>.
- [34] Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. https://pjreddie.com/media/files/papers/yolo_1.pdf.
- [35] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2014.) Going deeper with convolutions. <https://arxiv.org/abs/1409.4842>.
- [36] Tensorflow. <https://www.tensorflow.org/>.
- [37] CUDA Zone. <https://developer.nvidia.com/cuda-zone>.
- [38] Keras: The Python Deep Learning library. <https://keras.io>.
- [39] Docker. <https://www.docker.com/>.
- [40] Simonyan K., Zisserman A. (2014.) Very deep convolutional networks for large-scale image recognition. <https://arxiv.org/pdf/1409.1556.pdf>.
- [41] Everingham M., Van Gool L., Williams, C. K. I., Winn J., Zisserman A. (2010.) The pascal visual object classes (voc) challenge. http://homepages.inf.ed.ac.uk/ckiwi/postscript/ijcv_voc09.pdf.
- [42] Cartucho. <https://github.com/Cartucho/mAP>.
- [43] Tensorflow Object Detection API.

- https://github.com/tensorflow/models/tree/master/research/object_detection.
- [44] He K., Zhang X., Ren S., Sun J. (2015.) Deep Residual Learning for Image Recognition. <https://arxiv.org/pdf/1512.03385.pdf>.
- [45] Darkflow. <https://github.com/thtrieu/darkflow>.
- [46] OpenCV library. <https://opencv.org/>.
- [47] Brkić H., Kaić Z., Keros J., Šoljan M., Turković K.. et al. *Forenzična stomatologija*. Zagreb, Školska knjiga. 2000. Str. 44-51.
- [48] Demirjian A., Goldstein H., Tanner JM. *A new system of dental age assessment*. Hum Biol. 1973 May; 45(2):211-27.
- [49] Galić J., Brkić H., Kaić Z., Čadež J. *Postmortalna identifikacija dobi postignuta zubarom*. Acta stomatol Croat. 1995;29(3):197-201.
- [50] Dumančić J., Kaić Z., Njemirovskij V., Brkić H., Zečević D. *Dental identification after two mass disasters in Croatia*. Croat Med J. 2001 Dec;42(6):657-62.

Sažetak

Procjena zubnog statusa pomoću dubokog učenja

U ovom radu bavimo se primjenom metoda dubokog učenja na zadacima procjene zubnog statusa, procjene dobi i spola. Primjena je temeljena na analizi rendgenskih snimaka na kojima je moguće vidjeti različite faze razvoja zuba, ali i histološke promjene na zubima već završenog razvoja. Korištenjem umjetnih konvolucijskih neuronskih mreža navedene probleme rješavamo na vrlo učinkovit i formalan matematički način. Objasnjeno je koncept neuronskih mreža te je pružen detaljan opis svih korištenih modela. Godine određujemo metodom regresije, spol metodom klasifikacije, a zubni status korištenjem sustava za lokalizaciju i klasifikaciju objekata YOLO i Faster RCNN. Prosječna pogreška pri procjeni godina iznosi 4,9 godina, točnost procjene spola je 97,11%, a modeli YOLO i Faster RCNN pri određivanju zubnog statusa ostvaruju mAP od 55,01% i 77,88%, respektivno.

Ključne riječi: zubni status, godine, spol, duboko učenje, neuronske mreže

Summary

Dental status assessment using deep learning

In this paper we present the application of deep learning methods in dental status assessment, age and gender prediction. Application of deep learning methods is based on the analysis of X-ray images for which it is possible to see different stages of tooth development, as well as histological changes in the teeth that had already completed their development. By using artificial convolutional neural networks we solve these problems in a very effective and mathematical formal manner. We explain the concept of neural networks and we provide a detailed look into all used models. We use regression to predict age, classification to predict gender, and we assess dental status by utilizing localisation and classification systems such as YOLO and Faster RCNN. We achieve average age prediction error of 4,9 years with gender prediction accuracy of 97,11%. Using YOLO and Faster RCNN we obtain mAP score of 55,01% and 77,88% for dental status assessment, respectively.

Keywords: dental status, age, gender, deep learning, neural networks