

Primjena YOLO algoritma za detekciju tumora na slikama magnetske rezonance mozga

Vujević, Sven

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:921144>

Rights / Prava: [Attribution-NonCommercial-NoDerivs 3.0 Unported / Imenovanje-Nekomercijalno-Bez prerađivanja 3.0](#)

Download date / Datum preuzimanja: **2024-07-23**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Sven Vujević

PRIMJENA YOLO ALGORITMA ZA
DETEKCIJU TUMORA NA SLIKAMA
MAGNETSKE REZONANCE MOZGA

ZAVRŠNI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N

Sven Vujević

Matični broj: 1191249650

Studij: Informacijski i poslovni sustavi

**PRIMJENA YOLO ALGORITMA ZA DETEKCIJU TUMORA NA
SLIKAMA MAGNETSKE REZONANCE MOZGA**

ZAVRŠNI RAD

Mentor/Mentorica:

Doc. dr. sc. Petra Žugec

Varaždin, srpanj 2024.

Sven Vujević

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Tema ovog rada je primjena YOLO algoritma za detekciju tumora na slikama magnetske rezonance mozga. Rad priprema čitatelja za sadržaj koji slijedi na način da se na početku objasne osnovni pojmovi i koncepti vezani uz tumor, njegovo metastaziranje te vrste pregleda mozga uključujući magnetsku rezonancu koju ovaj rad koristi kao podlogu u svom praktičnom dijelu. Zatim su opisani začetci i razvoj umjetne inteligencije, te njena primjena u medicini. Nakon toga se nešto detaljnije razrađuju neuronske mreže, koncept jednostavnog perceptrona, prolaze se neke od uobičajenijih aktivacijskih funkcija i prikazana je i objašnjena struktura neuronske mreže. Slijedi poglavlje o konvolucijskim neuronskim mrežama te dva glavna ponavljajuća sloja u njima: konvolucijski sloj i sloj sabiranja. Nakon toga slijedi poglavlje o teorijskom načinu rada YOLO algoritma kao i o matematičkoj pozadini tog algoritma. Ova poglavlja integriraju brojne slike i formule radi lakšeg razumijevanja algoritma. Konačno, nakon adekvatnog teorijskog i matematičkog uvoda temi potrebnog za bolje razumijevanje praktičnog dijela rada slijedi i sam praktičan dio, koji je izveden pomoću programskog jezika Python i njegovog paketa ultralytics. Za anotiranje skupa podataka, koji se sastoji od nešto više od 200 slika magnetske rezonance, korišten je CVAT alat budući da su neke od slika iz skupa podataka tumor-negativne, a neke tumor-pozitivne (ove druge potrebno je adekvatno anotirati kako bi YOLO algoritam mogao raditi shodno očekivanjima). Naposljetku iznesen je zaključak rada i rezultata provođenja praktičnog dijela.

Ključne riječi: YOLO; strojno učenje; računalni vid; vjerojatnost; tumor; Python; umjetna inteligencija; konvolucijske neuronske mreže

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	2
3. Tumor	3
3.1. Tumor mozga	5
3.1.1. Magnetska rezonanca mozga	6
4. Umjetna inteligencija	9
4.1. Začetci umjetne inteligencije	9
4.2. Primjena umjetne inteligencije u medicini	10
5. Neuronske mreže	12
5.1. Umjetne neuronske mreže	12
5.1.1. Perceptron	13
5.1.2. Aktivacijske funkcije	14
5.1.3. Struktura neuronske mreže	15
5.2. Konvolucijske neuronske mreže	16
5.2.1. Konvolucijski sloj	17
5.2.2. Sloj sabiranja	19
6. YOLO algoritam	21
6.1. YOLOv1 algoritam	21
6.1.1. Matematička pozadina YOLOv1 algoritma	22
6.1.2. Arhitektura YOLO algoritma	28
6.1.3. Funkcija gubitka	29
7. Detekcija tumora mozga primjenom YOLOv8 algoritma	31
7.1. Razlike YOLOv1 i YOLOv8 algoritma	31
7.2. Priprema programskog okruženja	32
7.2.1. CVAT	32
7.2.2. Python	32
7.2.3. Prikaz i analiza grafova YOLOv8 modela	33
7.2.4. Prikaz i analiza rezultata treniranja YOLOv8 modela	37
8. Zaključak	44
Popis literature	45

Popis slika	50
1. Prilog 1	52
2. Prilog 2	54

1. Uvod

Tema ovog završnog rada jest, kao što se da naslutiti iz naslova istog, primjena YOLO algoritma za detekciju tumora na slikama magnetske rezonance mozga. Ova tema značajna je iz mnogo razloga. Kod tumora općenito, izuzetno je bitno što ranije otkrivanje kako bi liječenje moglo biti uspješno, a tim je važnija što ranija i točnija dijagnoza kod tumora mozga koji, nažalost, u još većem postotku prerano završava život pacijenta. Spomenuta brzina i točnost također bi pomogli smanjiti redove čekanja za termin snimanja magnetskom rezonancom na koju se u Hrvatskoj, nerijetko, čeka više od godinu dana ili čak dvije. U slučaju da naručena osoba ima tumor mozga, taj bi vremenski period nedvojbeno bio suviše dug i u konačnici uzaludan. Druga bitna stavka je točnost. Osim brzine, ukoliko pacijent već obavi pregled izuzetno je bitna točna dijagnoza. Ovo je element u kojemu se računala mogu vrlo dobro trenirati te pojedina istraživanja pokazuju rezultate gdje imaju točnije procjene nego medicinski stručnjaci iz područja s višegodišnjim iskustvom. Upravo ranije navedeno daje ovoj temi izuzetan značaj i osobnu motivaciju za proučavanje, bavljenje i pisanje ovog rada.

Rad se sastoji od teorijskog i praktičnog dijela. Način na koji je ovo provedeno u radu je sljedeći: prvi dio rada pokriva teorijsku podlogu potrebnu za razumijevanje važnosti i osnovnih pojmova vezanih uz predmet rada. Teorijska podloga kreće od tema tumora i umjetne inteligencije te sve uže i praktičnije prema neuronskim mrežama na čijem se radu zasniva YOLO algoritam, pa do samog YOLO algoritma objašnjenog kako teorijski tako i suštinski primjenom brojnih matematičkih formula i potpornih slika. Praktičan dio sastoji se od provedbe YOLO algoritma u Pythonu, dok je za skup podataka korišten besplatan i dostupan skup sastavljen od tumor-pozitivnih i tumor-negativnih slika. Zatim je skup adekvatno anotiran te lokalno pohranjen u formatu koji je potreban za nesmetan rad YOLO algoritma. Nakon toga rad je pokrenut u 100, 200 i 500 epoha te su rezultati i zaključci prikazani i analizirani u radu.

2. Metode i tehnike rada

U razradi teme ovog završnog rada objedinjeno je znanje i koncepti stečeni u cjelokupnom dosadašnjem visokom obrazovanju koje sam pohađao. Za prikupljanje i analizu materijala, informacija te podataka korišteni su razni web izvori, online enciklopedije kao i akademski članci, završni i diplomski radovi, knjige, prezentacije i video sadržaji. Jedan od primarnih izvora bio je upravo članak samih autora YOLO algoritma: You only look once: Unified, real-time object detection, autora: Joseph Redmon, Santosh Divvala, Ross Girshick i Ali Farhadi. Izrade pojedinih slika i grafova rađene su samostalno, dok su pri izradi rada korišteni i razni softverski alati: Overleaf u kojemu je pisan LaTeX kod, dakle i sam rad, CVAT alat za anotaciju skupa podataka, programski jezik Python za praktičnu primjenu YOLO algoritma, Geogebra pri izradi određenih grafova, paint.net za grafičko uređivanje pojedinih slika, Kaggle za preuzimanje skupa podataka i brže izvršavanje programskog koda te YouTube za pojedine video materijale i Visual Studio Code za virtualno okruženje programskom jeziku Python u prve dvije verzije koda. Nadalje, radi bolje i lakše sistematizacije rada, teme su podijeljene na manje jedinice koje se međusobno nadopunjuju. Konačno, rad je upotpunjen brojnim slikama i formulama radi lakšeg i potpunijeg razumijevanja materije.

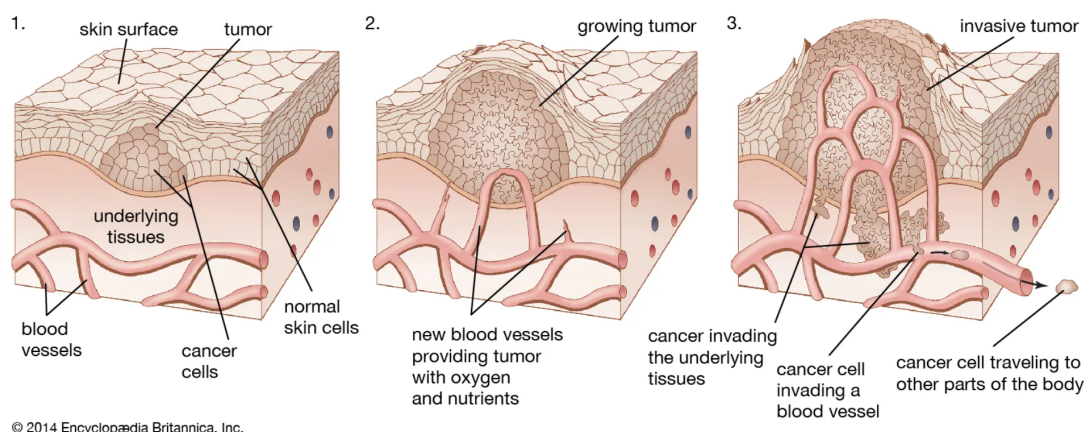
U izradu završnog rada krenuto je s mišlju kako bi YOLO algoritam mogao biti konkretna dodatna metoda korištena u medicini radi rasterećivanja medicinskog osoblja uštedom vremena, brže procjene pacijentovog stanja te pružanja drugog mišljenja stručnjacima u području u procesu donošenja odluka. Iako ovaj rad nije u mogućnosti u potpunosti ući u dubinu područja, kako radi hardverskih ograničenja tako i opsega posla koji nadilazi parametre završnog rada od kojih su neki i etičke implikacije primjene YOLO algoritma u medicini, ovaj rad dovoljan je pokazatelj potencijala i prostora moderniziranja medicine integriranjem najsvremenijih informatičkih postignuća u dotično područje. Prethodno spomenuto moguće je vidjeti na osnovi rezultata provođenja praktičnog dijela rada koji je na relativno malom skupu podataka, uz korištenje YOLOv8 nano verzije, dao izuzetno precizne predikcije obzirom na okolnosti i treniranjem na 357 epoha učenja.

3. Tumor

Riječ tumor dolazi od lat. *tumor*: oteklina [1]. Generalno govoreći, tumor je naziv za svaku oteklinu ili izraslinu, dok je značenje te riječi u užem smislu neoplazma ili novotvorina, odnosno nenormalna nakupina tkiva sa neusklađenim rastom u usporedbi sa normalnim tkivom te ga nadmašuje, a njegovo trajanje je postojano i nakon prestanka djelovanja uzroka koji ga je potaknuo [1]. Uz to, tumorske se stanice uvelike razlikuju od upala ili drugih oteklina iz razloga što su abnormalnog izgleda i drugih karakteristika, od kojih su najčešće: (1) *hipertrofija* ili povećanje veličine pojedinačnih stanica (ova značajka se može pojaviti u tumora, ali je prisutna i u mnogih drugih stanja u organizmu); (2) *hiperplazija* ili povećanje broja pojedinih stanica (ovo ponekad može biti jedini kriterij dijagnosticiranja tumora); (3) *anaplazija* ili regresija fizičkih karakteristika stanica prema primitivnijim oblicima (ovo je gotovo uvijek karakteristika zloćudnih tumora) [2]. Normalne se stanice pretvaraju u tumorske složenim procesom neoplastične preobrazbe koji još uvijek nije u potpunosti razjašnjen [1]. Također, tumori mogu nastati u bilo kojoj od trilijuna stanica u tijelu [3].

Obilježja tumora su:

1. *Nesvrhovitost* - nije koristan za organizam, a u nekim slučajevima može biti i štetan (rak) [1].
2. *Samostalnost* - nije ovisan o fiziološkim nadzornim sustavima u organizmu [1].
3. *Parazitizam* - iskorištava i iscrpljuje organizam u kojemu se razvija i raste [1].
4. *Neorganiziranost* - tumorske stanice ne stvaraju nova funkcionalna (normalna) tkiva ili organe [1].

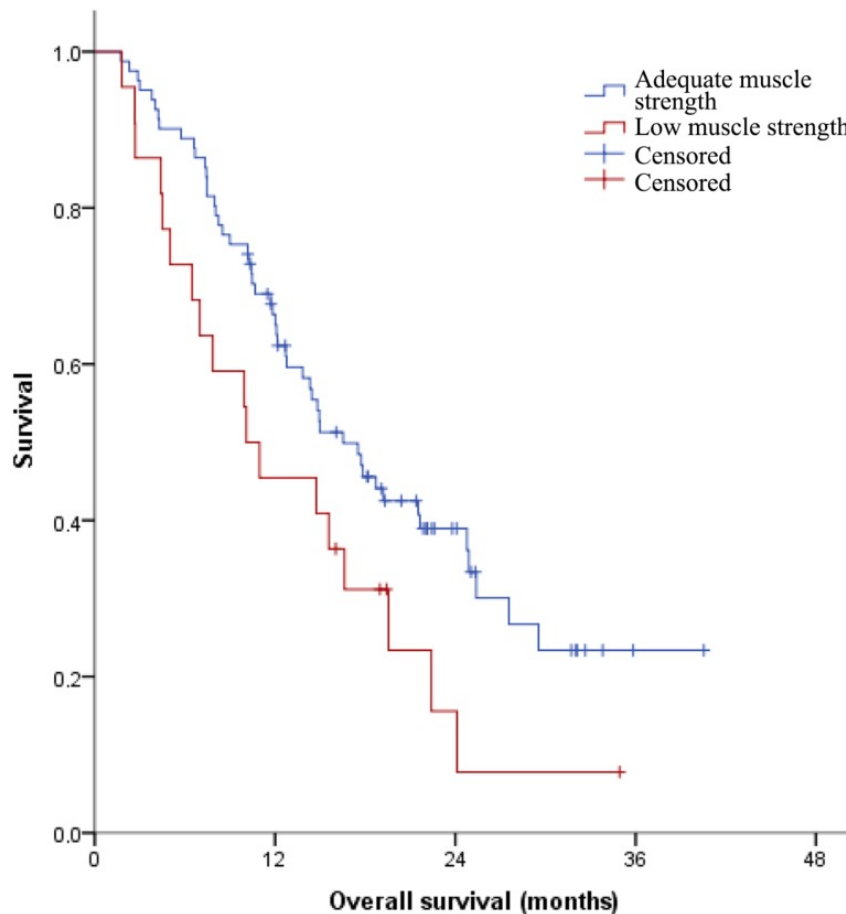


Slika 1: Razvoj tumora (Izvor: [2])

Nekoliko je ključnih faktora u uspješnom liječenju tumora, a napose tumora mozga koji je još smrtonosniji. Važno je znati da postoje IV kategorije klasifikacija tumora prema svjetskoj zdravstvenoj organizaciji (WHO). U I. kategoriju se ubrajaju benigni tumori, postoji mogućnost liječenja kroz samo operaciju bez potrebe za terapijama, sporo rastu i nisu invazivni; II. kategoriju karakteriziraju i dalje relativno sporo rastući tumori, donekle invazivni, mogu se pojaviti

kao tumori više kategorije; III. kategorija je okarakterizirana kao maligni tumor, invazivan, teži pojavljivanju kao tumor više razine; IV. kategorija su najviše maligni, agresivnog rasta, izuzetno invazivni, jako brzo se šire, neprestano se ponovo pojavljuju te su gotovo neizlječivi [4][5]. I. i II. kategorija se ubraja u nisku razinu dok se III. i IV. ubrajaju u visoku razinu tumora [4]. Iz netom spomenutog vidljivo je kako tumorske stanice mogu napredovati iz tumora niže kategorije u tumor više kategorije pa tako i razine. Dakle, vrijeme otkrivanja je izuzetno bitno pri uspješnom liječenju bilo kojeg tumora, a pogotovo tumora mozga koji je po mnogo toga izazovnije i teži za liječenje. Preventivni pregledi su od velikog značaja kada je ovaj faktor u pitanju. Osim vremena otkrivanja, lokacija, oblik i veličina tumora su također bitni faktori u određivanju težine cijelog procesa liječenja (varijabilna razina lakoće izvođenja medicinskog zahvata) [5]. Naravno, možda i najznačajniji faktor uspješnosti liječenja tumora jest vrsta tumora koji je prisutan u organizmu.

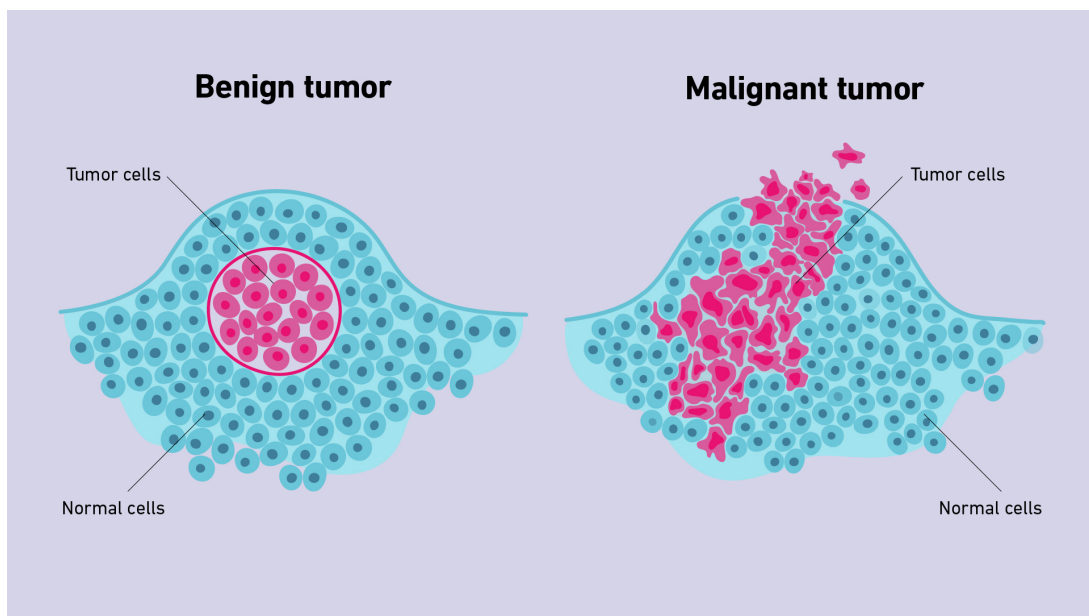
Valja napomenuti i kako osobe mlađe od 40 godina imaju veće šanse u procesu liječenja, kao i osobe sa većim udjelom mišićnog tkiva u tijelu (vidi sliku 2.) što je često međusobno povezano [5][6].



Slika 2: Smrtnost od napredovalog malignog tumora u ovisnosti o mišićnoj masi (Izvor: [6])

3.1. Tumor mozga

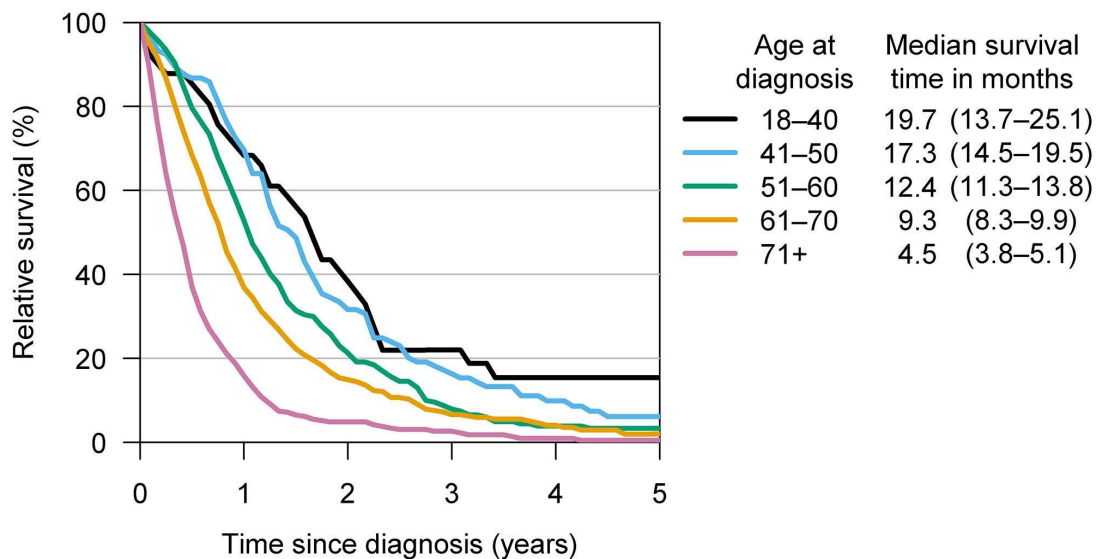
Tumor mozga je rast abnormalnih stanica u mozgu. Anatomija mozga je vrlo složena s različitim dijelovima odgovornim za različite funkcije živčanog sustava [7]. Tumori mozga mogu se pojaviti u bilo kojem dijelu mozga ili lubanje, uključujući njegovu zaštitnu ovojnica, donju stranu mozga (bazu lubanje), moždano deblo, sinuse i nosnu šupljinu te mnoga druga područja [7]. Postoji više od 150 vrsta tumora mozga ovisno o tkivu na kojemu se abnormalne stanice pojave [4]. Razlikuju se dvije grupe tumora mozga. Primarni, odnosno tumori koji se razvijaju u mozgu i sekundarni ili metastatski, tumori koji se razvijaju u nekom drugom dijelu tijela te se kasnije prošire do lubanjske šupljine ili mozga putem krvi [7][4][8][9]. Sekundarni tumori čine oko 20% svih moždanih tumora [9]. Nadalje, postoje benigni i maligni tumori. Benigni se tumori u pravilu sporo razvijaju i sporo rastu. Imaju vidljive granice i rijetko se šire. Maligni se tumori, poznatiji i pod nazivom rak, rapidno šire i napadaju susjedne zdrave stanice mozga (ili nekog drugog dijela tijela, ovisno o tome gdje se maligni tumor pojavio) [7][10].



Slika 3: Razlika između građe benignog i malignog tumora (Izvor: [11])

Slučajevi tumora mozga pojavljuju se kod 5-15 osoba na 100 000 ljudi [9]. Neki od faktora rizika su: (1) *godine*, iako se tumor mozga može pojaviti u bilo kojem stadiju života, ti su tumori u globalu učestaliji u nešto starije populacije; (2) *rasa*, iako pripadnici svake rase mogu dobiti bilo koju vrstu tumora, ipak su neke vrste učestalije u nekih rasi (npr. tako je u bijelaca učestaliji gliom, dok je u Afroamerikanaca učestaliji meningleom); (3) *izloženost zračenju*, osobe izložene visokoj razini radijacije imaju veću šansu razviti tumor mozga [8]. Ta se jaka vrsta radijacije naziva ionizirajuća radijacija te je dovoljno jaka da inicira promjene u deoksiribonukleinskoj kiselini (DNK) unutar stanica u tijelu. Navedene promjene u strukturi DNK mogu dovesti do tumora mozga. Također, ne postoji povezanost između niske razina radijacije poput radijacije pametnih telefona ili radiovalova i povećane šanse od tumora mozga [8]. Posljednji, ali ne i manje značajan, faktor rizika jest (4) *genetika*, odnosno pojedine promjene na DNK koje mogu povećati rizik od tumora mozga, a nasljedne su. Takve promjene uzro-

kuju: neurofibromatoza 1 i 2, tuberozna skleroza, Lynchov sindrom, Li-Fraumenijev sindrom, Von Hippel-Lindauova bolest, obiteljska adenomatozna polipoza, Cowdenov sindrom i Gorlinov sindrom [8].



Slika 4: Smrtnost kod glioblastoma po dobnim kategorijama (Izvor: [12])

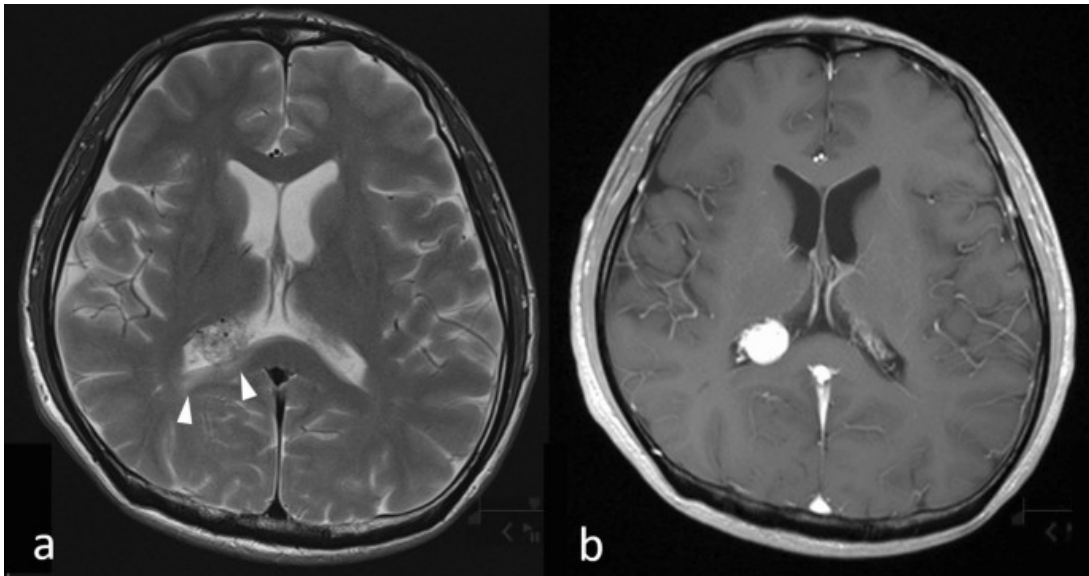
Na slici 4 (gore) prikazana je smrtnost, u mjesecima, nakon dijagnoze glioblastoma prema dobnim kategorijama. Dob se, u ovom slučaju, pokazala kao najbolji prediktor duljine života od dijagnoze. Vidljivo je kako graf razlikuje 5 linija različitih boja gdje svaka odgovara drugoj dobnj kategoriji. Tako crna linija predstavlja osobe dobi od 18 do 40, plava od 41 do 50, zelena od 51 do 60, narančasta od 61 do 70 te roza linija predstavlja osobe starije od 71 godinu. Iz legende pored grafa mogu se iščitati detaljnije informacije o dužini života u mjesecima nakon dijagnoze, no za usporedbu osobama iz dobne skupine od 18 do 40 godina je medijalno vrijeme iznosilo 19.7 mjeseci dok je isto osobama starijim od 71 godinu iznosilo tek 4.5 mjeseci.

3.1.1. Magnetska rezonanca mozga

Kao što je ranije spomenuto, rano otkrivanje tumora mozga je izrazito bitno u uspješnom liječenju ove bolesti. Iako se pri dijagnosticiranju tumora mozga mogu koristiti razne metode poput magnetske rezonance (MR), kompjuterizirane tomografije (CT), pozitronske emisijske tomografije (PET), lumbalne punkcije, biopsije mozga, praćenja tumorskih markera i mnogih drugih. Najčešće se koriste prve tri. Dodatno, PET i CT se vrlo često koriste zajedno pri dijagnosticiranju kao jedan spojeni postupak [13][14].

Magnetska rezonanca (MR) mozga je najbolja pretraga za detektiranje tumora mozga [13]. MR karakterizira jako magnetsko polje i radiovalovi vrlo visoke frekvencije koji se koriste za dobivanje detaljnih prikaza, u ovom slučaju mozga [9][15]. Kako pregled MR ne sadrži rendgenske zrake (kao npr. postupak kompjuterizirane tomografije koji se sastoji od više slika napravljenih pomoću rendgenskih zraka koje kasnije računalo pretvara u trodimenzionalnu sliku) vrlo je sigurna pretraga za pacijente [14]. Iznimke su osobe koje imaju ugrađene metale u

tijelu poput osoba sa pejsmejkerom [14][15]. Iz navedenih je razloga pogodan za korištenje i kod trudnica. Procedura MR podrazumijeva mehanički stol koji ulazi u veliki cilindrični skener koji producira izuzetno jako magnetsko polje. Ovo polje djeluje na protone (pozitivno nabijene čestice atoma) u tkivu tako da se poredaju prema magnetskom polju [15]. Nakon toga skener emitira puls radiovalova koji nakratko izbacuju protone iz njihova usmjerenja te kada se nakon toga ponovo poredaju (kako su bili i prije) oslobađaju energiju (tzv. signal) [15]. Snaga signala može se razlikovati od tkiva do tkiva, a MR skener snima te signale. Snimljeni se signali računalno obrađuju u svrhu signalne analitike i izrade MR slika [15].



Slika 5: Metastaziranje tumora mozga (MR) (Izvor: [16])

Prilikom upotrebe MR koristi se i kontrastno sredstvo koje sadrži gadolinij (paramagnetsko kontrastno sredstvo) [15][17]. Ovo se sredstvo ubrizgava u venu ili zglob, a svrha mu je, uz pomoć mijenjanja magnetskog polja, učiniti slike jasnijima povećanjem kontrasta među tkivima [15][18][17].

Prije snimanja magnetskom rezonancom osoba mora skinuti svu ili gotovo svu odjeću te dobiva ogrtač koji nema gumbe, kopče, patentne zatvarače ili druge metalne dijelove. Također, svi metalni predmeti sa kojima je osoba došla poput mobitela, nakita, ključeva, kovanica i svih predmeta koji mogu interagirati s magnetskim poljem poput kreditnih kartica i satova trebaju biti smješteni u drugu prostoriju [15][17]. Samo skeniranje može trajati do sat vremena te je glasno pa postoji mogućnost dobivanja slušalica ili čepića za uši ukoliko se zatraži [15].

Neke od prednosti MR nad drugima dijagnostičkim tehnikama su što ne koristi radijaciju (MR je u potpunosti sigurna za sve kategorije ljudi), nije invazivna i bezbolna je, može se primjenjivati na gotovo cijelom tijelu te je posebno dobra za meka tkiva, MR daje detaljniju sliku od npr. CT-a te može detaljno prikazati veličinu ili položaj tumora [13][14][15][17]. Nadalje, MR se preferira kod osoba koje su imale reakciju na jodna kontrastna sredstva koja se koriste od CT-a, a može se koristiti i za mjerenje određenih molekula u mozgu koje razlikuju tumor mozga od apscesa (ograničena nakupina gnoja kao posljedica gnojne upale u tkivu, organu ili tjelesnom prostoru) u mozgu [15][19]. Također, postoji više vrsta MR, neke od poznatijih su: funkcionalna

MR, perfuzijski MR, difuzijski ponderirana MR, magnetska rezonancijska spektroskopija, magnetska rezonancijska angiografija (MRA), magnetska rezonancijska venografija i echo planar snimanje (EPI) [14][15].

Usprkos brojnim prednostima MR ima i neke nedostatke. Dužina snimanja može biti nedostatak koja je duža nego u kompjuterizirane tomografije, MR je skuplja, u pravilu se duže čeka na MR te u hitnim slučajevima poput moždanog udara CT može biti bolja opcija [15]. Neki od ostalih situacija kada se MR izbjegava je ukoliko pacijent pati od klaustrofobije, ima metalne dijelove u tijelu poput šrapnela ili ima implantirane uređaje koji mogu interagirati s magnetskim poljem ili kada osoba razvije reakciju na MR kontrastno sredstvo [15].

Vidljivo je kako je MR za većinu slučajeva najbolja opcija kada se radi o dijagnosticiranju tumora mozga te će se iz tog razloga u poglavlju 7. "Detekcija tumora mozga primjenom YOLOv8 algoritma" koristiti podatci upravo od MR pozitivnih i MR negativnih slika mozga po aspektu tumora mozga za treniranje računalnog programa za ispravnu detekciju tumora u pacijenata na temelju MR slika.

4. Umjetna inteligencija

Umjetna inteligencija ili AI (eng. Artificial Intelligence) je tehnologija koja strojevima i računalima omogućuje sposobnost oponašanja ljudske inteligencije i vještine problemskog razmišljanja [20]. Razvoj umjetne inteligencije se zasniva na strojnom učenju, odnosno na neuronskim mrežama [21]. Neuronske mreže koje su temelj strojnog učenja funkcioniraju oponašanjem neuronskih mreža živih bića koje se sastoje od neurona i sinapsi. Konkretno, umjetne neuronske mreže nastale su kao imitacija ljudskog mozga (više o tome u poglavlju 5. "Neuronske mreže") [22].

Ova tehnika strojevima i računalima omogućuje učenje golemih količina znanja koje su potrebne za npr. komunikaciju s čovjekom na prirodnom jeziku, ovdje se ističu Microsoftov Copilot, Googleov Gemini (bivši Bard) i OpenAI-ov ChatGPT, prvotno lansiran široj populaciji kao prototip krajem 2022. godine koji je i započeo lavinu GPT-eva na što su Google i Microsoft morali reagirati u što kraćem roku kako bi ostali konkurentni na tržištu. Osim mogućnosti komunikacije na prirodnom jeziku strojno učenje omogućuje širok spektar vještina poput učenja temeljem iskustva, donošenja vlastitih zaključaka, prilagodbe ponašanja ovisno o vanjskim faktorima, složeno planiranje i dr. [21]. No, ove sposobnosti još nisu toliko razvijene pa se, za razliku od čovjeka, strojevi i računala u pravilu moraju specijalizirati za uzak raspon mogućnosti.

Umjetna se inteligencija dijeli na slabu i jaku. Svi do sad kreirani oblici umjetne inteligencije se ubrajaju upravo u slabu umjetnu inteligenciju. Jaka umjetna inteligencija bi podrazumijevala opću razinu umjetne inteligencije slične ljudskoj, ili više od ljudske (superinteligencija) [21]. Neki izvori dodatno distinkcioniraju opću umjetnu inteligenciju i superinteligenciju u dvije zasebne razine. U takvim podjelama govorimo o tri razine umjetne inteligencije: sužena umjetna inteligencija (*engl. Artificial Narrow AI*), opća umjetna inteligencija (*engl. General AI* ili *Strong AI*) i umjetna superinteligencija (*engl. Super AI*) [23][24]. Također, i pri ovoj podjeli sva trenutna umjetna inteligencija se ubraja u prvu kategoriju.

4.1. Začetci umjetne inteligencije

Kao začetak priče o umjetnoj inteligenciji svakako bi se mogao, s pravom, definirati Turingov test koji je objavljen 1950. godine od strane Alana Turinga pod nazivom "Računalna tehnologija i inteligencija". Alan Turing (1912.-1954.) bio je engleski matematičar i logičar te se često smatra ocem računarstva [25]. Nadalje, cilj Turingovog testa bio je definirati mogu li računala razmišljati te posjeduju li inteligenciju [26].

Turingov test bio je koncipiran na način da ispitivač (osoba) komunicira s dva subjekta, A i B, na način da s njima razmjenjuje pisane poruke [25]. Jedan od dva spomenuta subjekta jest čovjek, dok je drugi računalo. Ukoliko nakon testa ispitivač ne uspije razlučiti koji od njih (subjekt A i B) je računalo, a koji je čovjek tada se test smatra uspješnim te ga je računalo položilo [25]. Premisa na kojoj se zasniva taj zaključak je da računalo posjeduje jednaku razinu inteligencije kao i čovjek ukoliko ga na prirodnom jeziku kroz komunikaciju nije moguće razlikovati od drugog ljudskog bića [25].

Ovdje se dodatno pojavila problematika: "Može li se inteligentno ponašanje izjednačiti s inteligencijom?". Odnosno, ukoliko se netko ili nešto ponaša, u ovom slučaju kao čovjek, znači li to da je uistinu inteligentan. Glavi argument koji se koristi u ovoj situaciji jest argument kineske sobe [25]. On nalaže, ukratko, ukoliko je neka osoba zaključana u sobu te se sa njom komunicira na kineskom jeziku pomoću pisama kroz otvor za poštu, osoba izvana može imati dojam da osoba unutar sobe zna i razumije kineski jezik iako to uopće ne mora biti slučaj jer se osoba unutar sobe, na primjer, služi opsežnim i detaljnim priručnikom kineskog jezika [25].

Arthur Samuel je 1952. godine razvio prvi program koji je samostalno igrao igru dame i to je bio prvi takav program u bilo kojoj igri. Ernst Dickmann i njegov tim sa Sveučilišta u Münchenu su 1986. godine kreirali prvi automobil bez vozača koji je mogao voziti do 90 kilometara na sat na praznim cestama bez drugih automobila ili prepreka [27]. Godine 1997. "Deep Blue" prvi je računalni program koji je pobijedio svjetskog prvaka u šahu, Garija Kasparova, a razvila ga je kompanija IBM. Godine 2002. proizvedena je prva "Roomba", autonomni robotski usisavač. Apple je 2011. godine proizveo Siri, prvu virtualnu asistenticu, 2012. godine dva znanstvenika iz Google-a (Jeff Dean i Andrew Ng) istrenirali su neuronsku mrežu za prepoznavanje mačaka na raznim slikama. OpenAI je 2020. godine započeo sa beta testiranjem GPT-3 koji je, iako nije bio prvi svoje vrste, kreirao sadržaj koji je bilo gotovo nemoguće razlikovati od ljudskog [27].

Umjetna inteligencija danas je prisutnija nego se to generalno misli. Samo neki od primjera korištenja umjetne inteligencije u svakodnevnom životu su: digitalni asistenti na pametnim mobilnim uređajima i računalima, pametna klimatizacija, autonomni automobili, internet stvari (usisavači, hladnjaci, satovi, prozori...), internetska pretraga, strojni prijevodi, kibernetička sigurnost, borba protiv dezinformacija, optimizacija proizvoda i prodajnih puteva, kupnja i oglašavanje na internetu, roboti u tvornicama, pametna poljoprivreda (pravovremeno navodnjavanje usjeva, hranjenje životinja, uklanjanje korova) i dr. [28].

4.2. Primjena umjetne inteligencije u medicini

Medicina je jedno od područja koje može najviše profitirati od uvođenja umjetne inteligencije te olakšati i ubrzati brojne procese uz bolje i točnije dijagnoze te operacije koje su prije bile nemoguće učiniti mogućima. Jedan od takvih primjera bi bila upravo i dijagnostika raznih pretraga uključujući i MR slike mozga, no više o tome kasnije. Kada govorimo o umjetnoj inteligenciji u medicini možemo je podijeliti u dvije potkategorije: (1) virtualnu i (2) fizičku [29]. U virtualnu umjetnu inteligenciju u medicini ubrajaju se razne aplikacije, od elektronskih sustava zdravstvenih zapisa do sustava za dijagnostiku medicinskih slika temeljenih na neuronskim mrežama, poput ovog algoritma koji je dio završnog rada. U fizičku umjetnu inteligenciju u medicini ubrajaju se razni roboti koji pomažu u izvođenju operacija, inteligentne proteze za starije osobe, itd. [29].

Umjetna inteligencija (UI) u medicini više nije stvar budućnosti, ona je uvelike prisutna i djeluje transformirajuće u vidu obrazaca i procedura koje su se godinama ili stoljećima vršile na jedan način, no implementacijom UI to više nije tako. Primjerice: (1) detektiranje aritmija srca,

ovo stanje je do nedavno bilo vrlo izazovno za otkriti budući da zahtijeva konstantno monitoriranje srca putem elektrokardiograma (EKG) oko pulsa i srčanog ritma, no sad postoje digitalni uređaji za praćenje zdravlja poput AliveCor Kardia koji je uređaj za praćenje EKG-a po medicinskim standardima, a pacijenti ga mogu pratiti od kuće ili gdje god da se nalaze; (2) aplikacije za provjeru kože, one omogućavaju korisnicima da slikaju dijelove kože ako su sumnjičavi oko mogućih lezija te da ih učitaju, nakon čega algoritam umjetne inteligencije uspoređuje i procjenjuje istu sa ogromnim bazama te dolazi do inicijalne dijagnoze u svega nekoliko sekundi koja je vrlo precizna te bi možda bila previđena od strane tradicionalnog medicinskog sustava; (3) detekcija tumora, puno je načina na koje UI može poboljšati performanse u ovom području, od toga da računala mogu pregledati stotine slika u par sekundi, dok medicinskom bi stručnjaku za isti zadatak trebalo mnogo više vremena, osim toga neka istraživanja sugeriraju kako računala pojedinu dijagnostiku mogu raditi i s većom točnošću nego doktori s višegodišnjim iskustvom te bi to rasteretilo medicinsko osoblje i pomoglo u donošenju kvalitetnijih odluka vezanih za svoje pacijente. Također, prema Harvardskoj školi javnog zdravstva, korištenje umjetne inteligencije za postavljanje dijagnoza može smanjiti troškove liječenja do 50% i poboljšati zdravstvene ishode za 40% [30][31][32]. Ovo su samo neki od brojnih primjera primjene UI u medicini koji su transformirali mogućnosti i percepciju kako o medicini tako i o umjetnoj inteligenciji.

Kao što je vidljivo, mogućnosti primjena su mnogobrojne, a područje izuzetno unosno i veliko te vapi za primjenom. Ono premješta u mnogim slučajevima raznu dijagnostiku koja je ranije bila moguća isključivo u medicinskim ustanovama u udobnost vlastitog doma ili uslijed lake prenosivosti i fleksibilnosti u školu, na posao ili putovanje putem mnogobrojnih aplikacija, prijenosnih medicinskih uređaja i sl. Može se naslutiti da postupak detekcije tumora mozga, ili bilo koje vrste, pomoću prepoznavanja malformacija (tumora) na slikama također ima nezaustavljiv potencijal i izuzetnu dodanu vrijednost budući da se radi o, nažalost, smrtnim stanjima ukoliko se ne otkriju dovoljno rano (napose u slučaju tumora mozga).

Upravo ranije navedena postignuća i razvoj područja u ovom smjeru je izvor motivacije za sam rad te YOLO algoritam koji vrlo brzo (u stvarnom vremenu) i efikasno može prepoznavati objekte na slikama za koje je treniran. U ovom slučaju su ti objekti tumori, a slike su slike MR mozga pacijenata.

5. Neuronske mreže

Neuronska mreža, u računalstvu i informatici, skup je međusobno povezanih jedinica zvanih umjetni neuroni (*engl. artificial neurons*) koji oponašaju djelovanje neurona (živčanih stanica) npr. u ljudskom mozgu. Primjer umjetnog neurona je *perceptron* [33]. Matematički model za algoritam učenja perceptrona razvio je još 1958. godine Frank Rosenblatt [34]. Tek nekoliko desetljeća kasnije došlo je do eksplozije ovog područja kada su se sama računala počela drastično i rapidno poboljšavati iz godine u godinu u svojim performansama te je ideja o neuronskim mrežama zaživjela drugi puta kada je matematička ideja postala izvediva [22].

Kasnije se javljaju i takozvane duboke neuronske mreže (*engl. Deep Neural Networks*). Duboke se neuronske mreže sastoje od više slojeva međusobno povezanih umjetnih neurona te su one sposobne riješiti izuzetno složene zadatke temeljem iskustvenog učenja [22]. Jedan od već uobičajenih primjera jest neuronska mreža za prepoznavanje rukom pisanih brojki (ulazi su slike ručno pisanih jednoznamenkastih brojki, dok je izlaza 10, znamenke: 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9 te opcionalno postotak koliko je računalo sigurno u znamenku koju je "ponudilo" kao izlaz) [35][36].

Nadalje, neke od vrsta neuronskih mreža su: (1) *umjetne neuronske mreže* (*engl. Artificial Neural Networks*); (2) *konvolucijske neuronske mreže* (*engl. Convolution Neural Networks*) i (3) *rekurentne neuronske mreže* (*engl. Recurrent Neural Networks*). Umjetne su se neuronske mreže razvile prve te su dobre za rješavanje kompleksnih problema. Konvolucijske neuronske mreže najbolje su pri prepoznavanju slika, dok se rekurentne neuronske mreže najviše koriste za procese vezane uz prirodni jezik [37].

U nastavku bit će nešto detaljnije opisane umjetne neuronske mreže, kako temelj i preteča u neuronskim mrežama u informatici te konvolucijske neuronske mreže koje su relevantne za sam YOLO algoritam i središnju temu ovog završnog rada.

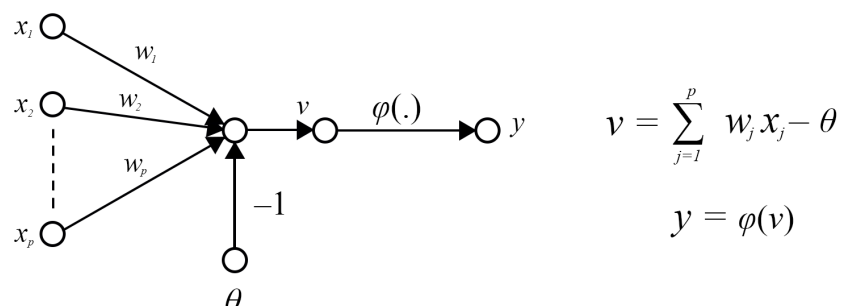
5.1. Umjetne neuronske mreže

Umjetne neuronske mreže nastale su kao direktna imitacija ljudskog mozga te njegovih neurona i međudjelovanja između istih. Još 1943. godine znanstvenici s MIT-a počeli su proučavati ljudski mozak te su predstavili matematički okvir za njegovo djelovanje u okviru teorije automata [22]. Prve izvedbe pojavile su se tek desetljeće kasnije, dok je šira upotreba i potpora računala u rješavanju ostvarena još kasnije u vidu eksponencijalnog rasta i razvoja hardverske opreme kao što je već spomenuto ranije [22].

Ono što je neuron za mozak, to je umjetni neuron za neuronsku mrežu, a najjednostavniji model za umjetni neuron jest perceptron. Perceptron se sastoji od p ulaza te jedinstvenog izlaza, preciznije, radi se o matematičkoj funkciji (funkcija φ na Slici 6) [34].

5.1.1. Perceptron

Važno je napomenuti kako je perceptron najjednostavniji prikaz neuronske mreže te je primjenjiv za uzorke koji se promatraju koji su linearno separabilni, odnosno za takve koji leže na suprotnim stranama hiperravnine [34][38].



Slika 6: McCulloch-Pitts model neurona - perceptron (Izvor: izrada autora)

Ulazi od x_1 do x_p predstavljaju ulazne vektore koje perceptron klasificira u jednu od dvije klase, C_1 ili C_2 . Također, svakom od p ulaza pridružene su težine (*engl. weight*) od w_1 do w_p koje određuju značaj ulaznih parametara. U slučaju klasifikacije u dvije dimenzije granica je pravac definiran jednažbom: $w_1 x_1 + w_2 x_2 - \theta = 0$. Taj pravac ravninu dijeli na dvije poluravnine gdje svaka poluravnina odgovara jednoj klasi (C_1 ili C_2).

Nadalje, prag (*engl. bias*) neurona (θ), je pomak koji se dodaje ulaznom sloju kako bi se perceptronu pružila dodatna fleksibilnost u modeliranju složenih uzoraka u ulaznim podacima, možemo ga prikazati kao fiksni ulaz vrijednosti -1 te pripadnom težinom θ . [30] Tada $p + 1$ dimenzionalni ulazni vektor definiramo kao $x(n) = [-1 \ x_1(n) \ x_2(n) \ \dots \ x_p(n)]^T$, $p + 1$ dimenzionalni ulazni vektor težina definiramo kao $w(n) = [\theta(n) \ w_1(n) \ w_2(n) \ \dots \ w_p(n)]^T$. Izlaz neurona $v(n)$ je realan broj te se računa kao skalarni produkt vektora $x(n)$ i $w(n)$. Dakle, $v(n) = w^T(n)x(n)$.

Budući da govorimo o linearno separabilnim klasama, tada postoji vektor težina w takav da je $w^T x \geq 0$ za svaki $x \in C_1$ i $w^T x < 0$ za svaki $x \in C_2$.

Problem učenja sastoji se u određivanju vektora težine w za koji bi klasifikacija bila ispravna. Za rješavanje ovog problema definira se algoritam učenja na sljedeći način:

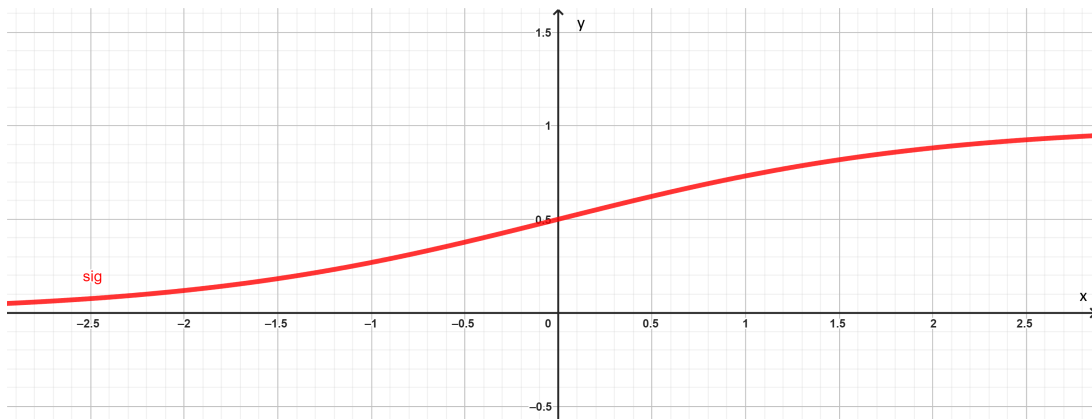
1. Ukoliko je n -ti vektor $x(n)$ korektno klasificiran, težina $w(n)$ se ne mijenja, dakle vrijedi:
 - $w(n + 1) = w(n)$ ako $w(n)^T x(n) \geq 0 \wedge x(n) \in C_1$
 - $w(n + 1) = w(n)$ ako $w(n)^T x(n) < 0 \wedge x(n) \in C_2$
2. Ukoliko je n -ti vektor $x(n)$ nije korektno klasificiran, težina $w(n)$ se mijenja, vrijedi sljedeće:
 - $w(n + 1) = w(n) - \eta(n)x(n)$ ako $w(n)^T x(n) \geq 0 \wedge x(n) \in C_2$
 - $w(n + 1) = w(n) + \eta(n)x(n)$ ako $w(n)^T x(n) < 0 \wedge x(n) \in C_1$

pri čemu je $\eta(n)$ pozitivan parametar koji određuje brzinu učenja [34].

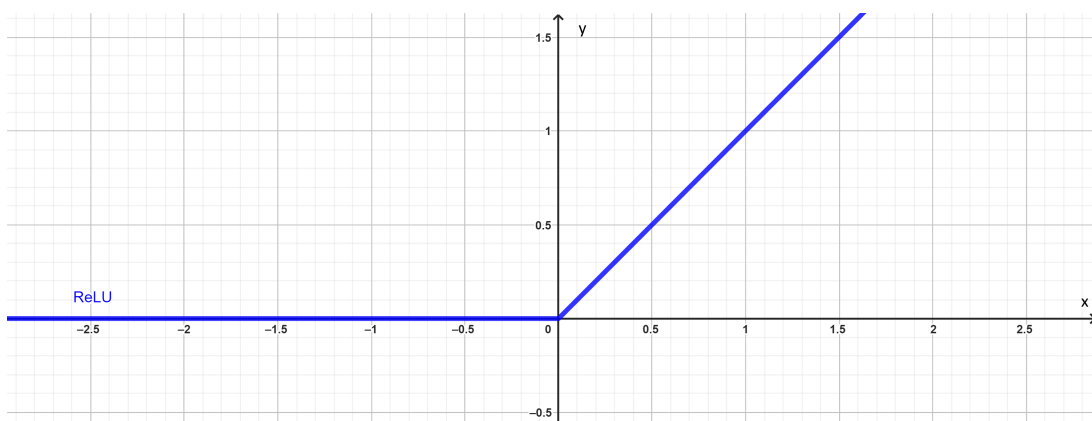
5.1.2. Aktivacijske funkcije

Važno je primijetiti da ova funkcija ($y = \varphi(v)$), kako je trenutno navedena za izlaz po- prima samo dvije vrijednosti (u ovom slučaju su to $y = -1$ i $y = 1$). Upravo iz tog razloga ovakav model kada se spoji sa više istih perceptrona u mrežu veće dubine, odnosno sa više međuslojeva, ne daje previše reprezentativne rezultate zato što minimalna promjena u jednom od njenih jednoslojnih perceptrona može drastično utjecati na ishod cijele mreže [22].

Ovo je razlog iz kojeg se javila potreba za prikazom izlaznih vrijednosti koje mogu po- primiti sve realne brojeve u nekom intervalu (najčešće se uzima interval $[0, 1]$). Na taj način ne postoji rubni slučaj koji bi napravio veliki skok u izlaznoj vrijednosti kao kod ranije opisanog modela [22]. Jedna od često korištenih takvih funkcija je sigmoidalna funkcija koja je prikazana na slici 7 (dolje), a definirana je formulom: $sig(x) = \frac{1}{1+e^{-x}}$.



Slika 7: Sigmoidalna funkcija (Izvor: izrada autora)



Slika 8: ReLU funkcija (Izvor: izrada autora)

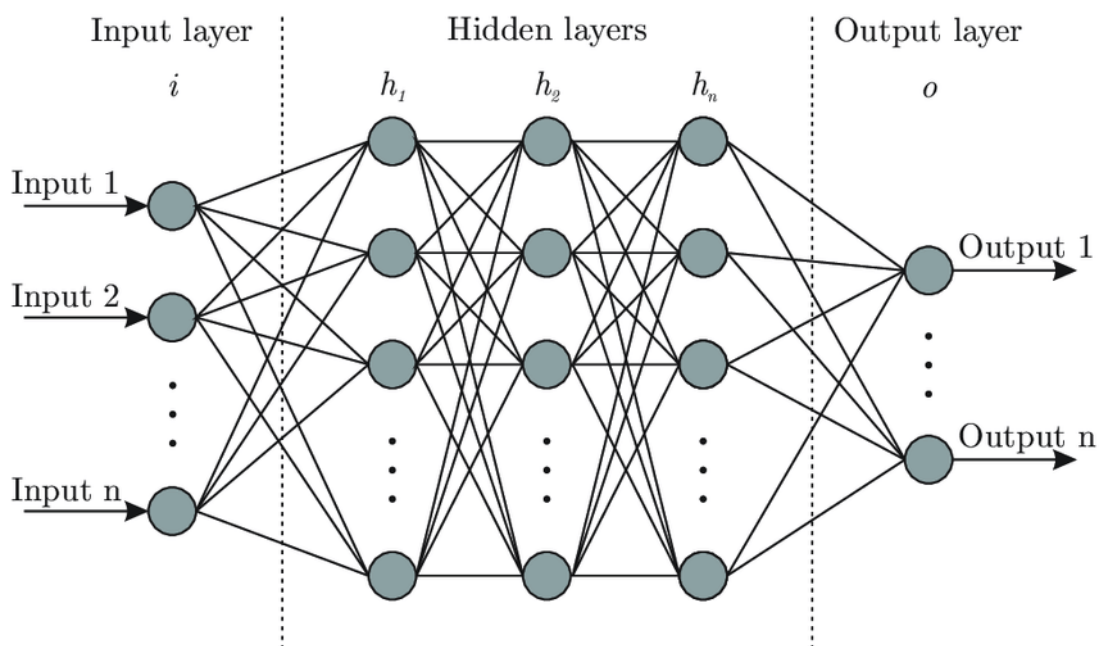
Prije se u ovu svrhu najviše koristila sigmoidalna funkcija koja za bilo koje ulazne pa- rametre vraća vrijednost upravo iz ranije spomenutog intervala $[0, 1]$, no kasnije se pokazalo kako neuronske mreže puno brže uče uz funkciju imena: *ispravljena linearna jedinica* (engl. *rectified linear units, ReLU*). Funkcija je definirana kao $ReLU(x) = \max\{0, x\}$ (vidi sliku 8.).

Ideja za primjenom ove funkcije u neuronskim mrežama proizašla je iz grubog oponašanja prirodnih neurona koji se aktiviraju tek ukoliko dođe do podražaja koji prelazi određeni prag (*engl. threshold*) te se sukladno jačini podražaja izlaz gradi. U suprotnom je aktivacijski izlaz jednak nuli [35][39].

5.1.3. Struktura neuronske mreže

Neuronske mreže se sastoje od mnogo umjetnih neurona koji su međusobno povezani u jedinstvenu mrežu. Ti umjetni neuroni su osnovne gradivne jedinice ovakvih mreža (kao što su prirodni neuroni osnovne gradivne jedinice prirodnih neuronskih mreža u živim bićima koja imaju živčani sustav) [22]. Kada se govori o umjetnim neuronskim mrežama često se spominje i njihova dubina. Dubina umjetnih neuronskih mreža određena je brojem slojeva od kojih je ista građena. U slučaju najprostijeg perceptora, čija je funkcija i izgled opisan u poglavlju 5.1.1. njegova je dubina jednaka jedan. Upravo zato što se sastoji "samo od sebe".

Generalno govoreći razlikujemo tri osnovna sloja neuronskih mreža: (1) *ulazni sloj* (*engl. input layer*), koji je prvi te prima ulazne podatke iz svoje okoline; (2) *izlazni sloj* (*engl. output layer*), koji daje krajnji rezultat neuronske mreže (uglavnom je dozvoljeno da samo jedan umjetni neuron u izlaznom sloju ima aktiviran output); (3) *skriveni sloj/slojevi* (*engl. hidden layers*), to su svi slojevi koji se nalaze između ulaznog i izlaznog sloja. Može ih biti i više desetaka ili stotina u istoj neuronskoj mreži, a može biti i samo jedan takav [40].



Slika 9: Struktura umjetne neuronske mreže (Izvor: [40])

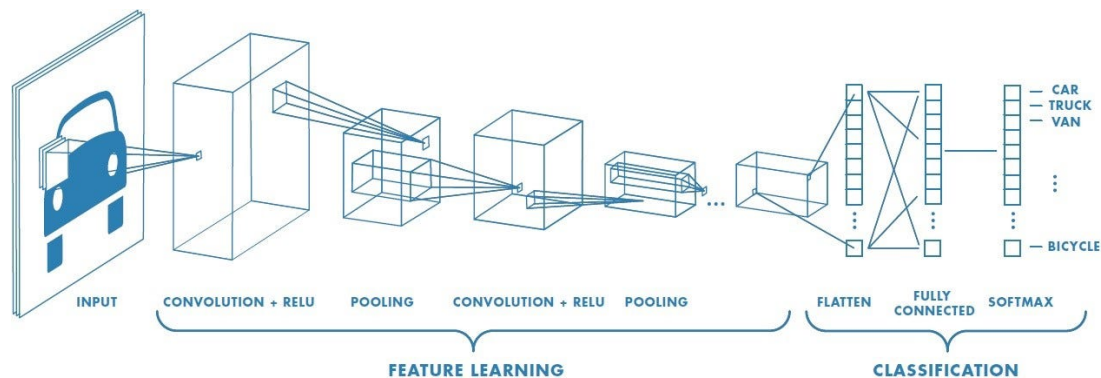
Kao što je to vidljivo i na slici 9, izlazi jednog sloja služe kao ulazi drugog. Takve neuronske mreže gdje izlazi prethodnog sloja služe kao ulazi sljedećeg nazivamo unaprijedne neuronske mreže. kada se govori o neuronskim mrežama generalno se misli upravo na te [22]. Također, možemo govoriti i potpuno povezanim neuronskim mrežama, koje su i uobičajene. U potpuno povezanih neuronskih mreža svaki neuron iz prethodnog sloja svojim izlazima je

povezan na svaki neuron iz sljedećeg sloja te je svaki neuron iz $i + 1$ -og sloja svojim ulazima povezan sa izlazom svakog od neurona i -tog sloja. Dok, kada govorimo o djelomično povezanim neuronskim mrežama ili nepotpunim neuronskim mrežama, to nije slučaj, već su veze unutar mreže samo podskup svih veza koje je moguće napraviti na prethodno opisan način (engl. *PCNN - partially connected neural network*) [41].

5.2. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (CNN)(engl. *Convolutional Neural Networks*) mogu se promatrati upravo kao nadogradnja na obične višeslojne unaprijedne neuronske mreže. Konvolucijske neuronske mreže također imaju jedan ulazni, jedan izlazni te više skrivenih slojeva [22]. U konvolucijskim su neuronskim mrežama specifični konvolucijski slojevi (engl. *convolutional layers*) i slojevi sabiranja (engl. *pooling layers*), a često se koriste i potpuno povezani slojevi (engl. *fully connected layers*) [22][39].

Kao što je to i ranije spomenuto, konvolucijske se neuronske mreže koriste primarno za prepoznavanje objekata na fotografijama ili videozapisima te se koriste u području računalnog vida (engl. *computer vision*). Arhitektura konvolucijskih neuronskih mreža temelji se na istim principima kao i vizualni korteks u ljudskome mozgu. Odnosno, nisu svi neuroni zaduženi za cijelo vidno polje već obrađuju samo njegov dio za koji su zaduženi [39]. Analogno tome u konvolucijskim neuronskim mrežama umjetni su neuroni zaduženi samo za male jedinice neke fotografije te u cijelosti čine particiju slike gdje sliku čine svi njezini pikseli.



Slika 10: Jednostavni prikaz slojeva konvolucijske neuronske mreže (Izvor: [42])

Sa slike 10 vidljivo je kako je proces moguće podijeliti u dva glavna dijela: (1) *izvlačenje značajki* (engl. *feature learning*) i (2) *klasifikacija* (engl. *classification*). Prvi se dio sastoji od dva sloja: konvolucije i grupiranje, dok se drugi dio sastoji od potpuno spojene mreže koja je zaslužna za klasifikaciju [39]. Dakle, razlikujemo tri glavna sloja u konvolucijskim neuronskim mrežama: konvolucijski sloj, sloj sabiranja ili grupiranja te sloj potpuno povezane mreže [37]. Osim ova tri sloja i dalje postoje i ulazni i izlazni sloj kao što je to bio slučaj i kod umjetnih neuronskih mreža. Također, najčešće se koristi više iteracija konvolucijskog sloja te sloja sabiranja jedan iz drugoga [22].

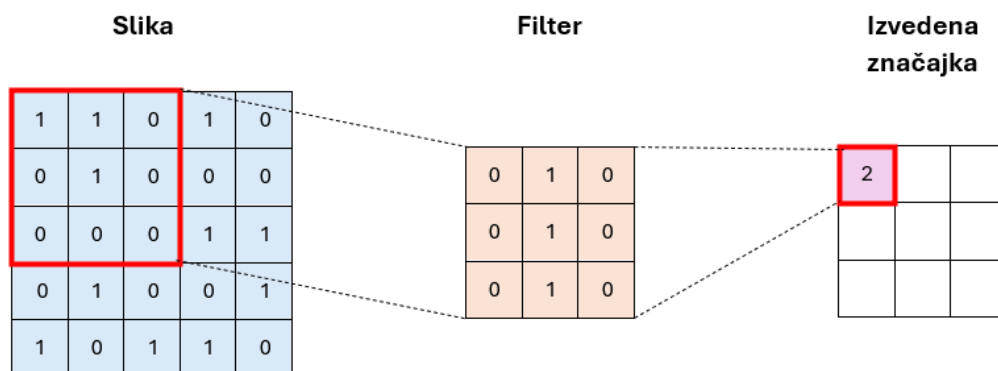
5.2.1. Konvolucijski sloj

Konvolucija je: "Matematička funkcija nastala integriranjem umnoška dviju funkcija po intervalu njihove definicije gdje su te funkcije ravnopravne tako da svaka infinitezimalna promjena jedne funkcije utječe na drugu funkciju u cijelome intervalu definicije." [43]. Konvolucija daje treću funkciju kao rezultat spajanja neke dvije. Definirana je matematičkom formulom na način kao što je prikazano u izrazu 5.1 u nastavku:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (5.1)$$

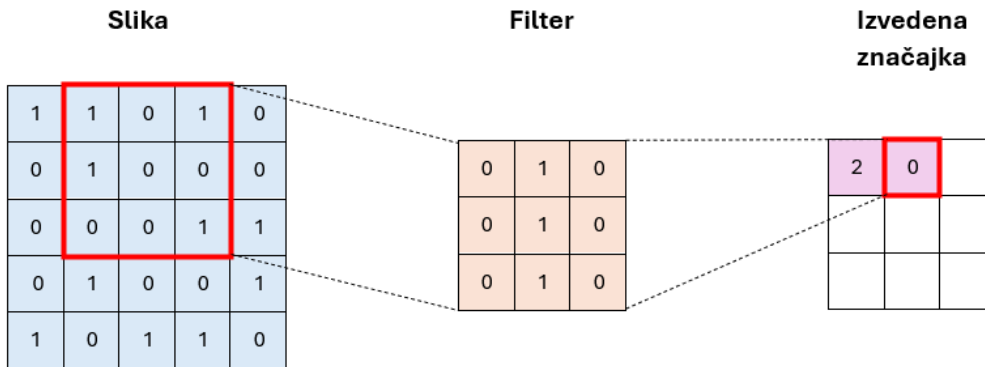
Kod konvolucijskih neuronskih mreža pojam konvolucija se odnosi na prepoznavanje značajki neke slike uz pomoć filtera (*engl. Kernel - jezgra*). Na taj se način dobivaju izvedene značajke slike [39].

Filter, u kontekstu CNN-a, jest matrica vrijednosti. Ta matrica ne mora biti jednaka ulaznoj matrici i u pravilu je manja od ulazne matrice koja se sastoji od rezolucije ulazne slike, npr. HD (*engl. High Definition*) slike su dimenzija 1280×720 piksela, dakle to bi bila veličina ulazne matrice za tu sliku. Veličina matrice filtra za nju bi mogla biti npr. 5×5 . No, dubina filtra uvijek mora biti jednaka dubini ulaza [22]. Kako se boje piksela na slici klasificiraju najčešće pomoću tročlanog vektora sa vrijednostima svakog člana od 0-255 (RGB klasifikacija), to znači da je dubina u ovom slučaju tri te da trebaju postojati tri matrice filtra (po jedan za svaku komponentu boje) [22][44].



Slika 11: Izvršavanje konvolucije uz pomoć filtra 1. iteracija (Izvor: izrada autora)

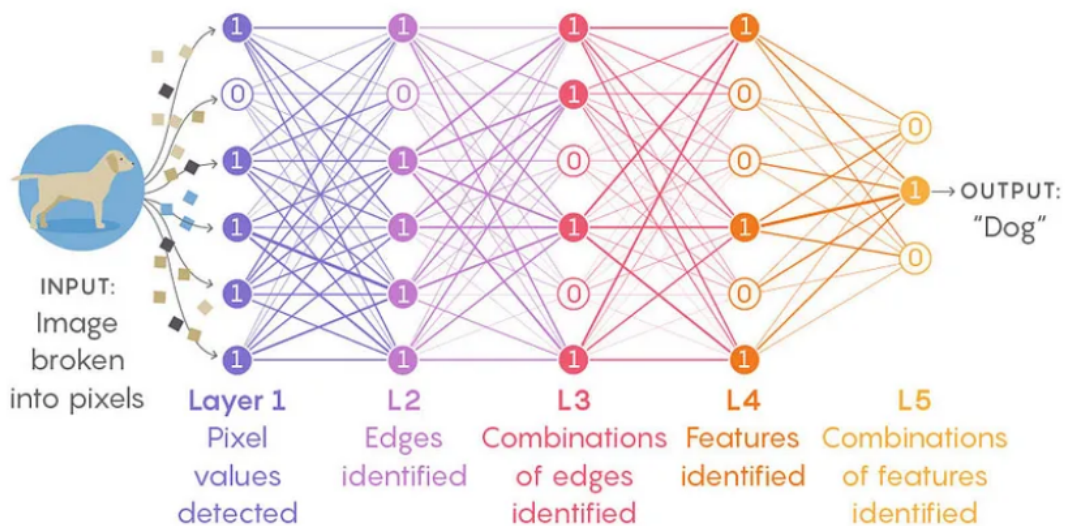
Na slikama 11 i 12 prikazano je izvršavanje konvolucije uz pomoć filtra. Plava tablica predstavlja ulaznu sliku veličine 5×5 u matricnom obliku, filter je prikazan narančastom tablicom dimenzija 3×3 te je tablica izvedene značajke istih dimenzija kao i filter. Filter se primjenjuje na ulaznu sliku na način da prolazi preko njenih podskupova dimenzije 3×3 počevši od lijevog gornjeg kuta pa sve do njenog desnog donjeg kuta. U svakom od tih podskupova veličine 3×3 pokušava se pronaći odgovarajuća karakteristika na način da se nad vrijednostima unutar početne tablice podskupa ulaza (na slici 11 označena sa lijevim crvenim kvadratom) provodi konvolucija tablicom filtra rotiranom za 180° (prema definiciji konvolucije) te se rezultat upisuje u odgovarajuće polje u tablici izvedene značajke [44]. Nakon toga filter se pomiče udesno za



Slika 12: Izvršavanje konvolucije uz pomoć filtra 2. iteracija (Izvor: izrada autora)

unaprijed određeni korak (*engl. stride*) [39]. Kada filter dođe do kraja "reda" pomiče se za unaprijed određeni korak prema dolje te se vraća na početak i prema desno prolazi kroz taj, novi, "red".

Uobičajeno je kako su prvi konvolucijski slojevi zaduženi za općenitije značajke te se sve više i više specijaliziraju kroz svoje iteracije [39][40]. Jedan primjer gradijalne specijalizacije značajki prikazan je na slici 13 (dolje). Na slici 13 raspodjela posla po iteracijama u prepoznavanju pasa sa slike je sljedeća: identificiranje vrijednosti piksela, identificiranje rubova objekta (u ovom slučaju psa), identificiranje kombinacija rubova objekta, identificiranje značajki psa te na kraju identificiranje skupova značajki psa [40].



Slika 13: Moguće zadaće skrivenih slojeva u prepoznavanju objekata na slici (Izvor: [40])

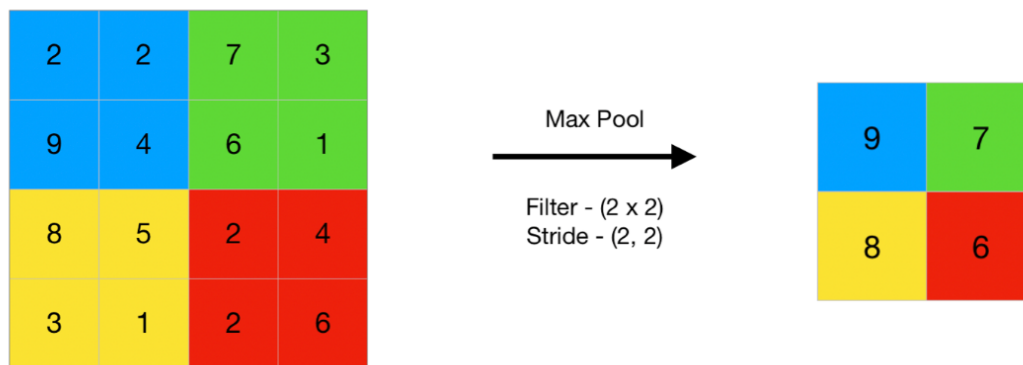
Također, rezultat konvolucijskog sloja može biti jednakih dimenzija u odnosu na ulaz, no može biti i manjih (slike 11 i 12) ili čak većih od početnog ulaza. Obloge (*engl. padding*) se mogu koristiti za te svrhe na način da se ulazna slika nadopuni dodatnim pikselima oko rubova slika [39]. Ti novi pikseli se najčešće inicijaliziraju na vrijednosti nula kako ne bi utjecali na postupak traženja određenih značajki na slici.

5.2.2. Sloj sabiranja

Slojevi sabiranja, (*engl. pooling layers*) također izvršavaju određene operacije nad ulaznom slikom. "Oni su odgovorni za smanjenje prostornih dimenzija ulaznih podataka, u smislu širine i visine, a zadržavaju najvažnije informacije." [45]. Smanjivanje dimenzija koristi se zbog smanjenja potrebne snage računala za obradu slike te smanjivanja potrebnih konvolucijskih slojeva do matrica oblika 1×1 [46].

Na primjer, za sliku rezolucije 1000×1000 piksela bez slojeva sabiranja bi trebalo 500 konvolucijskih slojeva da se dođe do 1×1 matrice, dok bi sa slojevima sabiranja koji smanjuju matricu pri svakoj iteraciji za faktor razine 2 trebalo samo 17 slojeva ukupno (8 konvolucijskih, 8 sabiranja te 1 potpuno povezani sloj) [46].

Pri sabiranju najčešće se koriste sljedeće vrste, *max pooling*, *average pooling*, *global max pooling* i *global average pooling* [47]. Max pooling radi na način da se uzima najveća vrijednost iz prozora kojeg trenutno obuhvaća filter (slika 14) [47].



Slika 14: Primjer sabiranja korištenjem max pooling-a (Izvor: [48])

Na slici 14 dimenzije ulaznih podataka su 4×4 te je izlaz manji za faktor 2 dakle, izlaz je dimenzija 2×2 . Filter je dimenzija 2×2 te je korak jednak 2, 2 (pomiče se za dva mjesta u desno dok ne dođe do kraja retka te se vraća na početak i spušta također za dva mjesta odakle kreće ponovo u koracima od dva prema desno itd.). Odnosno, u prvom koraku filter se nalazi iznad podskupa tablice 4×4 obojanog plavom bojom te uzima maksimalnu vrijednosti, u ovom slučaju je to broj 9, te ju upisuje u plavo polje u 2×2 izlaznoj tablici. Nakon toga filter se pomiče za dva polja udesno te se nalazi iznad podskupa tablice 4×4 obojanog u zeleno; njegov maksimum je 7. Analogno se izlazna matrica popuni do kraja.

Da se radilo o *average pooling*-u tada bi plava polje u izlaznoj tablici bilo popunjeno sa vrijednosti 4.25, koja odgovara aritmetičkoj sredini odgovarajuća četiri broja u prvom koraku ($((2 + 2 + 9 + 4)/4 = 4.25)$), zeleno polje bi također poprimilo vrijednost 4.25 kao i žuto, dok bi crveno imalo vrijednost 3.5.

Global max pooling i *global average pooling* pretvaraju 3-dimenzionalni ulaz (npr. 3 matrice svaka za jednu komponentu RGB klasifikacije piksela) u jednodimenzionalni izlaz (jednodimenzionalni vektor). Ovaj pristup se koristi kada se CNN ili dio za izdvajanje značajki želi

spojiti na potpuno povezani dio mreže [47]. Uz to, kada govorimo o *global max pooling-u* i *global average pooling-u*, dimenzije filtra jednake su dimenziji ulaza te ove dvije vrste sabiranja uzimaju maksimum odnosno aritmetičku sredinu cijele tablice ulaza.

6. YOLO algoritam

YOLO (*You Only Look Once*) algoritam primarno se koristi za detekciju objekata na slikama ili videozapisima koji je 2015. godine razvio Joseph Redmon, a napisan je u frameworku Darknet [49]. Kao što mu samo ime sugerira, podoban je i za detekciju u stvarnom vremenu (*engl. Real-Time*). Ovo ga čini sposobnim za korištenje u autonomnoj vožnji, kamerama za prepoznavanje lica, na sportskim natjecanjima za praćenje igrača itd. [50].

Suprotno prethodnom radu u području detekcije objekata gdje su rađene prenamjene klasifikatora za detekciju, YOLO algoritam se bazira na problemu regresije na način da sliku dijeli na njenu particiju sastavljenu od pravokutnih podskupova i računa vjerojatnosti pojedinih podskupova, te za to koristi CNN [51][52]. Granične okvire tih podskupova kao i vjerojatnosti klase predviđa neuronska mreža izravno iz pune slike u jednoj procjeni [51]. Prednost arhitekture koju koristi YOLO jest brzina. "YOLO model obrađuje slike u stvarnom vremenu u 45 sličica po sekundi. Manja verzija mreže, Fast YOLO, obrađuje nevjerojatnih 155 sličica u sekundi dok još uvijek postižu dvostruko veći mAP (*engl. mean average precision*) od drugih detektora u stvarnom vremenu. [51]". mAP označava srednju vrijednost prosječnih preciznosti. Prethodno navedeno odnosi se na inicijalnu verziju YOLO algoritma, kasnije verzije su dodatno optimizirane te su rezultati u vidu brzine i točnosti detekcije još uspješniji.

Model je postao popularan nakon što su ga njegovi kreatori, Joseph Redmon, Santosh Divvala, Ross Girshik i Ali Farhadi, predstavili na konferenciji CVPR Computer Vision and Pattern Recognition Conference 2016. godine te tim modelom čak osvojili nagradu: OpenCV People Choice Awards [49].

YOLO algoritam trenutno ima 10 temeljnih verzija, od prve, YOLOv1 verzije, preko YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv8, do YOLOv9 i najnovije YOLOv10 verzije [49]. Osim spomenutih verzija javljale su je i neke međuverzije. U nastavku biti će detaljno objašnjen princip rada prve verzije YOLO algoritma, te nastavno u cjelini 7. ovog rada bit će istaknute bitnije razlike između YOLOv1 i YOLOv8, verzije koja se koristi u tom, praktičnom, dijelu rada. Razlog korištenja YOLOv8 umjesto npr. YOLOv9 ili najnovije YOLOv10 verzije koje su obje objavljene tek ranije ove godine jest činjenica kako još uvijek nisu dovoljno testirane i pouzdane. Iz istog razloga YOLOv9 i YOLOv10 još uvijek nisu u potpunosti zaživjele.

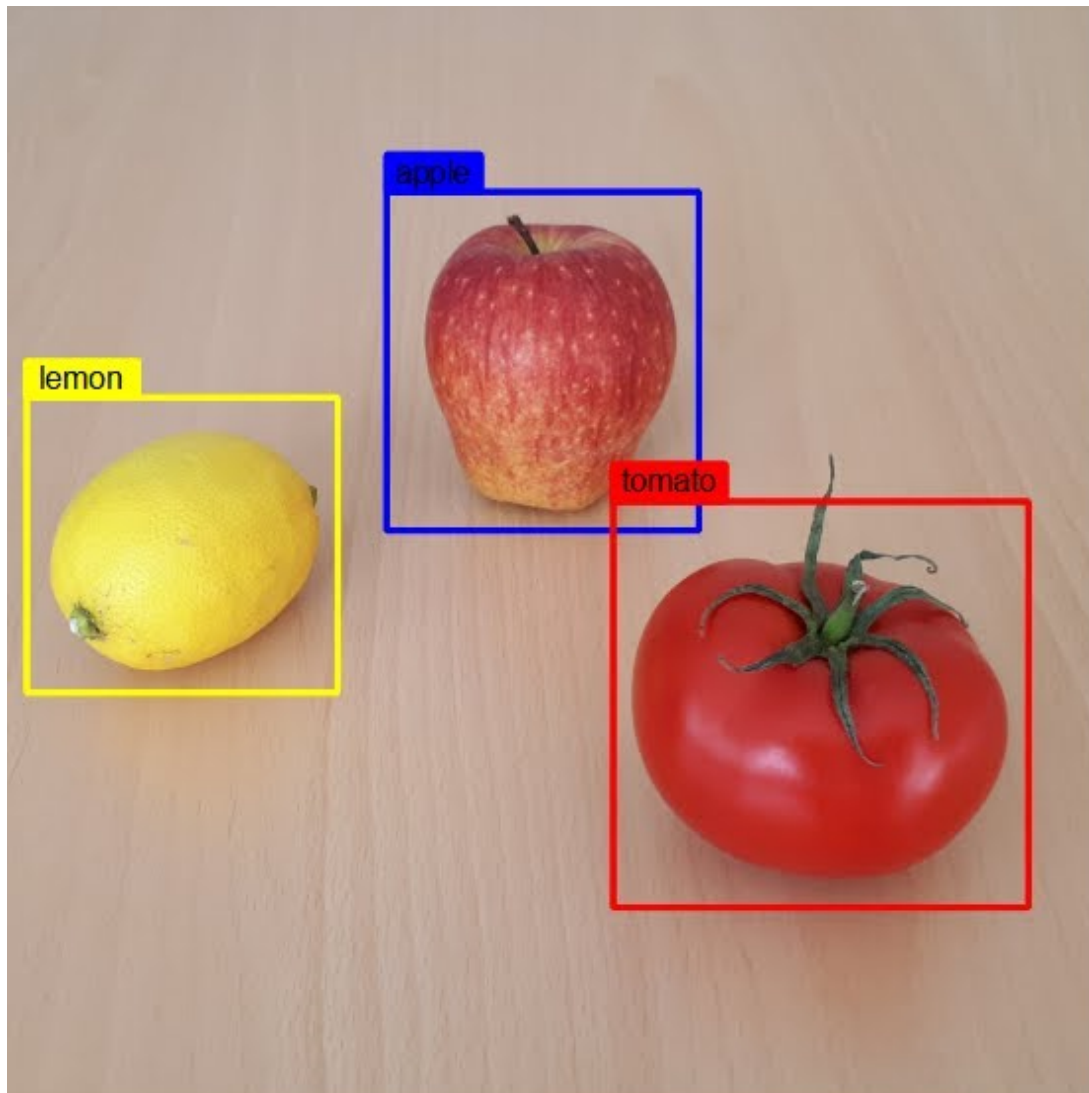
6.1. YOLOv1 algoritam

Kao što je to ranije spomenuto, prvi YOLO algoritam nastao je 2015. godine, a napisan je u prilagođenom frameworku Darknet. [49] U istom frameworku će kasnije biti napisane i sve ostale verzije ovog algoritma uključujući i YOLOv8 koja se koristi u programskom kodu ovog rada [49].

Prva verzija YOLO algoritma trenirana je na PASCAL VOC (*engl. Visual Object Classes*) skupu podataka [53]. Aktualnije verzije poput YOLOv8 verzije uglavnom koriste Microsoft COCO skup podataka za treniranje o čemu će biti pisano kasnije. PASCAL VOC skup podataka

sastojao se od 20 klasa objekata, ti objekti su: (1) zrakoplov; (2) bicikl; (3) brod; (4) autobus; (5) ptica; (6) boca/flaša; (7) mačka; (8) krava; (9) automobil; (10) stolac; (11) pas; (12) blagovaonski stol; (13) konj; (14) motocikl; (15) osoba; (16) biljka u vazi; (17) ovca; (18) kauč; (19) televizijski ekran/monitor i (20) vlak [53][54].

Cilj algoritma je ispravno postaviti granične okvire objektima koje pokušavamo detektirati na slici te reći što se nalazi u tim okvirima koje smo dodijeliti objektima.

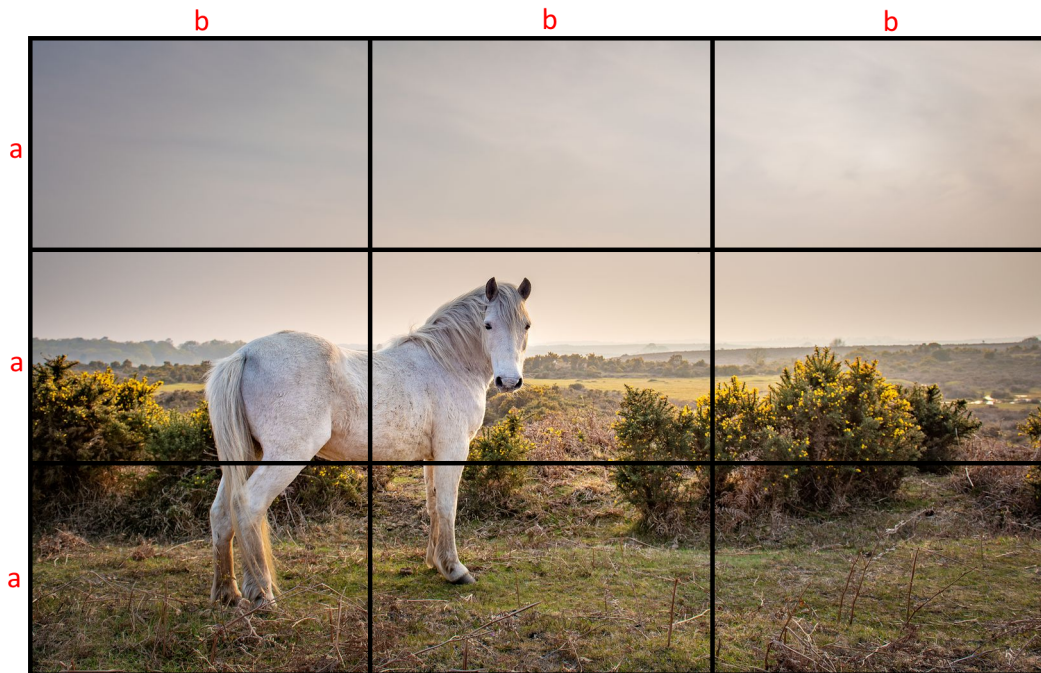


Slika 15: Primjer detekcije korištenjem YOLOv1 algoritma (Izvor: [55])

6.1.1. Matematička pozadina YOLOv1 algoritma

Prvi korak YOLO algoritma je da podijeli sliku na rešetku sastavljenu od $S \times S$ ćelija [51][53]. Ukoliko bi odredili da vrijedi $S = 3$, tada bi rešetka bila oblika 3×3 te bi mogla izgledati nešto poput slike 16. (u nastavku). Vidljivo je kako svaka od tih S^2 ćelija ne mora biti kvadratnog oblika sa duljinama stranica $a \times a$, već može biti i pravokutnog oblika, kao što je to slučaj na slici 16. sa stranicama duljina $a \times b$.

Za detekciju pojedinog objekta sa slike odgovorna je ona ćelija rešetke u kojoj se nalazi



Slika 16: Primjer 3×3 rešetke nad slikom (Izvor: [56]) uz doradu autora

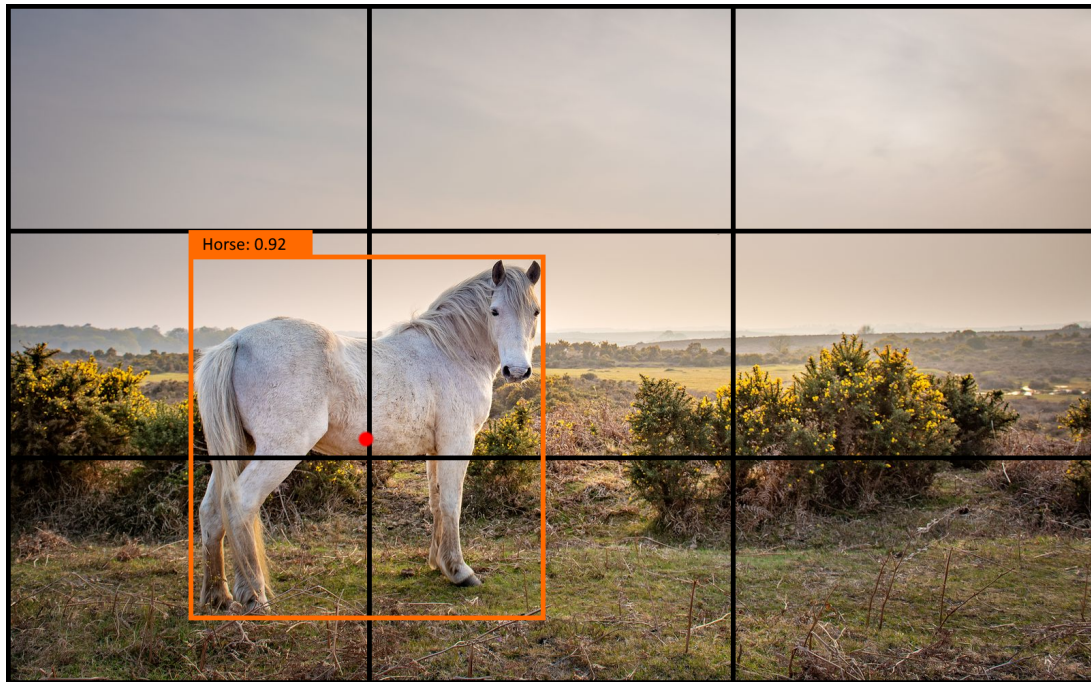
njegovo središte. U primjeru sa slike 16. objekt koji se pokušava pronaći jest konj. Na slici 17. (dolje) prikazan je mogući granični okvir za konja prikazan narančastom bojom. Za granični okvir objekta prikazano je i njegovo središte crvenom točkom. Vidljivo je kako se crvena točka nalazi u ćeliji koja pripada drugom retku i prvom stupcu ove 3×3 rešetke. Za svaku ćeliju sa slike definirana je jedinstvena klasifikacija varijabli $i, j \in \{1, \dots, S\}$. U ovom slučaju vrijednosti i, j su elementi skupa $\{1, 2, 3\}$. Konkretno, ćelija koja sadrži središte prikazanog graničnog okvira jest *ćelija*_{2,1}.

Nadalje, primjenom YOLO algoritma svaka ćelija generira B graničnih okvira zajedno sa pripadnom procjenom pouzdanosti (*engl. confidence*), tj. taj parametar govori koliko je algoritam siguran da se na slici nalazi ono što je napisao da se nalazi unutar granica okvira koje je odredio (na slici 17. procjena pouzdanosti poprimila je vrijednost 0.92 te je klasa koju je algoritam prepoznao *horse*, odnosno hrvatski *konj*) [57]. Vrijednost varijable B u YOLOv1 algoritmu postavljena je na $B = 2$. Parametar pouzdanosti prikazuje se brojkom od $[0, 1]$ gdje je 0 najmanja, a 1 najveća razina pouzdanosti [58]. Ako se u i -toj ćeliji $S \times S$ mreže ne nalazi objekt tada bi parametar predikcije trebao biti blizu 0 (u idealnom slučaju predikcije algoritma jednak 0), ukoliko se na slici nalazi objekt tada se on računa prema izrazu 6.1:

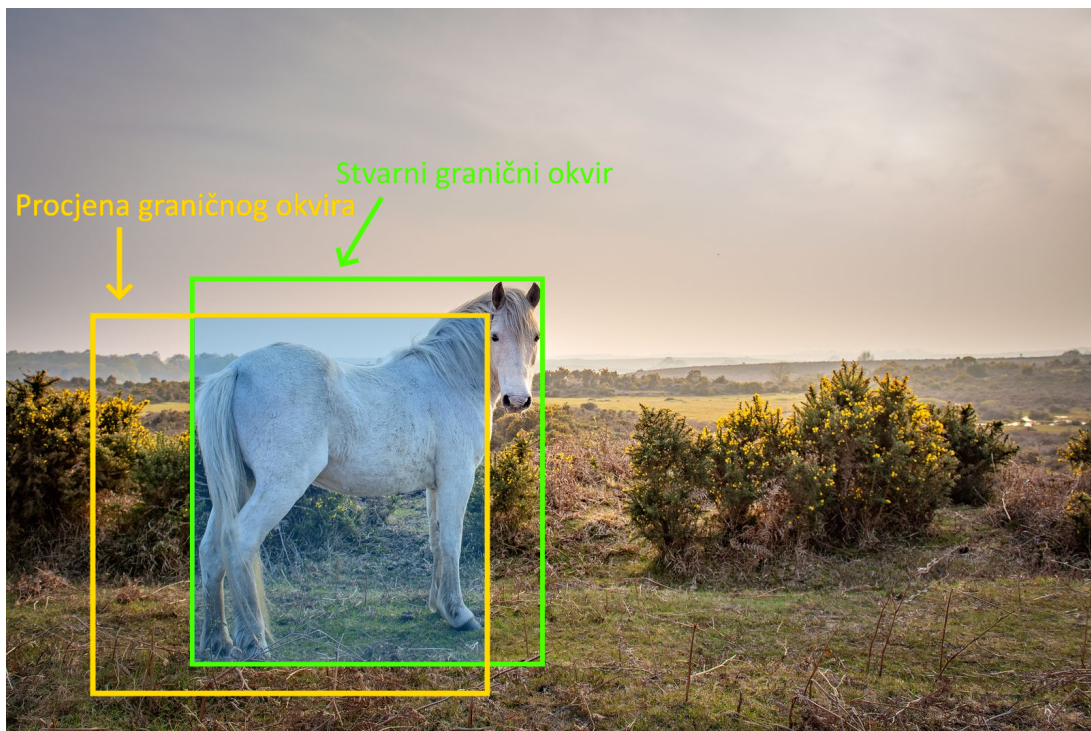
$$\text{pouzdanost} = Pr(\text{Object}) * IOU_{pred}^{truth} \quad (6.1)$$

gdje IOU_{pred}^{truth} (*engl. Intersection Over Union*, kratko *IOU*) označava kvocijent presjeka i unije stvarnog okvira (*engl. ground truth*) i njegove predikcije. Tj. ovaj se postupak vrši za svaku od B predikcija graničnih okvira te uzima onaj sa najvećom vrijednošću *IOU*-a [59][60].

Na slici 18. (gore) stvarni okvir objekta (u ovom slučaju konja) prikazan je zelenom



Slika 17: Primjer graničnog okvira (Izvor: [56]) uz doradu autora)



Slika 18: Primjer rada IOU -a na slici (Izvor: [56]) uz doradu autora)

bojom, dok je jedna (od mogućih B) predikcija graničnog okvira modela prikazana žutom bojom. Dodatno, plavom je bojom naznačen presjek ova dva okvira. Dakle, razumljivo je kako IOU vrijednost poprima vrijednosti iz intervala $[0, 1]$ (budući da vrijedi: $A \cap B \leq A \cup B$) (slika 19. (dolje)).

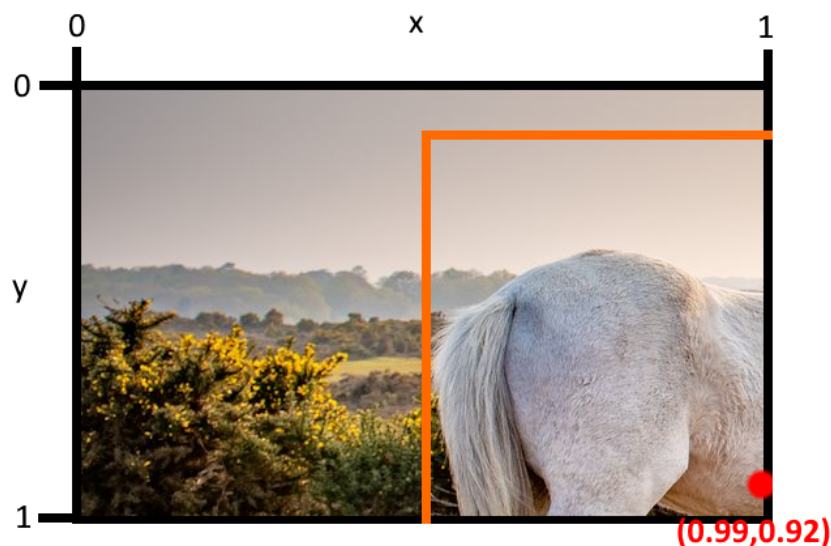
Brojevi bliže 1 predstavljaju bolje rezultate, predikcije graničnih okvira, dok brojevi bliže

$$IoU = \frac{\text{Površina preklapanja}}{\text{Površina unije}}$$

Slika 19: Vizualni prikaz IOU formule (Izvor: izrada autora)

0 lošije. Predikcija graničnog okvira za koju je $IOU = 1$ je idealna (predikcijski okvir podudara se sa stvarnim graničnim okvirom), dok se predikcija za koju je $IOU = 0$ nimalo ne poklapa sa stvarnim okvirom dotičnog objekta. Osim IOU metode za odbacivanje nepotrebnih i suvišnih graničnih okvira može se prije nje koristiti i metoda NMS (*engl. non-maximum suppression*). U ovoj se metodi arbitrarno (odluka IT ili drugog stručnjaka) odredi neka vrijednost pouzdanosti (*engl. confidence*), npr. $c = 0.5$ te će svi granični okviri prethodno detektirani na slici, a koji imaju $c \leq 0.5$ biti odbačeni [58]. Zatim slijedi drugi korak ove metode koji preostale granične okvire rangira prema pouzdanosti te odabire samo onaj sa najvećom vrijednošću pouzdanosti kao ispravan za neki objekt ukoliko ne postoji definiran stvarni granični okvir za njega [58].

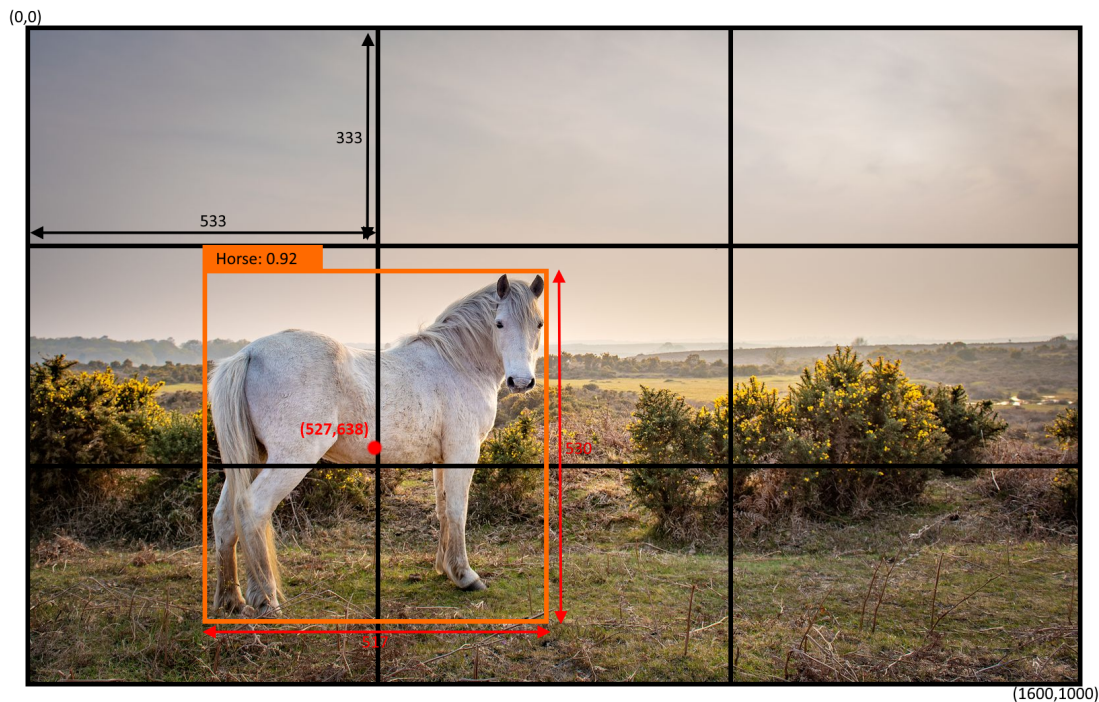
Svaki okvir za koji je zadužena pojedina ćelija biti će definiran pomoću 5 parametara: x , y , h , w i *pouzdanost*. x i y predstavljaju koordinate gdje se nalazi središte okvira u odnosu na ćeliju u kojoj se nalazi pripadajuće središte na način da je gornji lijevi kut označen kao $x = 0$ i $y = 0$, a donji desni kao $x = 1$ i $y = 1$. Kao što je vidljivo na slici 20, vrijedi da su: $\forall x \text{ i } \forall y \in [0, 1]$.



Slika 20: Uvećana ćelija na poziciji (2,1) (Izvor: [56]) uz doradu autora)

Kažemo da se x i y koordinate izražavaju relativno u odnosu na ćeliju. U primjeru sa slike budući da se središte predikcijskog okvira za objekt nalazi u neposrednoj blizini donjeg

desnog kuta ćelije, znamo kako će i x i y varijable biti brojke vrlo blizu jedinici sa manje strane. Na slici piše kako je $x = 0.99$, što je logično budući da crvena točka središta dodiruje sam crni rub ćelije, a $y = 0.92$ što se odokativno također čini kao realna vrijednost u odnosu na poziciju točke ovisno o osi ordinata. Konkretno, x koordinata na primjeru sa slike 20, odnosno slike 21 (dolje) računa se kao $x = \frac{527-0}{533} \approx 0.99$, a y koordinata računa se kao $y = \frac{638-333}{333} = \frac{305}{333} \approx 0.92$ pri čemu su $(527, 638)$ apsolutne koordinate središta graničnog okvira [61]. Osim crvene točke središta graničnog okvira objekta na slici 21. vidljiv je i sam granični okvir prikazan narančastom bojom.



Slika 21: Određivanje koordinata graničnog okvira (Izvor: [56]) uz doradu autora)

Nadalje, w i h predstavljaju redom širinu (*engl. width*) i visinu (*engl. height*) okvira objekta. Ove dvije mjere mogu poprimiti samo vrijednosti iz intervala $(0, 1]$. No, za razliku od parametara x i y koji su se izražavali relativno u odnosu na ćeliju, a parametri h i w izražavaju se relativno u odnosu na cijelu sliku [51][52][53]. Može se zaključiti kako se, generalno, vrijednosti parametara w, h računaju redom kao širina graničnog okvira podijeljeno sa širinom cijele slike, odnosno, kao visina graničnog okvira objekta podijeljena sa visinom cijele slike. U slučaju sa slike 21 vrijednosti parametara širine i visine računaju se kao: $w = \frac{517}{1600} \approx 0.32$ i $h = \frac{530}{1000} \approx 0.53$. Dakle, ono što je važno primijetiti je da su x i y parametri izraženi relativno u odnosu na ćeliju na koju se odnose, a ne na cijelu sliku, dok su w i h izraženi u odnosu na cijelu sliku, a parametar pouzdanosti se računa prema izrazu 6.1.

Osim toga, svaka ćelija $S \times S$ rešetke predviđa i C uvjetnih vjerojatnosti klasa, koje se određuju prema izrazu 6.2:

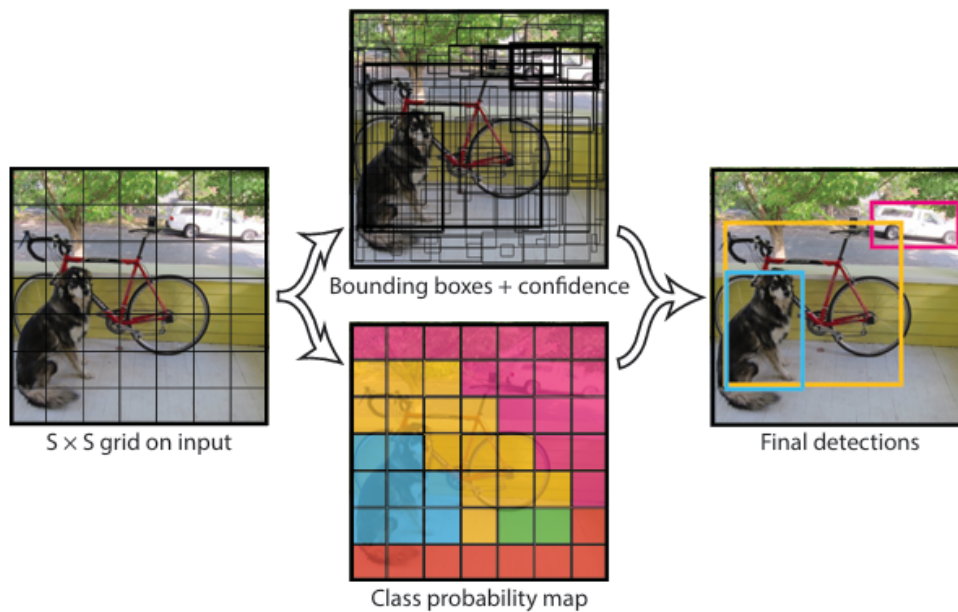
$$Pr(Class_i|Objekt) \quad (6.2)$$

Ove vjerojatnosti određuju ćelije koje sadrže objekt [51][52][53]. Ograničenje YOLO

algoritma je da jedna ćelija može detektirati samo jedan objekt. No, ako želimo detektirati više objekata za rešetku $S \times S$ može se uzeti da S bude nešto veći te da rešetka bude sitnija i samim time da može detektirati više objekata (npr. za $S = 3$, $S \times S = 9$, dakle ova mreža može detektirati najviše 9 objekata, no ako za S uzmemo vrijednost npr. 12 tada se naša rešetka sastoji od 144 ćelije te može detektirati toliko objekata) [51][53].

Mjera pouzdanosti za pojedinu klasu računa se prema izrazu 6.3:

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth} \quad (6.3)$$



Slika 22: Vizualni prikaz detekcije klasa po ćelijama (Izvor: [51])

Izraz 6.3 nam daje produkt vjerojatnosti pojedine klase u nekoj ćeliji ($Pr(Class_i)$) i kvalitete graničnog okvira (IOU_{pred}^{truth}) [51].

Nadalje, oznaka pojedine ćelije na slici biti će oblika kao što je prikazano u izrazu 6.4:

$$Oznaka_{\text{ćelija}} = [c_1, c_2, \dots, c_{20}, p_c, x, y, h, w], \quad (6.4)$$

gdje c vrijednosti predstavljaju sve moguće klase unutar PASCAL VOC skupa podataka, p_c predstavlja vjerojatnost nalazi li se objekt unutar okvira te vrijednosti x, y, h, w predstavljaju koordinate graničnog okvira za taj objekt [53]. Predikcije pojedinih ćelija izgledat će vrlo slično kao i oznake uz dodatak da će postojati dva skupa vjerojatnosti objekata i njegovih graničnih okvira (s ciljem kako bi se oni specijalizirali da npr. jedan daje više, a drugi šire granične okvire), vidi izraz 6.5:

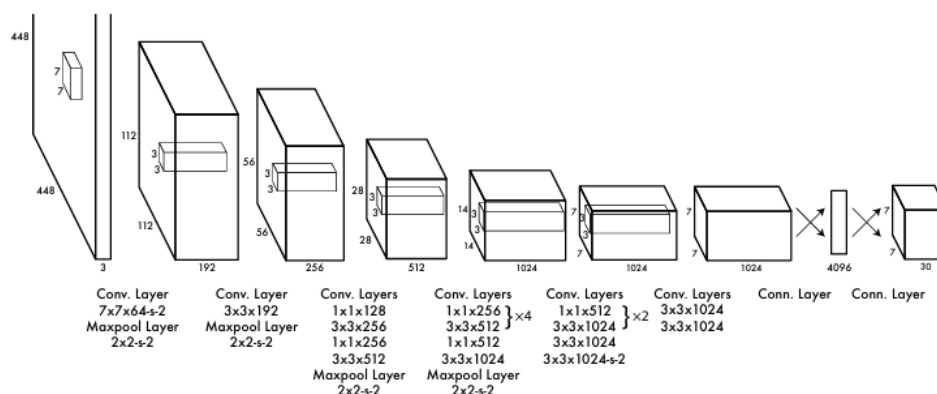
$$Predikcija_{\text{ćelija}} = [c_1, c_2, \dots, c_{20}, p_{c1}, x, y, h, w, p_{c2}, x, y, h, w] \quad (6.5)$$

Budući da se prethodno napisane vrijednosti odnose na pojedinačne ćelije unutar rešetke, izlaz cijele slike sastoji se od $S \times S \times (B * 5 + C)$ tenzora gdje je S veličina rešetke, B broj graničnih okvira koje svaka ćelija generira, u ovom slučaju 2 (na slici 21. prikazana je mreža sastavljena od 7×7 , tj. 49 ćelija, gdje kao što je rečeno svaka generira $B = 2$ granična okvira što znači da će u ovom slučaju biti generirano 98 graničnih okvira, vidi srednji gornji dio na slici 21), 5 označava broj ranije spomenutih parametara kojima se definira granični okvir objekta te je C broj mogućih klasa, u ovom slučaju 20. Izlaz se dodatno mogao zapisati kao i $S \times S \times 30$ tenzora. U slučaju 3×3 rešetke sa slike 16. tada bi oblik bio $3 \times 3 \times 30$ tenzora, tj. 270 tenzora ukupno.

6.1.2. Arhitektura YOLO algoritma

Ovaj model implementira se kao CNN te se procjenjuje na PASCAL VOC skupu podataka. Početni konvolucijski slojevi zaduženi su za izvlačenje značajki iz slike dok su potpuno povezani slojevi zaduženi za predikciju izlazne pouzdanosti i koordinate graničnog okvira [51].

Arhitektura je inspirirana *GoogLeNet*-ovim modelom za klasifikaciju slika, a sastoji se od 24 konvolucijska i 2 potpuno povezana sloja. "Umjesto početnih modula koje koristi GoogLeNet, jednostavno koristimo 1×1 redukcijske slojeve nakon kojih slijede 3×3 konvolucijski slojevi, slično Lin et al [51]." Cijelu mrežu moguće je vidjeti na slici 23 (dolje).



Slika 23: Prikaz arhitekture YOLOv1 algoritma (Izvor: [51])

Završni sloj kao izlaz daje predikciju izlazne pouzdanosti i koordinate graničnog okvira kao što je to već spomenuto ranije. Visina i širina graničnog okvira normalizira se u odnosu na sliku tako da se njihove vrijednosti nalaze između 0 i 1 [51]. Ovo omogućuje konzistentnost pri izlaznim rezultatima čak i ukoliko bi došlo do poboljšanja ili smanjenja kvalitete slike u rezoluciji [53]. Izlazni sloj koristi linearnu aktivacijsku funkciju dok svi ostali slojevi koriste *leaky ReLU* aktivacijsku funkciju koju je moguće vidjeti u nastavku, formula 6.6 (x predstavlja ulaz, a $f(x)$ aktivacijsku funkciju):

$$f(x) = \begin{cases} x & x > 0 \\ 0.1x & x \leq 0 \end{cases} \quad (6.6)$$

6.1.3. Funkcija gubitka

Funkcija gubitka (*engl. loss function*) ima ulogu uspoređivanja koliko se trenutni predikcijski okviri razlikuju od stvarnih te na osnovu tih podataka optimizirati proces učenja i smanjiti gubitke kroz iteracije učenja, tj. epohe. Funkcija gubitka računa se samo za okvire koji imaju najbolji *IOU*, u našem modelu (YOLOv1) to je bolji od dva okvira koja nudi svaka ćelija. Generiranje više graničnih okvira vodi do specijalizacije istih za pojedine oblike, npr. jedan bi se mogao specijalizirati za "vertikalne", a drugi za "horizontalne" okvire [53]. Koristi se formula za sumu kvadrata pogrešaka koja je izmijenjena i prilagođena našem cilju (vidi dio 6.8 u funkciji gubitka niže). Neke značajke formule za sumu kvadrata pogrešaka, a koje nisu idealne u našem slučaju, su da inicijalna formula: (1) jednako tretira lokalizacijske pogreške kao i klasifikacijske; (2) većina ćelija na bilo kojoj slici ne sadrži objekt te takve ćelije nadjačavaju one koje ih sadrže; (3) jednakom jačinom se ocjenjuju, tj. kažnjavaju greške u velikih i malih graničnih okvira.

Funkcija gubitka sastoji se od više dijelova (6.7, 6.8, 6.9, 6.10, 6.11), a nalazi se u nastavku:

$$\lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \quad (6.7)$$

$$+ \lambda_{coord} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \quad (6.8)$$

$$+ \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (6.9)$$

$$+ \lambda_{noobj} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (6.10)$$

$$+ \sum_{i=1}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (6.11)$$

Da bi se adresirale i optimizirale značajke (1) i (2) koriste se dva parametra: λ_{coord} i λ_{noobj} u svrhu da bi se pojačao gubitak predikcija koordinata graničnog okvira, pomoću prvog parametra postavljenog na vrijednost $\lambda_{coord} = 5$ te da bi se smanjio gubitak koeficijenta pouzdanosti za one ćelije koje ne sadrže objekte na način da se drugi parametar postavi na vrijednost $\lambda_{noobj} = 0.5$ [51].

Prije nastavka elaboracije funkcije gubitka dobro je spomenuti i što označavaju ostale varijable te funkcije. S^2 predstavlja ukupan broj ćelija u mreži slike, B je broj graničnih okvira koji se predviđaju za svaku ćeliju, tj. u YOLOv1 $B = 2$, x_i i y_i predstavljaju predikciju središta pojedinih graničnih okvira, dok \hat{x}_i i \hat{y}_i predstavljaju stvarne vrijednosti za te objekte. Analogno slijedi i za w_i i h_i koje predstavljaju predikcije širine i visine za odgovarajuće granične okvire, dok su \hat{w}_i i \hat{h}_i njihove stvarne vrijednosti. Karakteristične funkcije $\mathbb{1}_i^{obj}$ i $\mathbb{1}_i^{noobj}$ služe kao maska te $\mathbb{1}_i^{obj}$ poprima vrijednost 1 ukoliko se u pojedinoj ćeliji nalazi objekt, a u suprotnom je jednaka

0, dok obrnuto vrijedni za funkciju $\mathbb{1}_i^{noobj}$. Ukoliko se objekt ne nalazi u dotičnoj ćeliji tada je $\mathbb{1}_i^{noobj}$ jednaka 1, u suprotnom je jednaka 0. Varijable C_i i \hat{C}_i predstavljaju redom, predikciju pouzdanosti nalazi li se u nekoj ćeliji objekt ili ne (vrijednost iz raspona $[0, 1]$) te podatak o tome je li u toj ćeliji zaista objekt ili ne (element iz skupa $\{0, 1\}$). Na posljeticu, $p_i(c)$ vjerojatnost za pojedinu klasu u i -toj ćeliji, dok $\hat{p}_i(c)$ predstavlja podatak tome nalazi li se pojedina klasa u i -toj ćeliji ili ne (vrijednosti 1 ili 0).

Razumljivo je kako bi se greške jednake apsolutne veličine u manjih okvirima trebale "kažnjavati" više nego u većim okvirima. Dakle, da bismo poboljšali problem koji se javlja vezano uz značajku (3), barem djelomično, radi se predikcija korijena visine i širine graničnog okvira umjesto da se radi predikcija visine i širine direktno (vidi dio 6.9 u funkciji gubitka niže) [51].

Dijelovi 6.9 i 6.10 se nadopunjuju te donose informaciju o gubitku uslijed određivanja pouzdanosti za svaki granični okvir. Gubitak se računa analogno kao i u dijelu 6.11 u nastavku te se računa za 2 slučaja: (1) kada se objekt nalazi u dotičnoj ćeliji i kada postoji odgovoran granični okvir (6.9); (2) kada se objekt ne nalazi u dotičnoj ćeliji i kada postoji odgovoran granični okvir (6.10) [51].

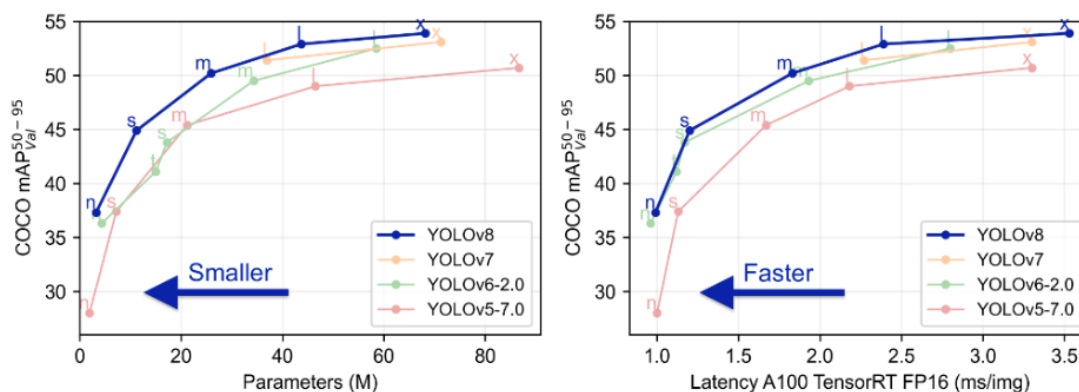
U zadnjem dijelu funkcije gubitka (6.11) računa se gubitak vjerojatnosti klasa. Kao što je spomenuto ranije, računa se razlika vjerojatnosti modela za pojedinu klasu i podatka o stvarnoj klasi koja se nalazi u konkretnoj ćeliji. Dok su vrijednosti $p_i(c)$ iz intervala $[0, 1]$ one također mogu biti nejednake 0 za više od jedne klase po ćeliji (najčešće, u praksi, upravo to i jest slučaj), dok su vrijednosti $\hat{p}_i(c)$ uvijek jednake 0 za svaki $c \in classes$ osim za jednu koja je jednaka 1 (Vidljivo je kako su ove vrijednosti tipa *bool*). Zaključno, posljednji (6.11) dio funkcije gubitka nam daje informaciju o tome koliki je gubitak vjerojatnosti klase, no samo kada se objekt nalazi u toj ćeliji, u suprotnom je ova vrijednost jednaka 0 [51].

7. Detekcija tumora mozga primjenom YOLOv8 algoritma

Ovo je poglavlje gdje se nalazi praktični dio rada. Objasnjene su razlike između YOLOv1 i YOLOv8 verzije, objašnjen je postupak anotiranja i pripremanja skupa podataka kako bi bili podobni za korištenje u programskom kodu te su objašnjeni rezultati treniranja modela. Skup podataka na kojemu se model trenira i testira preuzet je sa Kagglea [62].

7.1. Razlike YOLOv1 i YOLOv8 algoritma

Prva od razlika koje valja spomenuti jest skupu podataka na kojima su trenirane ove dvije verzije YOLO algoritma. Dok je YOLOv1, kao što je i opisano u dosadašnjim poglavljima treniran na PASCAL VOC skupu podataka, YOLOv8 treniran je na Microsoft-ovom COCO skupu podataka koji se sastoji od preko 330 000 slika i 80 klasa [63]. No, u praktičnom dijelu ovog rada model će biti treniran na vlastitom skupu podataka. Također, kod YOLOv8 algoritma granični se okviri određuju pomoću tri sidrene kutije (*engl. Anchor box algorithm*) za razliku od $B = 2$ granična okvira po ćeliji kod YOLOv1 algoritma [58]. Osim toga postoje i određene razlike u funkciji gubitka, no ono što je najbitnije i što je bila primarna motivacija za odabir YOLOv8 verzije za praktičan dio završnog rada jest poboljšana preciznost i optimizirana brzina samog algoritma. Na slici 24 u nastavku prikazan je odnos brzine i točnosti YOLOv8 algoritma u usporedbi sa nekim od ranijih verzija.



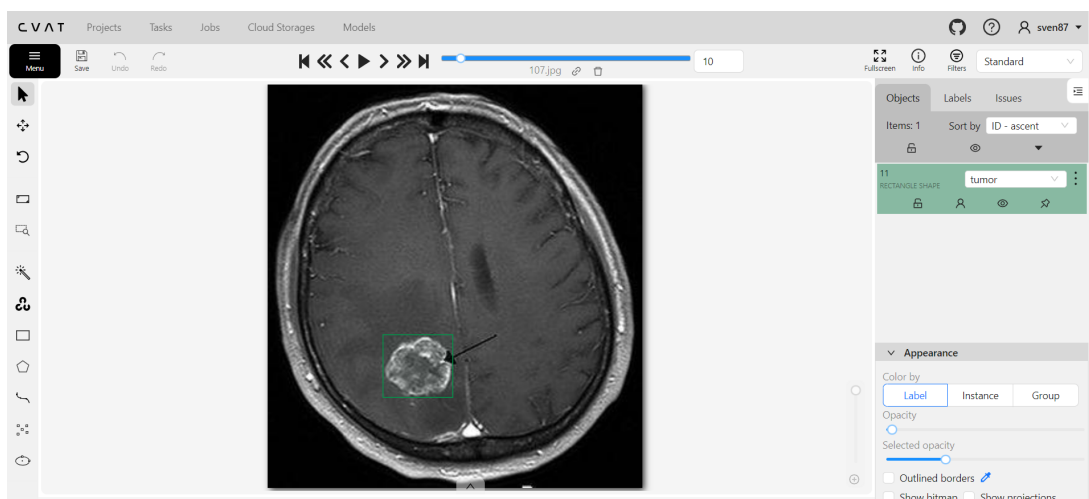
Slika 24: Usporedba YOLOv8 algoritma i ranijih verzija (Izvor: [64])

Na lijevom grafu prikazan je odnos broja parametara i srednje vrijednosti prosječnih preciznosti (mAP), dok je na desnom grafu prikazan odnos vremenske latencije (za 100 tenzora) u milisekundama po slici i srednje vrijednosti prosječnih preciznosti (mAP). Vidljivo je kako je YOLOv8 u globalu memorijski manje zahtjevan te brži od svojih prethodnika uz višu razinu preciznosti od svih.

7.2. Priprema programskog okruženja

7.2.1. CVAT

Prije same provedbe programskog koda korišten je alat *CVAT* za anotiranje skupa podataka na kojemu će model biti testiran. *CVAT* je besplatan online alat sa kojim je anotirano 129 tumor-pozitivnih slika te uvezeno i preostalih 98 tumor-negativnih slika radi potrebnog formata u kojemu će slike biti izvezene i kako bi se mogle jednostavno koristiti u YOLOv8 algoritmu kasnije u kodu. Uz to, u alatu je kreirana je i jedna klasa, klasa tumora kojoj pripadaju svi anotirani objekti na slikama. Granični okvir tumora pri procesu anotiranja označen je zelenom bojom. Primjer rada i anotacije tumora na MR slikama nalazi se u nastavku, slici 25.



Slika 25: CVAT anotiranje tumora (Izvor: izrada autora)

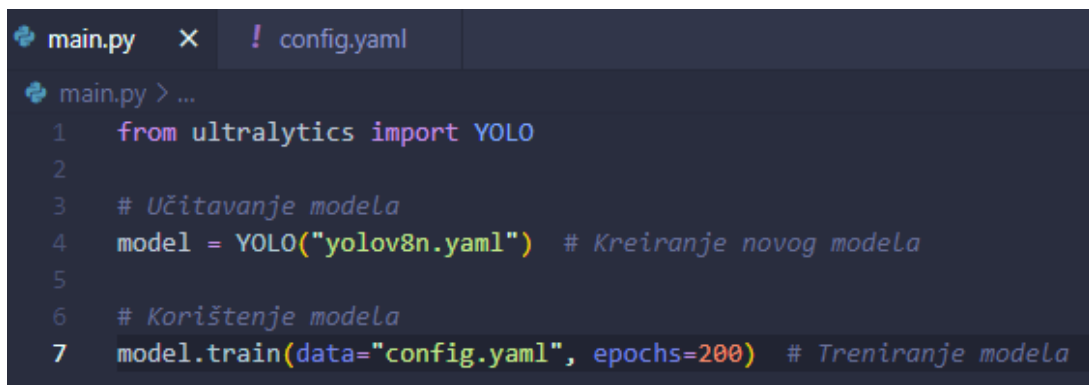
Nakon preuzimanja slika iz *CVAT* alata uz same datoteke slika kreira se i *.txt* datoteka koja sadrži 5 parametra: brojanu oznaku klase te parametre: x , y , w i h . Ti se podatci koriste kasnije kao temeljna istina pri učenju modela o tome gdje se uistinu nalazi tumor na pojedinoj MR slici, tj. nalazi li se uopće.

7.2.2. Python

Za izradu i provedbu ovog modela korišten je Python kao programski jezik zbog dostupnosti potrebne biblioteke za jednostavnu integraciju YOLOv8 algoritma. Prvi korak u radu, nakon preuzimanja *Python* alata, bilo je instaliranje paketa *ultralytics* u kojemu je integriran kod za YOLOv8 algoritam. Programski kod je izuzetno jednostavan i sastoji se od svega 3 linije (slika 26, dolje). U prvoj liniji se iz biblioteke *ultralytics* uveze funkcija *YOLO*. U drugoj liniji se model učita, "yolov8n" znači kako se koristi nano verzija zbog svoje male veličine i brzine. Inače postoji 5 različitih inačica YOLOv8 algoritma glede veličine skupova parametara, redom u rastućem poretku one su: nano(n), small(s), medium(m), large(l), i extra large(x) [65]. Nakon što je model učitao, trenira se na prethodno spomenutom, vlastitom skupu podataka. Dio koda: `data="config.yaml"` govori kako se do podataka dolazi putem *config.yaml* datoteke. Ta datoteka sadrži putanju do skupa podataka koji se koristi te nazive svih klasa koje se javljaju u skupu

podataka (u našem slučaju to je samo jedna klasa, klasa tumora).

Konačno, broj epoha (*engl. epochs*) označava u koliko iteracija će se provoditi učenje ovog modela. U primjeru sa slike 26 taj je broj jednak 200. Bitno je primijetiti kako će model biti tim uspješniji što je njegovo učenje duže i cjelovitije, dakle veći broj epoha. Drugim riječima njegovo se učenje, dakle i preciznost u prosjeku povećava iz epohe u epohu do određene granice kada to više nije slučaj. Više govora o tome biti će u narednom poglavlju u analizi rezultata dobivenih treniranjem ovog modela. Osim broja epoha, drugi faktor koji utječe na uspješnost treniranja modela jest i veličina skupa podataka te njegova adekvatna priprema. U ovom slučaju priprema podataka je nakon anotiranja tumora podrazumijevala podjelu cjelokupnih podataka na skup za treniranje i testiranje. U radu model je testiran 3 puta: prvi put testiran je na način da je skup za testiranje bio jednak skupu za treniranje kako bi se vidjelo jesu li ovi podatci adekvatno pripremljeni te može li model učiti na njima. Broj epoha u prvom treniranju bio je jednak 100. Zaključak je kako su podatci adekvatno pripremljeni, no čini se kako nakon 100 epoha program ima još prostora za učenje. Tako je došlo do druge verzije čiji je prikaz na slici 26, jedina razlika u odnosu na prvu verziju jest broj epoha koji je sada postavljen na 200. Zaključak je bio kako su podatci i rezultati vrlo dobri te da je vrijeme za podijeliti model na dva skupa.



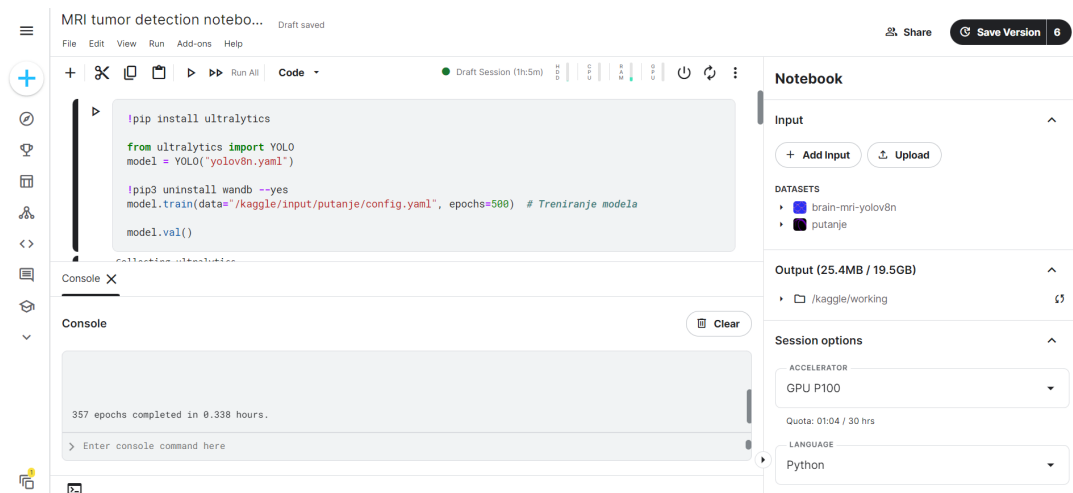
```
main.py X ! config.yaml
main.py > ...
1 from ultralytics import YOLO
2
3 # Učitavanje modela
4 model = YOLO("yolov8n.yaml") # Kreiranje novog modela
5
6 # Korištenje modela
7 model.train(data="config.yaml", epochs=200) # Treniranje modela
```

Slika 26: Prikaz Python koda (Izvor: izrada autora)

Ukupan skup podataka je podijeljen na skup za treniranje i skup za testiranje, kao što je spominjano ranije, u omjeru 80:20 u korist skupa za treniranje. Osim toga, u trećoj verziji korišteno je i drugo okruženje. Još uvijek je korišten Python, no korišten je Kaggle za provedbu koda zbog akceleratora koje isti nudi. Konkretno, korišten je akcelerator *GPU P100* koji je omogućio da bude zadan veći broj epoha zbog brzine izvođenja koda. Broj epoha postavljen je na 500, slika 27, no vidjet će se kasnije kako je model prekinuo izvršavanje nakon 357. epohe iz razloga što su se trendovi pada krivulja gubitaka i trendovi rasta krivulja preostalih metrika zaustavili te su postali približno jednaki pravcu $y = konst.$, slika 31.

7.2.3. Prikaz i analiza grafova YOLOv8 modela

Kao rezultat pokretanja *YOLO* funkcije iz *ultralytics* biblioteke u Python kodu kreira se nova datoteka u mapi gdje se nalaze i podatci na računalu gdje je pokrenut kod. Štoviše, nova datoteka kreira se svakim novim pokretanjem koda. Ova datoteka sadrži brojne korisne



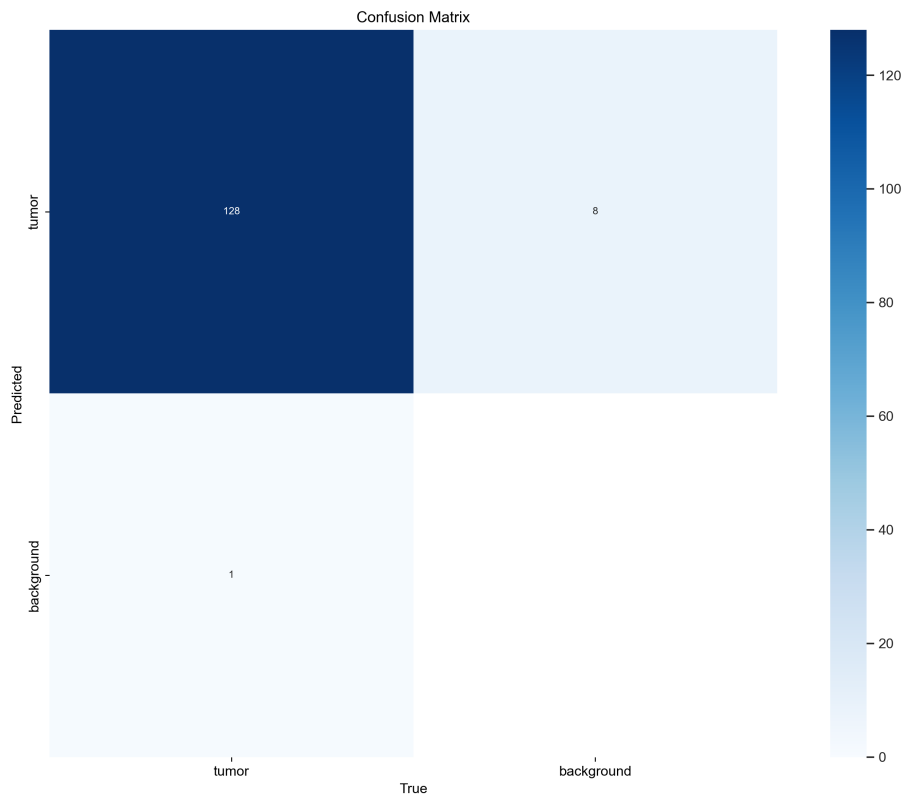
Slika 27: Prikaz Python koda i okruženja u Kaggleu (Izvor: izrada autora)

podatke, slike i grafikone za razumijevanje i evaluiranje procesa treniranja te informacije o tome koliko je dobro program detektirao objekte na slikama koje su mu predočene kada je bilo vrijeme validacije spomenutog postupka treniranja. U ovom slučaju, koliko je dobro anotirao tumore na MR slikama mozga.

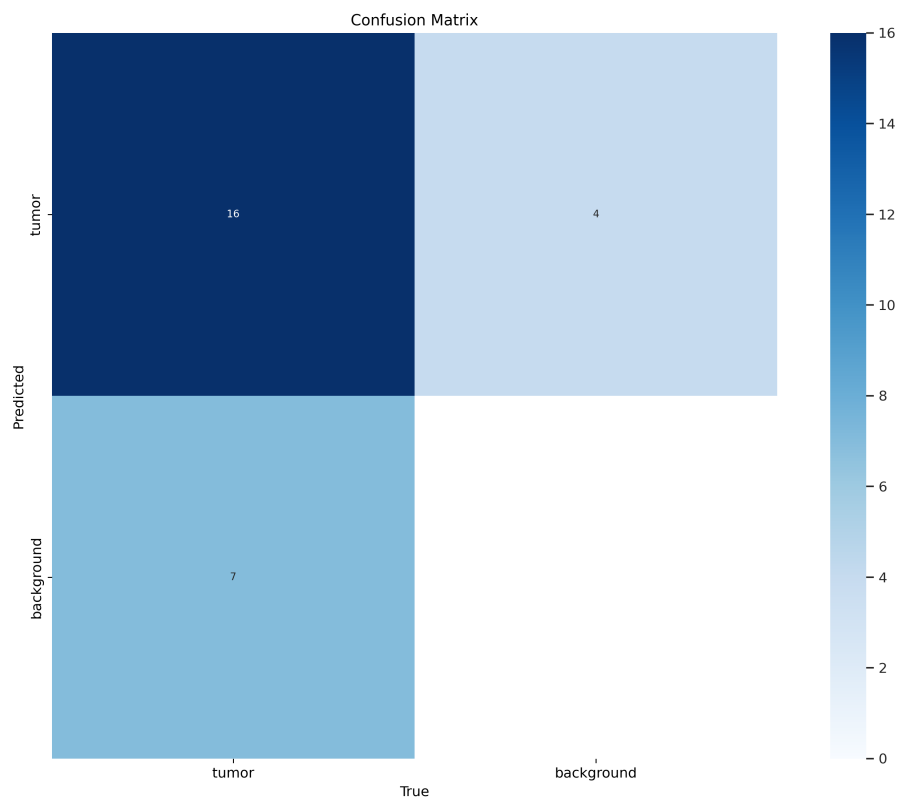
Budući da je model treniran na samo jednoj klasi, klasi tumora, matrica zabune vrlo je jednostavna. Prvo će biti analizirana 2. verziju modela (slika 28). Vidljivo je kako je velika većina tumora točno klasificirana, tj. 128 od ukupno 129 tumora model je uspješno prepoznao kao tumore, takva detekcija podataka zove se *true positive*. Osim toga, vidljivo je i kako je model 8 puta detektirao tumor na slikama gdje ga nije niti bilo, takav primjer detekcije zove se *false positive*. Nadalje, vidljivo je i kako model nije uspio detektirati jedan tumor koji je trebao, ovakav slučaj detekcije, tj. njenog izostanka zove se *false negative*. U okviru posla za koji je model treniran najopasniji su upravo ovi potonji slučajevi jer ukoliko model ne detektira tumor postoji mogućnost da ga niti medicinski stručnjak neće detektirati te da će tumor kod dotičnog pacijenta metastazirati i da bi se situacija mogla neopravdano zakomplicirati. U slučaju *false positive* detekcije model će skrenuti pozornost medicinskog stručnjaka na te slike MR mozga, no tada će on zaključiti da se radi o pogrešnoj detekciji i šteta po pacijenta neće biti učinjena.

Analogni opis vrijedi za matricu zabune 3. verzije modela (slika 29). Jedina razlika jest u tome da su podatci u toj matrici nešto lošiji zbog toga što je gore opisan model treniran i testiran na jednakim podacima, dok je ovaj testiran na novim, nikad viđenim podacima što dolazi do izražaja zbog relativno malog skupa ukupnog, pa tako i testnog, skupa podataka. Ovo vrijedi i za sve datoteke u nastavku pa neće biti detaljno objašnjavano.

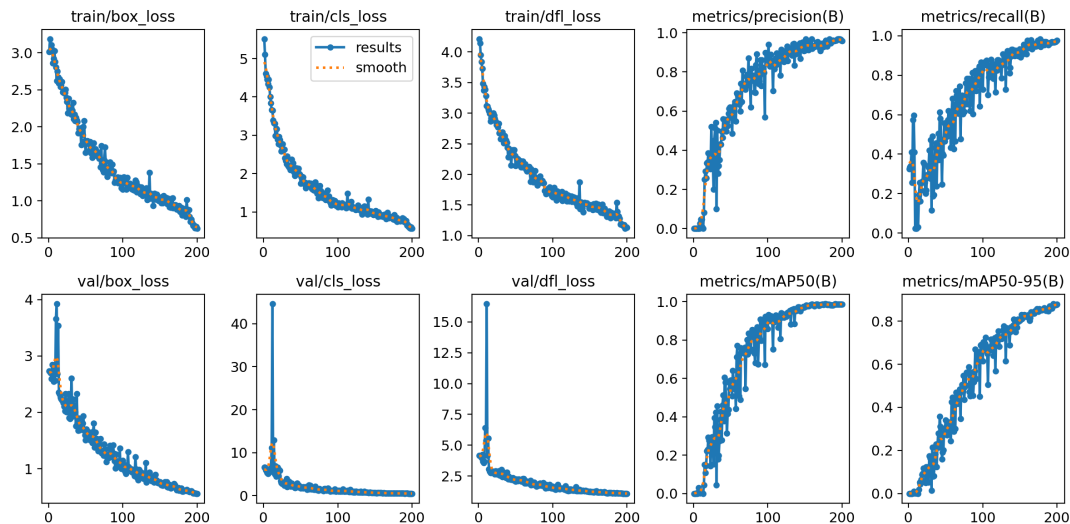
Druga datoteka koja će biti analizirana bit će *.png* datoteka naziva "results". Na slici 30. u nastavku prikazano je 10 različitih grafova za drugu verziju modela. Preciznost graničnih okvira nakon 200 epoha bila je jednaka 0.95895. Ovaj podatak očitao je iz excel tablice sa vrijednostima svih točaka grafova sa slike u nastavku. Za treću verziju modela grafovi se nalaze na slici 31. Ova verzija trenirana je na 357 epoha učenja te joj je prosječna preciznost zadnjih 10 epoha 0.83116 dok je zadnja najlošija od tih 10 te nije reprezentativna sa svojom preciznošću od tek 0.69012.



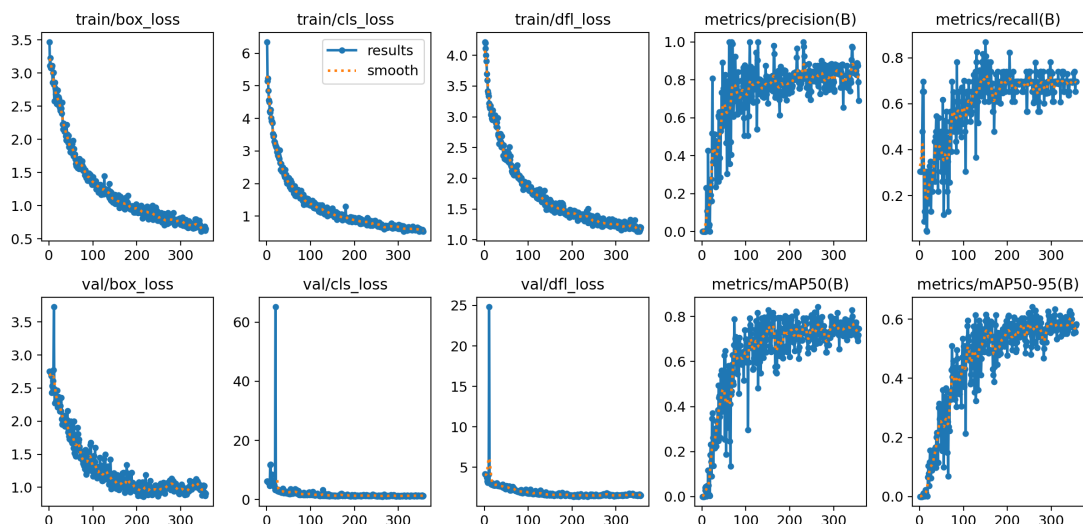
Slika 28: Matrica zabune za 2. verziju modela - (Izvor: izrada autora)



Slika 29: Matrica zabune za 3. verziju modela - (Izvor: izrada autora)



Slika 30: Prikaz datoteke "results" - 200 epoha (2. verzija modela) (Izvor: izrada autora)

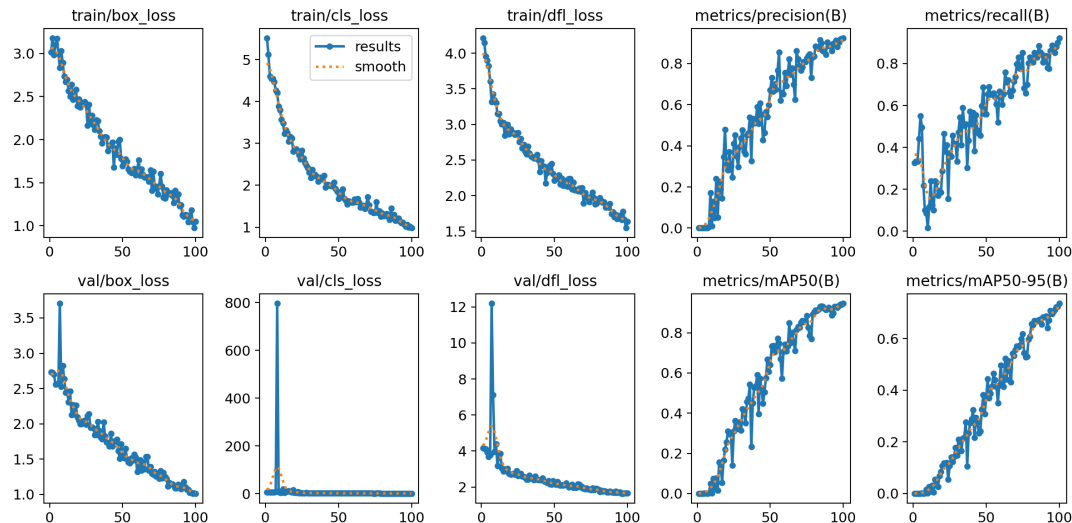


Slika 31: Prikaz datoteke "results" - 357 epoha (3. verzija modela) (Izvor: izrada autora)

Generalno, vidljivo je kako se prvih 6 grafova na lijevoj strani, koji se odnose na segmente funkcije gubitka, kontinuirano smanjuju. To je dobar pokazatelj budući da je poželjno da gubitci u predikcijama modela budu minimalni. Također, vidljivo je kako se sva četiri grafa sa metrikama na desnoj strani slike kontinuirano povećavaju što je također željeni ishod.

Na osnovu ranije spomenute informacije možemo zaključiti kako model s vremenom uči. Ovo je vrlo važno jer u suprotnom ne bismo mogli postići željeni ishod: naučiti model pomoću YOLO algoritma da prepoznaje tumore na slikama magnetske rezonance mozga. Nadalje, ako detaljnije gledamo grafove sa slika 30. i 31. može se primijetiti kako se kod svakog pojedinog grafa pripadajući koeficijent smjera krivulja rasta, odnosno pada postepeno približava vrijednosti nula. Upravo to je posljedica koju bismo i očekivali u dobro koncipiranog i provedenog modela učenja. Naime, ovo se događa iz razloga što model na početku ima štošta za naučiti i svaka nova epoha učenja znatno doprinosi cjelokupnoj performansi modela dok je kasnije taj učinak sve manji i manji iz razloga što se model već nalazi na razumnoj razini zrelosti te su

mu gubitci razmjerno mali, odnosno performanse razmjerno dobre. Upravo približavanje koeficijenta smjera grafova nuli dobar je pokazatelj da se model bliži vrhuncu svojih mogućnosti učenja. Prije nego što je ovaj algoritam pokrenut na 200 epoha, pokrenut je na 100. No, kako je bilo resursa i vremena za povećanje ovog broja, to je i učinjeno. Razlog iza te odluke nalazi se upravo u tome da pojedini grafovi na rezultatima od 100 epoha nisu pokazivali značajnije znakove usporavanja, tj. da je moguće program bolje naučiti željenoj svrsi prepoznavanja tumora. Na slici 32. (dolje) nalaze se rezultati za 100 epoha YOLOv8 algoritma. Zbog mogućnosti koje Kaggle pruža kasnije je bilo moguće broj epoha dodatno povećati što je i učinjeno.



Slika 32: Prikaz datoteke "results" - 100 epoha (1. verzija modela) (Izvor: izrada autora)

7.2.4. Prikaz i analiza rezultata treniranja YOLOv8 modela

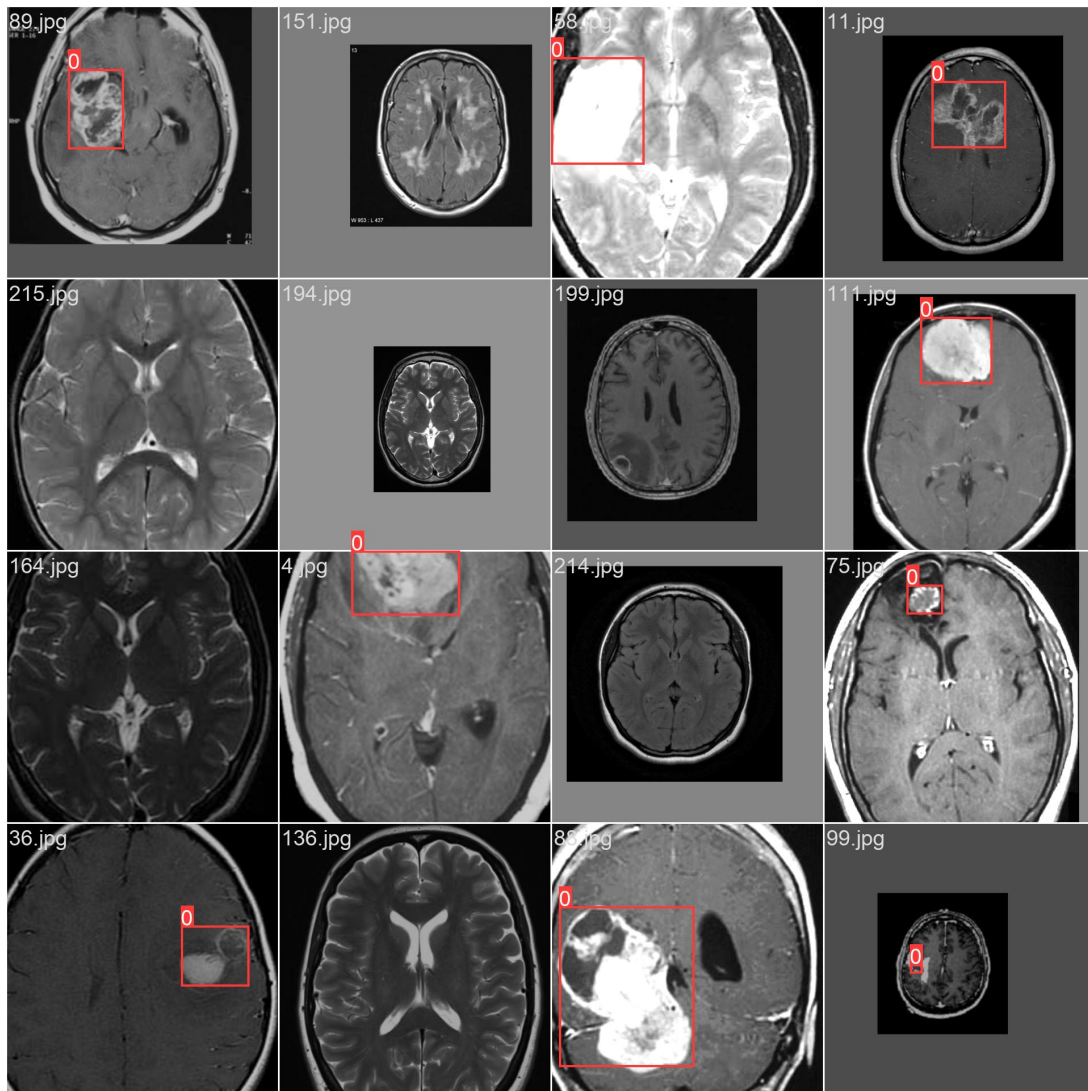
Prvo će biti prikazani i elaborirani rezultati druge verzije modela, kasnije će biti uspoređeni sa prvom verzijom modela te će na poslijetku biti priložene i slike rezultata treće verzije modela.

Pri treniranju, modelu su predočene slike na način da je 16 njih spojeno u jednu 4×4 sliku kao što je prikazano na slici 33. u nastavku.

Kao što je vidljivo, uz same slike na tumor-pozitivnim MR snimkama prikazani su i granični okviri koji su anotirani u CVAT programu kao što je opisano ranije. Dodatno, broj označava oznaku klase, u ovom slučaju postoji samo jedna klasa, klasa tumora i njoj je designiran broj 0. To su granični okviri s kojima se će uspoređivati granični okviri predikcija modela kako bi se utvrdilo koliko su predikcije modela točne te kako bi model mogao učiti na temelju tih povratnih informacija.

U nastavku ćemo pogledati rezultate predikcija ovog modela treniranog na 200 epoha te kako kotiraju u usporedbi sa stvarnim okvirima anotiranim u CVAT alatu. Na kraju bit će prikazana i sažeta usporedba modela treniranog na 100 epoha.

Slika 34, u nastavku, prikazuje jednu od slika koja je dana modelu da napravi predikciju graničnih okvira te da odredi pouzdanost za iste. Na slici 34. nalaze se stvarni granični okviri,

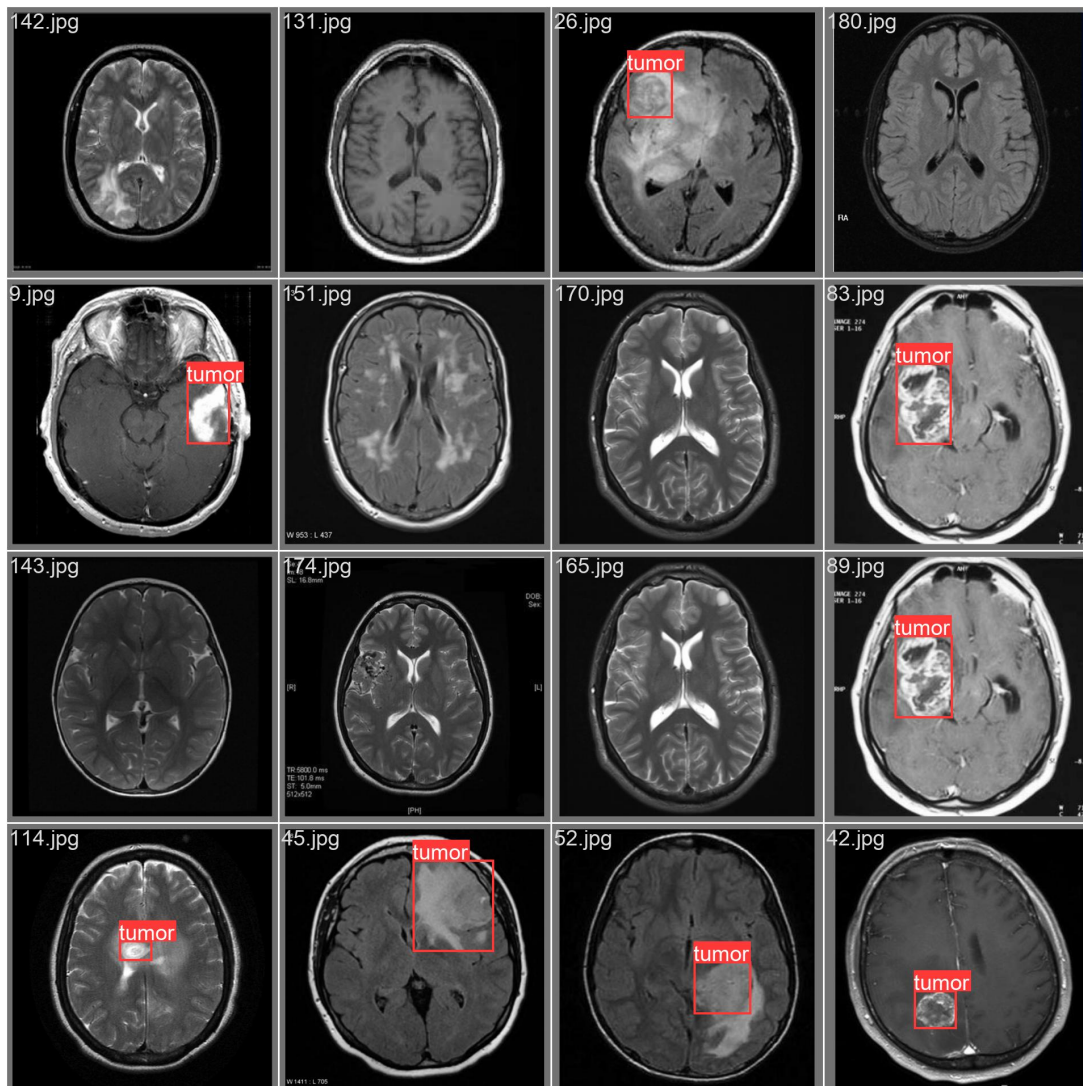


Slika 33: Prikaz slika za treniranje modela (Izvor: izrada autora)

tj. oni anotirani u CVAT alatu (koje model u trenutku donošenja predikcija nije vidio), dok se na slici 35. nalaze predikcije modela. Na ovaj način vrlo se lako može vizualno utvrditi radi li model "dobar posao" ili ne.

Vidljivo je kako je razlika između slike 34. i slike 35. minimalna, tj. jedina razlika je u tome da je model na slici 35. za jedan tumor prepoznao dva. Ovdje je također vrlo bitno vidjeti pouzdanosti modela za svaku predikciju. Vidjet ćemo kako jedino za taj granični okvir (koji nije trebao niti postojati) vrijednost parametra pouzdanosti odudara od ostalih. Naime, jedino za taj granični okvir vrijednost parametra pouzdanosti iznosi tek 0.3, dok ostale vrijednosti iznose redom 0.7 (jednom), 0.8 (dva puta), 0.9 (tri puta) te 1.0 (dva puta). Dakle, ukoliko bismo odredili da prag prikazivanja graničnih okvira bude jednak npr. 0.5 svi granični okviri bili bi točno prediktirani.

Osim što bi na taj način svaki tumor bio prepoznat i to sa relativno visokom vrijednosti parametra pouzdanosti, bitno je primijetiti i kako nije prepoznat niti jedan tumor od strane modela tamo gdje se on uistinu ne nalazi. Ova tvrdnja vrijedni ne samo na izabranoj slici već kroz



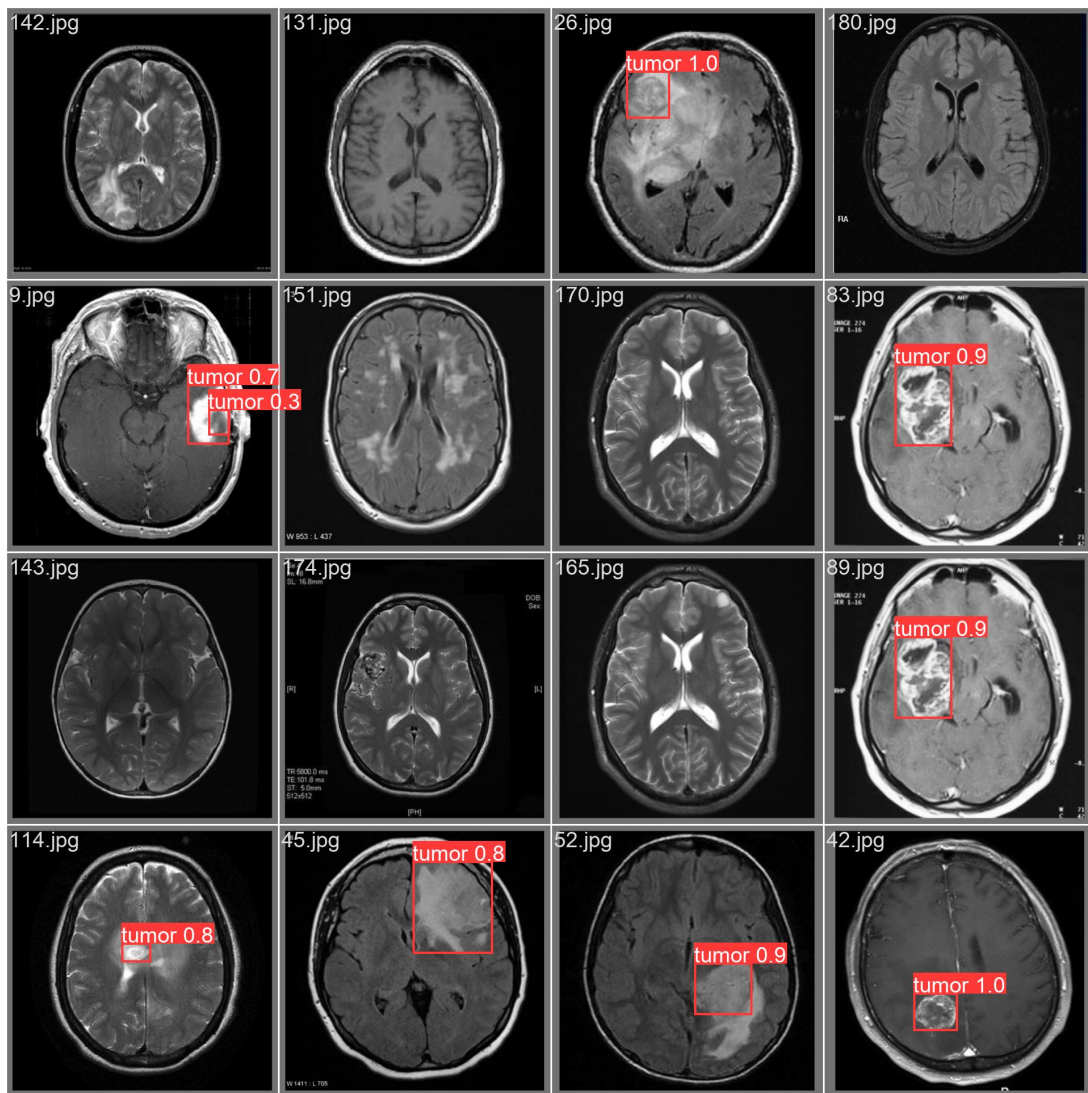
Slika 34: Prikaz stvarnih graničnih okvira (Izvor: izrada autora)

sve dostupne rezultate predikcija modela za koje postoje slike.

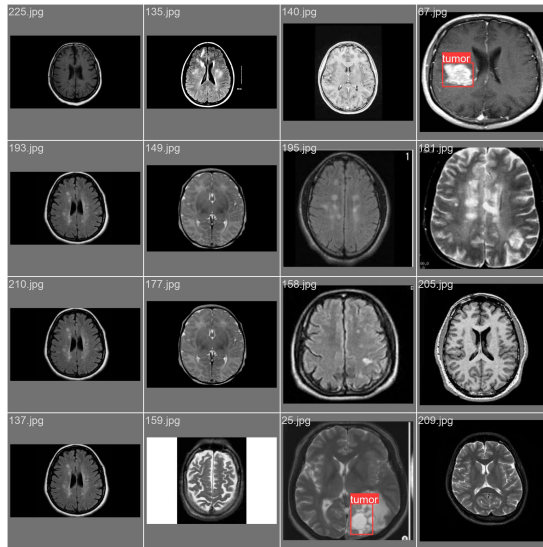
U nastavku, slika 36., prikazana je usporedba između stvarnih graničnih okvira te predikcija modela treniranog na 100 epoha i predikcija modela treniranog na 200 epoha. Razlike modela treniranih na 100 i na 200 epoha su vidljive. Model treniran na 200 epoha na istoj priloženoj slici ispravio je sve greške koje je model treniran na 100 epoha učinio te je sigurniji u svoje predikcije.

Zaključak ove usporedbe je kako je odluka o provođenju dodatnog treniranja modela na više od inicijalnih 100 epoha bila ispravna. Tj. dobro je da je model treniran na 200 epoha budući da pokazuje znatno bolje rezultate te da gotovo ne griješi u svojim predikcija (barem na ovom skupu podataka), osim na slučaju koji je spomenut ranije. Ipak i to je bila greška manjeg razmjera u medicinskom kontekstu jer je tumor svakako bio prisutan na tom području, ali se radilo o jednom, a ne dva tumora.

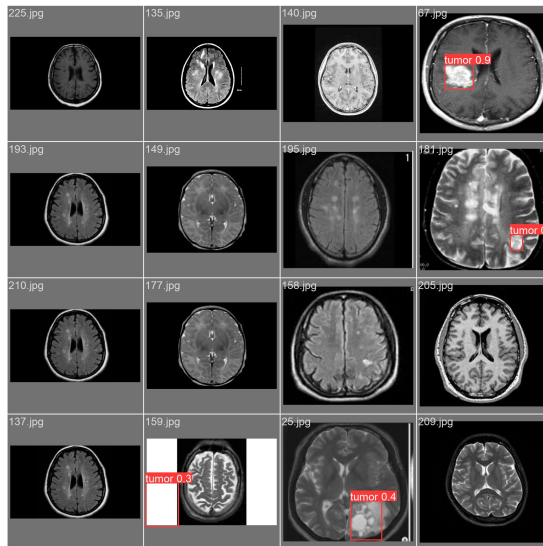
Na slici 37 nalaze se stvarni granični okviri testnih podataka 3. verzije modela, dok se na slici 38 nalaze predikcije istog modela.



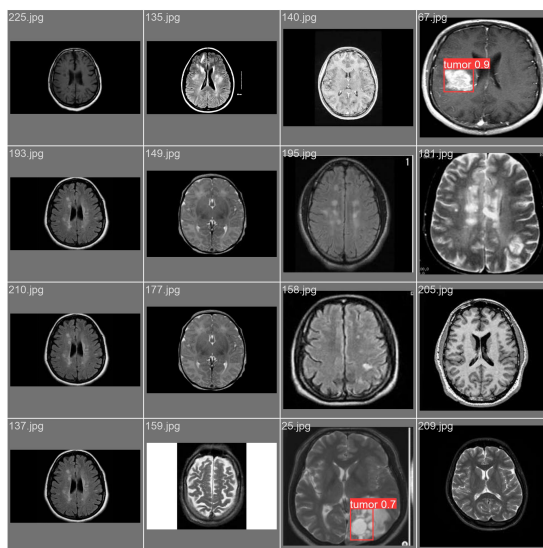
Slika 35: Prikaz predikcija graničnih okvira 2. verzije modela (Izvor: izrada autora)



Stvarni granični okviri

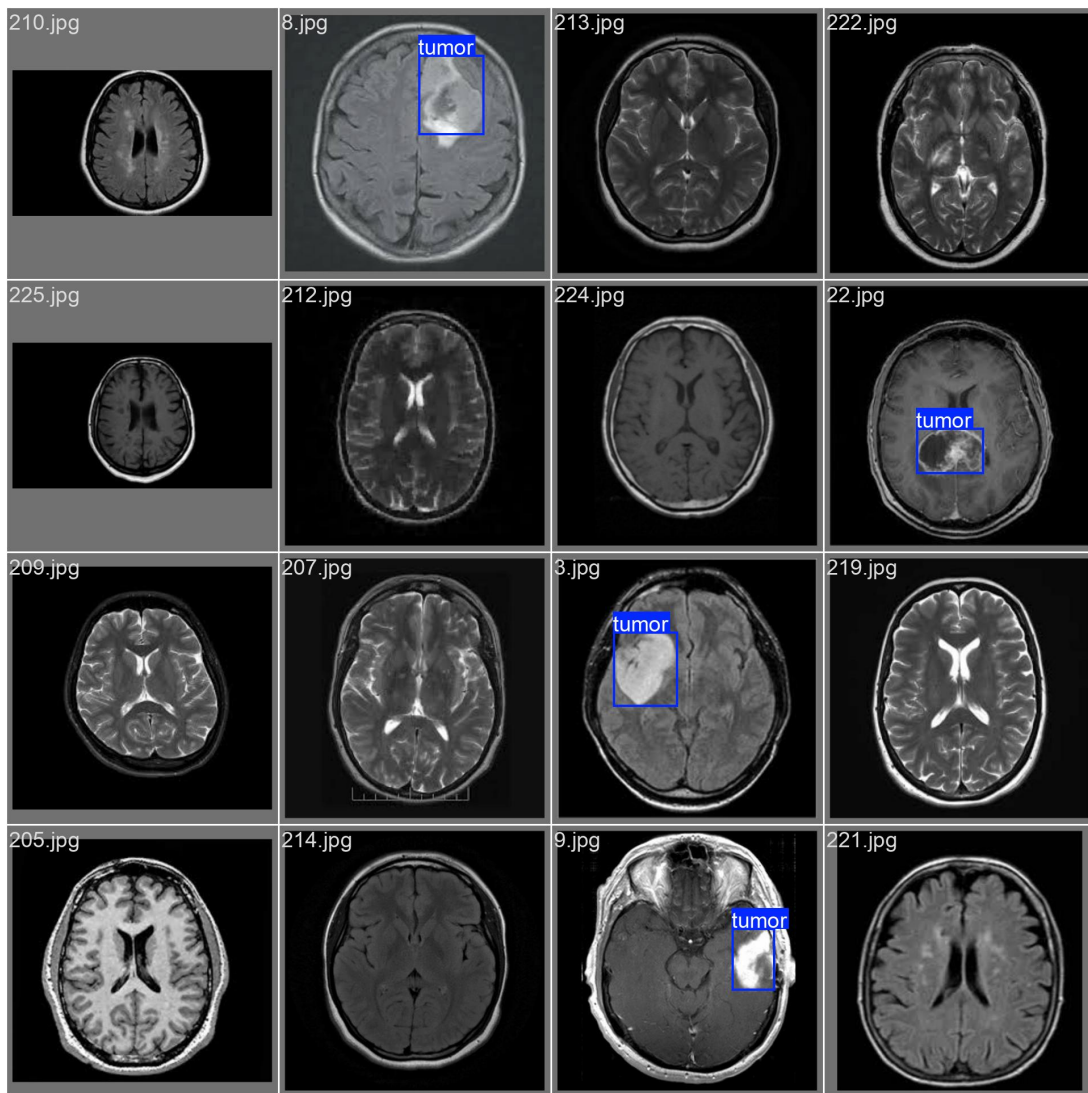


Predikcije modela treniranog na 100 epoha

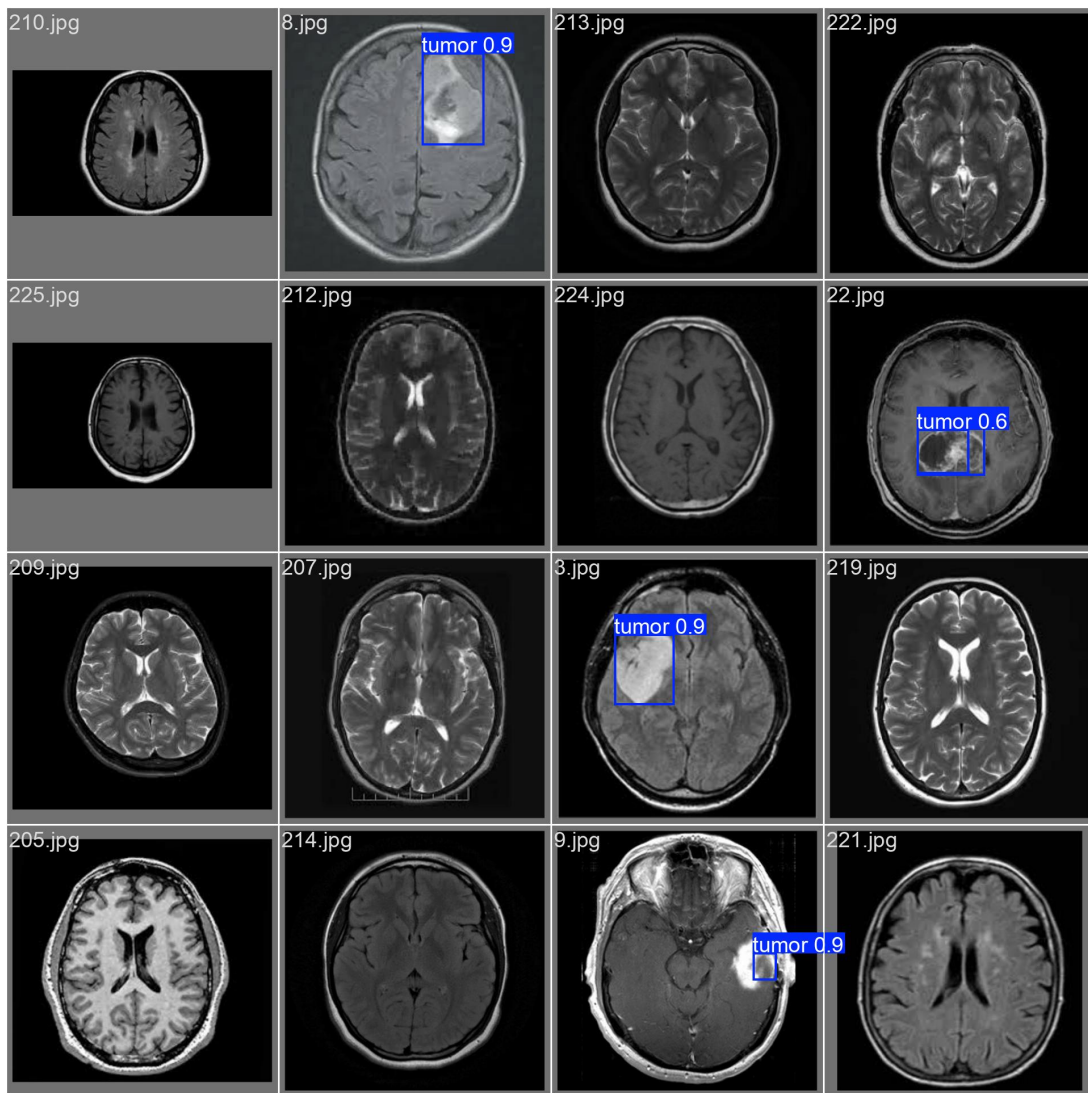


Predikcije modela treniranog na 200 epoha

Slika 36: Usporedba predikcija sa stvarnim graničnim okvirima (Izvor: izrada autora)



Slika 37: Prikaz stvarnih graničnih okvira (Izvor: izrada autora)



Slika 38: Prikaz predikcija graničnih okvira 3. verzije modela (Izvor: izrada autora)

8. Zaključak

Predmet ovog rada bila je primjena YOLO algoritma za detekciju tumora na slikama magnetske rezonance mozga. Prije same programske primjene detalji rada algoritama detaljno su opisani sa matematičkog stajališta. Za programsku provedbu algoritma korištena je platforma Kaggle i programski jezik Python te njegov paket ultralytics koji sadrži sam kod YOLO algoritma.

Budući da YOLO algoritam postoji u više verzija, važno je spomenuti kako je praktični, programski dio koristio YOLOv8n (YOLO verzija 8 nano) inačicu algoritma. Verzija 8 korištena je radi svoje brzine i pouzdanosti, dok je nano verzija korištena radi svoje brzine izvođenja, manje mjesta za pohranu te energetske efikasnosti.

Rezultat ovog završnog rada je uspješan iz razloga što su rezultati treniranja modela uspješni te u skladu sa očekivanjima visoke preciznosti za točno prepoznavanje tumora mozga i postotkom pouzdanosti od preko 83%. Model je inicijalno treniran na 100 epoha učenja, no procijenjeno je kako je taj broj premalen te da bi bilo optimalnije povećati isti na 200. To je i učinjeno te su se rezultati nakon 200 epoha učenja pokazali zadovoljavajućima te vrlo dobrima kao što je ranije i spomenuto. Zbog toga prelazi se na treću verziju modela kod koje je podijeljen skup podataka na skup za treniranje i skup za testiranje u omjeru 80 : 20, te je model u konačnici treniran na 357 epoha učenja. Upravo u ovoj verziji postignuta je gore spomenuta prosječna preciznost kroz posljednjih 10 epoha od preko 83%. Obzirom na mali ukupan skup podataka, kauzalno i na još manji testni skup podataka i proces anotiranja koji je vlastite izrade, rezultati su više nego zadovoljavajući.

U procesu pisanja ovog rada korišteni su razni alati i aplikacije, od Overleafa za pisanje LaTeX koda u kojemu je pisan sam rad, do grafičkih alata poput paint.net-a, Geogebre za izradu grafova funkcija, CVAT alata za anotiranje skupa podataka te već spomenutih Pythona i Kagglea za programsku podršku i izvedbu.

Popis literature

- [1] I. z. M. K. Hrvatska Enciklopedija. „tumor.” (), adresa: <https://www.enciklopedija.hr/clanak/tumor>. (proučavano: 28.03.2024.)
- [2] E. Britannica. „tumour.” (), adresa: <https://www.britannica.com/science/tumor>. (proučavano: 28.03.2024.)
- [3] C. C. Society. „Types of tumours.” (), adresa: <https://cancer.ca/en/cancer-information/what-is-cancer/types-of-tumours>. (proučavano: 28.03.2024.)
- [4] A. A. of Neurological Surgeons. „Brain Tumors.” (), adresa: <https://www.aans.org/patients/conditions-treatments/brain-tumors/>. (proučavano: 26.03.2024.)
- [5] H. zavod za javno zdravstvo. „Svjetski dan oboljelih od tumora mozga.” (), adresa: <https://www.hzjz.hr/aktualnosti/svjetski-dan-oboljelih-od-tumora-mozga/>. (proučavano: 28.03.2024.)
- [6] K. Versteeg, S. Blauwhoff-Buskermolen, L. Buffart i dr., „Higher Muscle Strength Is Associated with Prolonged Survival in Older Patients with Advanced Cancer,” *Oncologist*, sv. 23, br. 5, str. 580–585, 2018.
- [7] T. J. H. University, T. J. H. Hospital i J. H. H. System. „Brain Tumors and Brain Cancer.” (), adresa: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor>. (proučavano: 26.03.2024.)
- [8] M. Clinic. „Brain tumor.” (), adresa: <https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084>. (proučavano: 26.03.2024.)
- [9] V. Zdravlje. „TUMOR MOZGA - SIMPTOMI I LIJEČENJE.” (), adresa: <https://www.vasezdravlje.com/bolesti-i-stanja/tumor-mozga-simptomi-i-lijecenje>. (proučavano: 26.03.2024.)
- [10] I. M. Virtualna Ordinacija. „Rak, tumor ili karcinom- koja je razlika?” (), adresa: <https://www.virtualna-ordinacija.com/rak-tumor-ili-karcinom-koja-je-razlika/>. (proučavano: 26.03.2024.)
- [11] S. W. PhD. „Benign vs Malignant Tumors.” (), adresa: <https://www.technologynetworks.com/cancer-research/articles/benign-vs-malignant-tumors-364765>. (proučavano: 28.03.2024.)
- [12] Cmed. „Challenges in Glioblastoma Studies.” (), adresa: <https://cmedresearch.com/resources/glioblastoma-experience/>. (proučavano: 28.03.2024.)

- [13] N. A. M. M. Hrvatska elektronička medicinska edukacija Steven A. Goldman MD PhD. „Pregled tumora mozga.” (), adresa: <https://hemed.hr/Default.aspx?sid=16918>. (proučavano: 30.03.2024.)
- [14] A. C. Society. „Brain and Spinal Cord Tumor in Adults Early Detection, Diagnosis, and Staging.” (), adresa: <https://www.cancer.org/cancer/types/brain-spinal-cord-tumors-adults/detection-diagnosis-staging.html>. (proučavano: 30.03.2024.)
- [15] M. K. M. Hrvatska elektronička medicinska edukacija. „Magnetna rezonancija (MR).” (), adresa: <https://hemed.hr/Default.aspx?sid=19473>. (proučavano: 30.03.2024.)
- [16] M. Sakai, N. Kashiwagi, K. Nakanishi i dr., „Nonbrain metastases seen on magnetic resonance imaging during metastatic brain tumor screening,” *Japanese Journal of Radiology*, sv. 41, br. 4, str. 367–381, 2023.
- [17] K. S. M. EZRA. „How to Choose Between a PET Scan vs. MRI.” (), adresa: <https://ezra.com/blog/pet-scan-vs-mri>. (proučavano: 02.04.2024.)
- [18] M. K. M. Hrvatska elektronička medicinska edukacija. „Kontrastna sredstva u radiologiji.” (), adresa: https://hemed.hr/Default.aspx?sid=19470#v43824839_hr. (proučavano: 02.04.2024.)
- [19] I. z. M. K. Hrvatska enciklopedija. „apsces.” (), adresa: <https://www.enciklopedija.hr/clanak/apsces>. (proučavano: 01.07.2024.)
- [20] IBM. „What is artificial intelligence (AI)?” (), adresa: <https://www.ibm.com/topics/artificial-intelligence>. (proučavano: 07.04.2024.)
- [21] H. Enciklopedija. „umjetna inteligencija.” (), adresa: <https://www.enciklopedija.hr/clanak/umjetna-inteligencija#clanak>. (proučavano: 07.04.2024.)
- [22] D. Kopljar, „Konvolucijske neuronske mreže,” Završni rad, SVEUČILIŠTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA, 2016.
- [23] IBM. „Understanding the different types of artificial intelligence.” (), adresa: <https://www.ibm.com/think/topics/artificial-intelligence-types>. (proučavano: 07.04.2024.)
- [24] P. industrija tehnologija. „Tri vrste umjetne inteligencije: Razumijevanje umjetne inteligencije.” (), adresa: <https://www.pit.ba/tri-vrste-umjetne-inteligencije-razumijevanje-umjetne-inteligencije/>. (proučavano: 07.04.2024.)
- [25] E. o. A. Sveučilište u Helsinkiju Sveučilište u Zagrebu. „Filozofija umjetne inteligencije.” (), adresa: <https://course.elementsofai.com/hr/1/3>. (proučavano: 07.04.2024.)
- [26] E. Britannica. „Turing test.” (), adresa: <https://www.britannica.com/technology/Turing-test>. (proučavano: 07.04.2024.)
- [27] Tableau. „What is the history of artificial intelligence (AI)?” (), adresa: <https://www.tableau.com/data-insights/ai/history>. (proučavano: 08.04.2024.)

- [28] E. parlament. „Što je umjetna inteligencija i kako se upotrebljava?“ (), adresa: <https://www.europarl.europa.eu/topics/hr/article/20200827STO85804/sto-je-umjetna-inteligencija-i-kako-se-upotrebljava>. (proučavano: 08.04.2024.)
- [29] P. Malik, M. Pathania, V. K. Rathaur i dr., „Overview of artificial intelligence in medicine,” *Journal of family medicine and primary care*, sv. 8, br. 7, str. 2328–2331, 2019.
- [30] P. The Medical Futurist Dr. Bertalan Mesko. „Top 7 AI Examples In Healthcare - The Medical Futurist.” (), adresa: <https://www.youtube.com/watch?v=mkiDXTS6-mU>. (proučavano: 11.04.2024.)
- [31] IBM. „The benefits of AI in healthcare.” (), adresa: <https://www.ibm.com/think/insights/ai-healthcare-benefits>. (proučavano: 26.04.2024.)
- [32] IBM. „What is artificial intelligence in medicine?” (), adresa: <https://www.ibm.com/topics/artificial-intelligence-medicine>. (proučavano: 26.04.2024.)
- [33] I. z. M. K. Hrvatska enciklopedija. „neuronska mreža.” (), adresa: <https://www.enciklopedija.hr/clanak/neuronska-mreza>. (proučavano: 29.04.2024.)
- [34] P. d. s. S. L. Fakultet elektronike i računarstva. „Neuronske mreže: Perceptron.” (), adresa: https://www.fer.unizg.hr/_download/repository/04-Perceptron.pdf. (proučavano: 26.04.2024.)
- [35] G. S. 3Blue1Brown. „But what is a neural network? | Chapter 1, Deep learning.” (), adresa: https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=3. (proučavano: 26.04.2024.)
- [36] G. S. 3Blue1Brown. „Gradient descent, how neural networks learn | Chapter 2, Deep learning.” (), adresa: https://www.youtube.com/watch?v=IHZwWFHwa-w&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=4. (proučavano: 26.04.2024.)
- [37] M. Memon. „ANN vs CNN vs RNN: Neural Networks Guide.” (), adresa: <https://levity.ai/blog/neural-networks-cnn-ann-rnn>. (proučavano: 29.04.2024.)
- [38] ritvikmath. „Perceptron.” (), adresa: <https://www.youtube.com/watch?v=4Gac5I64LM4>. (proučavano: 02.05.2024.)
- [39] P. Fumić, „Duboko učenje: pregled područja,” Diplomski rad, Sveučilište u Zagrebu, Fakultet organizacije i informatike, 2021.
- [40] L. Shukla. „Designing Your Neural Networks.” (), adresa: <https://towardsdatascience.com/designing-your-neural-networks-a5e4617027ed>. (proučavano: 05.05.2024.)
- [41] B. Tripathy. „Can a neural network be partially connected?” (), adresa: https://www.researchgate.net/post/Can_a_neural_network_be_partially_connected. (proučavano: 05.05.2024.)
- [42] S. Saha. „A Guide to Convolutional Neural Networks — the ELI5 way.” (), adresa: <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>. (proučavano: 05.05.2024.)
- [43] I. z. h. j. i. j. Struna hrvatsko strukovno nazivlje. „konvolucija.” (), adresa: <http://struna.ihjj.hr/naziv/konvolucija/19228/>. (proučavano: 08.05.2024.)

- [44] G. S. 3Blue1Brown. „But what is a convolution?“ (), adresa: <https://www.youtube.com/watch?v=KuXjwB4LzSA>. (proučavano: 08.05.2024.)
- [45] Dremio. „Pooling Layers.“ (), adresa: <https://www.dremio.com/wiki/pooling-layers/>. (proučavano: 10.05.2024.)
- [46] IntuitiveML. „What is pooling? | CNN's 3.“ (), adresa: <https://www.youtube.com/watch?v=KKmCnwGzSv8>. (proučavano: 10.05.2024.)
- [47] Datanics. „Simple explanation of how pooling layer works in convolutional neural network?“ (), adresa: <https://www.youtube.com/watch?v=PuFNG721zM8>. (proučavano: 10.05.2024.)
- [48] S. Khosla. „CNN | Introduction to Pooling Layer.“ (), adresa: <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>. (proučavano: 10.05.2024.)
- [49] D. Spodarets. „A Guide to the YOLO Family of Computer Vision Models.“ (), adresa: <https://dataphoenix.info/a-guide-to-the-yolo-family-of-computer-vision-models/>. (proučavano: 15.05.2024.)
- [50] A. Mirkhan. „YOLO Algorithm: Real-Time Object Detection from A to Z.“ (), adresa: <https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z>. (proučavano: 15.05.2024.)
- [51] J. Redmon, S. Divvala, R. Girshick i A. Farhadi, „You only look once: Unified, real-time object detection,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016., str. 779–788.
- [52] T. Cvija, „Detekcija karcinoma mokraćnog mjehura korištenjem YOLO algoritma,” Diplomski rad, Sveučilište u Rijeci, Tehnički fakultet, 2022.
- [53] A. Persson. „YOLOv1 from Scratch.“ (), adresa: https://www.youtube.com/watch?v=n9_XyCGr-MI. (proučavano: 15.05.2024.)
- [54] S. M. Pattern Analysis i C. Learning. „Visual Object Classes Challenge 2012 (VOC2012).“ (), adresa: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html>. (proučavano: 23.05.2024.)
- [55] H. Ö. Computer Vision with Hüseyin Özdemir. „YOLO Algorithm, Object Detection with YOLOv1.“ (), adresa: <https://www.youtube.com/watch?v=NAwleAip-9Q>. (proučavano: 15.05.2024.)
- [56] C. Wetzel. „The Horses of Medieval Times Weren't Much Bigger Than Modern-Day Ponies.“ (), adresa: <https://www.smithsonianmag.com/smart-news/medieval-warhorses-were-actually-the-size-of-ponies-180979389/>. (proučavano: 23.05.2024.)
- [57] L. Z. Q. Zou. „YOLO: You Only Look Once Unified Real-Time Object Detection.“ (), adresa: <https://web.cs.ucdavis.edu/~yjlee/teaching/ecs289g-winter2018/YOLO.pdf>. (proučavano: 02.06.2024.)
- [58] A. Caplić, „Detekcija objekata na slikama temeljena na konvolucijskoj neuronskoj mreži,” Završni rad, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, 2022.

- [59] S. M. LearnOpenCV. „Intersection Over Union (IoU) in Object Detection Segmentation.” (), adresa: <https://www.youtube.com/watch?v=RgQbweTwrkU>. (proučavano: 23.05.2024.)
- [60] A. Rosebrock. „Intersection over Union (IoU) for object detection.” (), adresa: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>. (proučavano: 24.05.2024.)
- [61] M. Menegaz. „Understanding YOLO.” (), adresa: <https://hackernoon.com/understanding-yolo-f5a74bbc7967>. (proučavano: 13.06.2024.)
- [62] N. Chakrabarty. „Brain MRI Images for Brain Tumor Detection.” (), adresa: <https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>. (proučavano: 26.03.2024.)
- [63] D. Shah. „COCO Dataset: All You Need to Know to Get Started.” (), adresa: <https://www.v7labs.com/blog/coco-dataset-guide>. (proučavano: 13.06.2024.)
- [64] A. Mehra. „Evolution of YOLO Object Detection Model From V5 to V8.” (), adresa: <https://www.labellerr.com/blog/evolution-of-yolo-object-detection-model-from-v5-to-v8/>. (proučavano: 13.06.2024.)
- [65] F. Computer Vision Engineer. „Train Yolov8 object detection on a custom dataset | Step by step guide | Computer vision tutorial.” (), adresa: <https://www.youtube.com/watch?v=m9fH9OWn8YM&t=1980s>. (proučavano: 02.06.2024.)

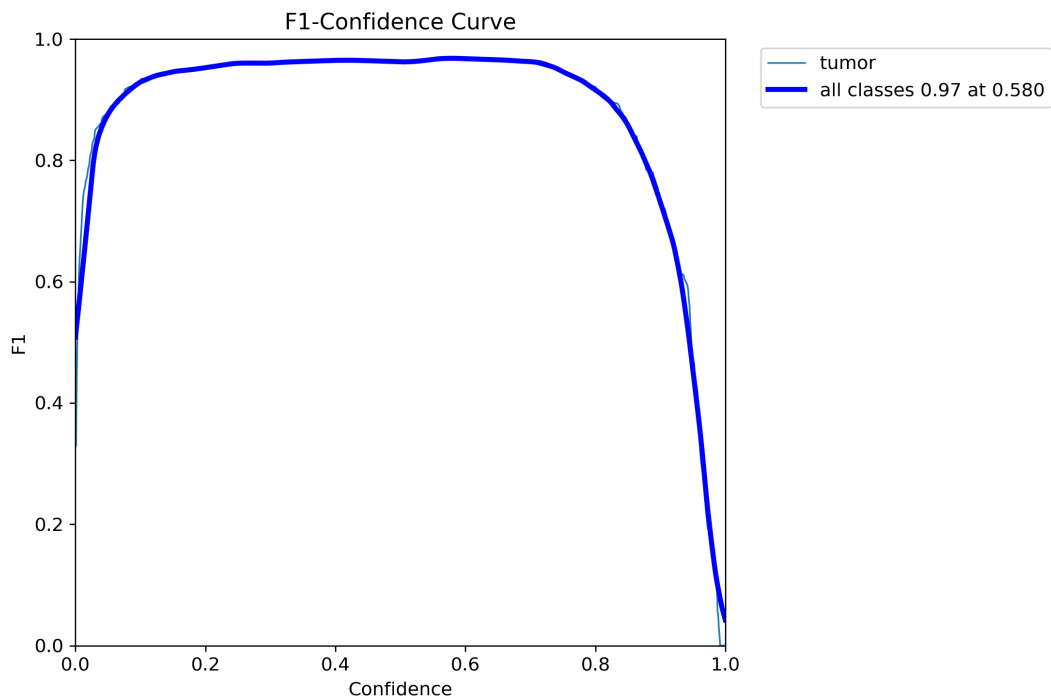
Popis slika

1.	Razvoj tumora (Izvor: [2])	3
2.	Smrtnost od napredovalog malignog tumora u ovisnosti o mišićnoj masi (Izvor: [6])	4
3.	Razlika između građe benignog i malignog tumora (Izvor: [11])	5
4.	Smrtnost kod glioblastoma po dobnim kategorijama (Izvor: [12])	6
5.	Metastaziranje tumora mozga (MR) (Izvor: [16])	7
6.	McCulloch-Pitts model neurona - perceptron (Izvor: izrada autora)	13
7.	Sigmoidalna funkcija (Izvor: izrada autora)	14
8.	ReLU funkcija (Izvor: izrada autora)	14
9.	Struktura umjetne neuronske mreže (Izvor: [40])	15
10.	Jednostavni prikaz slojeva konvolucijske neuronske mreže (Izvor: [42])	16
11.	Izvršavanje konvolucije uz pomoć filtra 1. iteracija (Izvor: izrada autora)	17
12.	Izvršavanje konvolucije uz pomoć filtra 2. iteracija (Izvor: izrada autora)	18
13.	Moguće zadatke skrivenih slojeva u prepoznavanju objekata na slici (Izvor: [40]) .	18
14.	Primjer sabiranja korištenjem max pooling-a (Izvor: [48])	19
15.	Primjer detekcije korištenjem YOLOv1 algoritma (Izvor: [55])	22
16.	Primjer 3×3 rešetke nad slikom (Izvor: [56]) uz doradu autora	23
17.	Primjer graničnog okvira (Izvor: [56]) uz doradu autora)	24
18.	Primjer rada <i>IOU</i> -a na slici (Izvor: [56]) uz doradu autora)	24
19.	Vizualni prikaz <i>IOU</i> formule (Izvor: izrada autora)	25
20.	Uvećana ćelija na poziciji (2,1) (Izvor: [56]) uz doradu autora)	25
21.	Određivanje koordinata graničnog okvira (Izvor: [56]) uz doradu autora)	26
22.	Vizualni prikaz detekcije klasa po ćelijama (Izvor: [51])	27
23.	Prikaz arhitekture YOLOv1 algoritma (Izvor: [51])	28

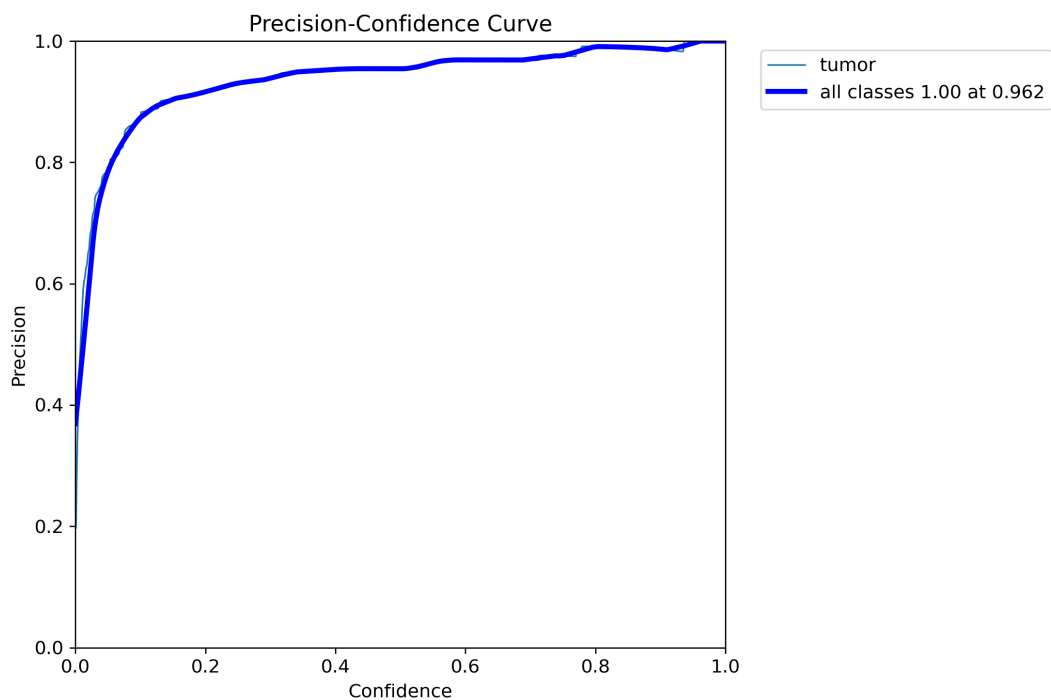
24.	Usporedba YOLOv8 algoritma i ranijih verzija (Izvor: [64])	31
25.	CVAT anotiranje tumora (Izvor: izrada autora)	32
26.	Prikaz Python koda (Izvor: izrada autora)	33
27.	Prikaz Python koda i okruženja u Kaggleu (Izvor: izrada autora)	34
28.	Matrica zabune za 2. verziju modela - (Izvor: izrada autora)	35
29.	Matrica zabune za 3. verziju modela - (Izvor: izrada autora)	35
30.	Prikaz datoteke "results" - 200 epoha (2. verzija modela) (Izvor: izrada autora) .	36
31.	Prikaz datoteke "results" - 357 epoha (3. verzija modela) (Izvor: izrada autora) .	36
32.	Prikaz datoteke "results" - 100 epoha (1. verzija modela) (Izvor: izrada autora) .	37
33.	Prikaz slika za treniranje modela (Izvor: izrada autora)	38
34.	Prikaz stvarnih graničnih okvira (Izvor: izrada autora)	39
35.	Prikaz predikcija graničnih okvira 2. verzije modela (Izvor: izrada autora)	40
36.	Usporedba predikcija sa stvarnim graničnim okvirima (Izvor: izrada autora)	41
37.	Prikaz stvarnih graničnih okvira (Izvor: izrada autora)	42
38.	Prikaz predikcija graničnih okvira 3. verzije modela (Izvor: izrada autora)	43
39.	F1 krivulja - (Izvor: izrada autora)	52
40.	P krivulja - (Izvor: izrada autora)	52
41.	PR krivulja - (Izvor: izrada autora)	53
42.	R krivulja - (Izvor: izrada autora)	53
43.	F1 krivulja - (Izvor: izrada autora)	54
44.	P krivulja - (Izvor: izrada autora)	54
45.	PR krivulja - (Izvor: izrada autora)	55
46.	R krivulja - (Izvor: izrada autora)	55

1. Prilog 1

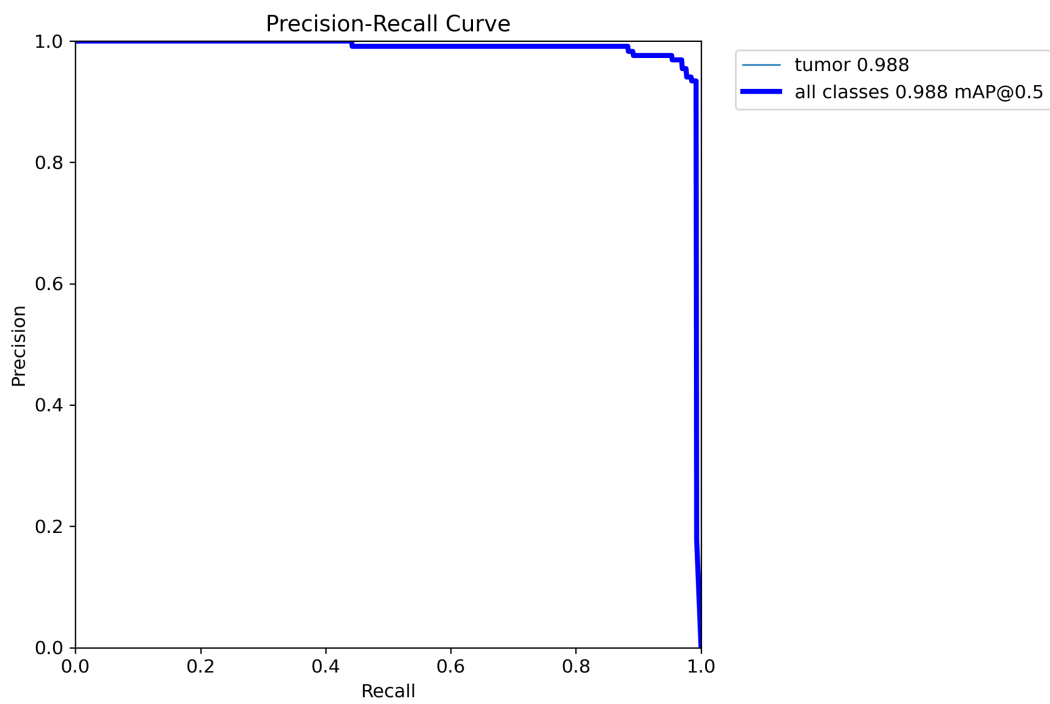
Uvid u proces učenja grafova (F1 krivulja, P krivulja, PR krivulja i R krivulja) za 2. verziju modela.



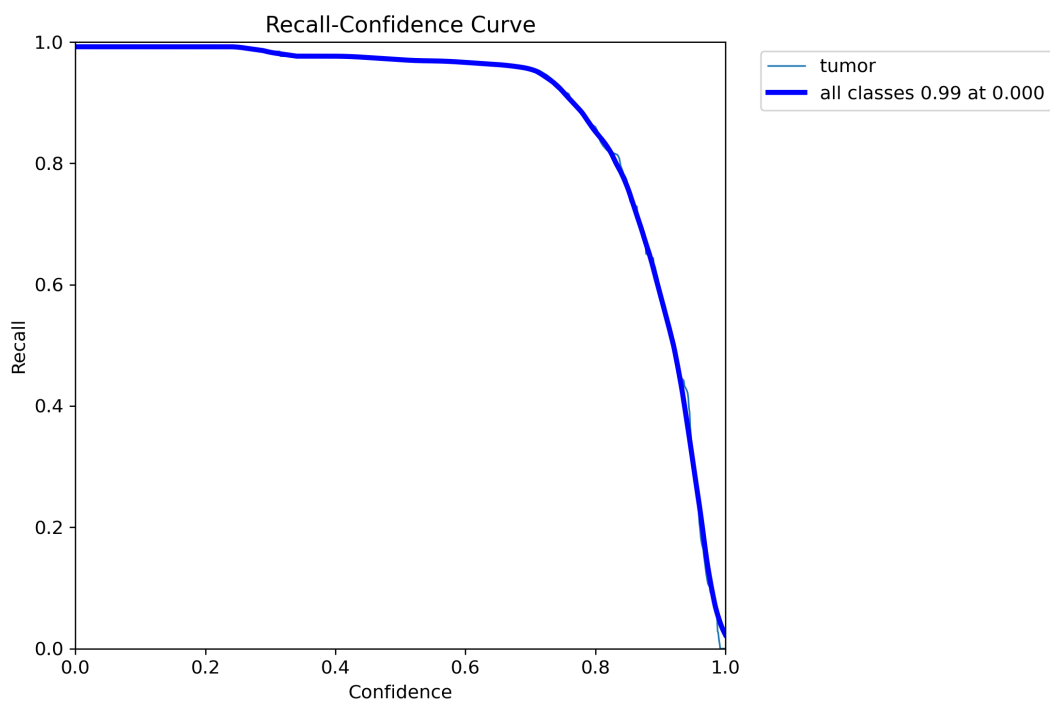
Slika 39: F1 krivulja - (Izvor: izrada autora)



Slika 40: P krivulja - (Izvor: izrada autora)



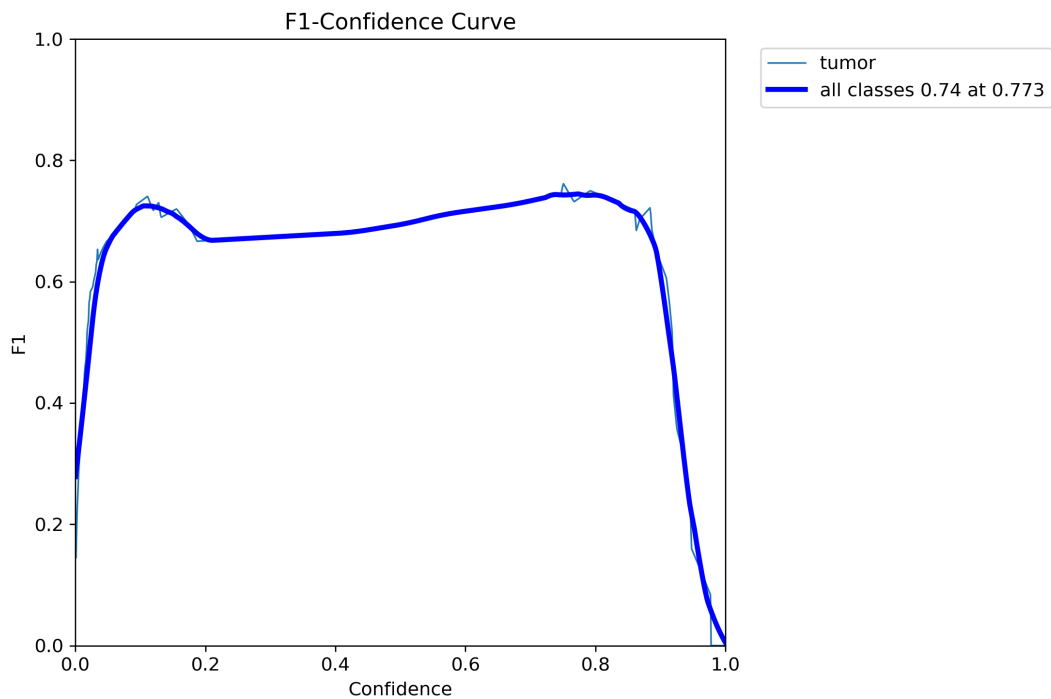
Slika 41: PR krivulja - (Izvor: izrada autora)



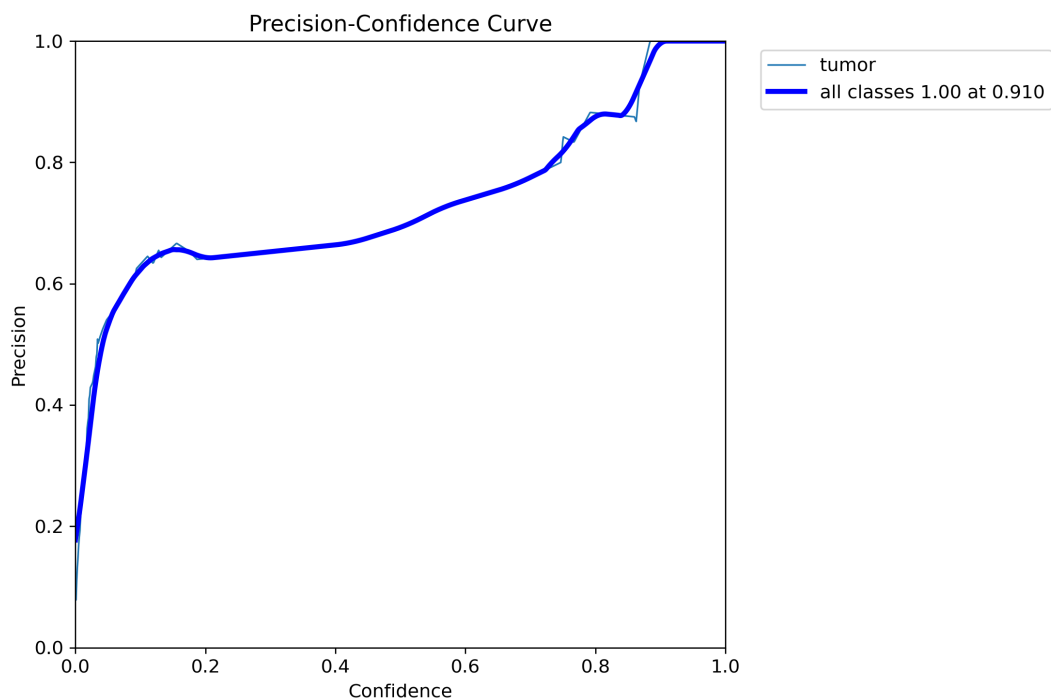
Slika 42: R krivulja - (Izvor: izrada autora)

2. Prilog 2

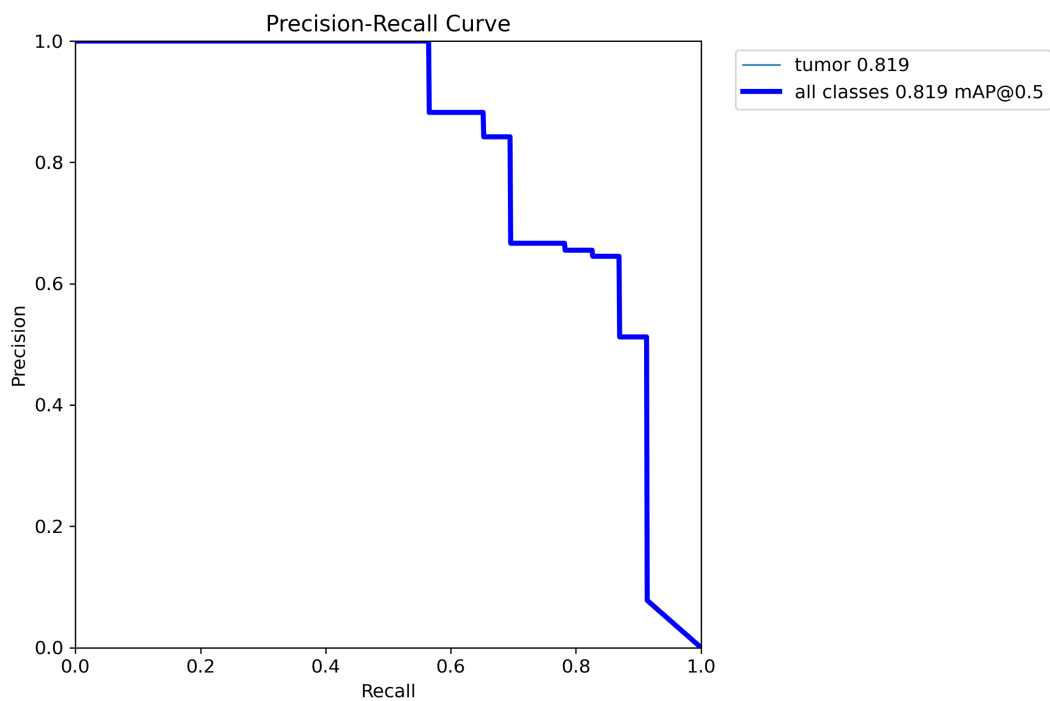
Uvid u proces učenja grafova (F1 krivulja, P krivulja, PR krivulja i R krivulja) za 3. verziju modela.



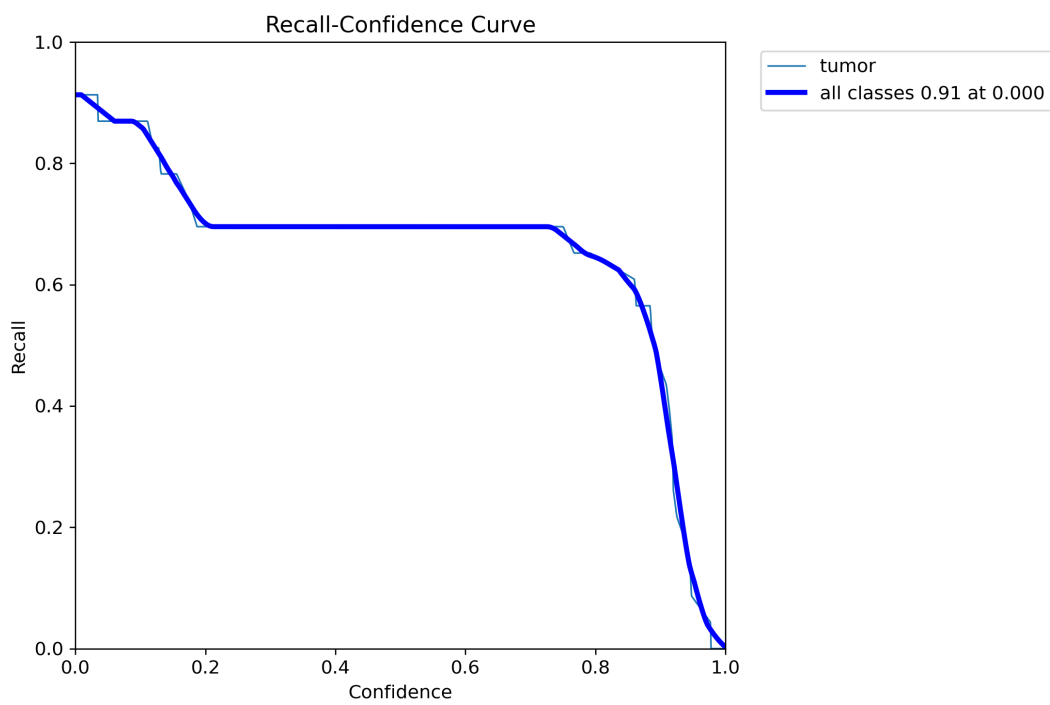
Slika 43: F1 krivulja - (Izvor: izrada autora)



Slika 44: P krivulja - (Izvor: izrada autora)



Slika 45: PR krivulja - (Izvor: izrada autora)



Slika 46: R krivulja - (Izvor: izrada autora)