# Detekcija karcinoma mokraćnog mjehura korištenjem YOLO algoritma

**Cvija, Tajana**

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

# DETEKCIJA KARCINOMA MOKRAĆNOG MJEHURA KORIŠTENJEM YOLO ALGORITMA

Rijeka, rujan 2022.

Tajana Cvija

0069079320

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

# DETEKCIJA KARCINOMA MOKRAĆNOG MJEHURA KORIŠTENJEM YOLO ALGORITMA

Mentor: Prof. dr. sc. Zlatan Car

Rijeka, rujan 2022.

Tajana Cvija

0069079320

**SVEUČILIŠTE U RIJECI**
**TEHNIČKI FAKULTET**
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za automatiku i elektroniku**
Predmet: **Primjena umjetne inteligencije**
Grana: **2.03.06 automatizacija i robotika**

# ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Tajana Cvija (0069079320)**
Studij: Diplomski sveučilišni studij elektrotehnike
Modul: Automatika

Zadatak: **Detekcija karcinoma mokraćnog mjehura korištenjem YOLO algoritma/Urinary bladder cancer detection using YOLO algorithm**

Opis zadatka:

Predstaviti problem detekcije objekata sa slike i postojeća rješenja istoga. Opisati konvolucijske neuronske mreže. Odabrati jedan od YOLO algoritama i primijeniti ga na problemu detekcije raka mokraćnog mjehura. Komentirati dobivene rezultate i učinkovitost algoritma.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Tajana Cvija

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Prof. dr. sc. Zlatan Car

Predsjednik povjerenstva za diplomski ispit:

Prof. dr. sc. Viktor Sučić

# IZJAVA

Sukladno članku 8. Pravilnika o diplomskom radu, diplomskom ispitu i završetku diplomskih sveučilišnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 1. veljače 2020. godine, izjavljujem da sam samostalno izradila diplomski rad prema zadatku preuzetom dana 21. ožujka 2022. godine.

Rijeka, 20. rujna 2022.

Tajana Cvija

Tajana Cvija

# ZAHVALA

Hvala prof. dr. sc. Zlatanu Caru, mom mentoru, i asist. dr. sc. Ivanu Lorencinu, mom neslužbenom mentoru kroz protekle dvije godine, na ukazanoj pomoći tijekom izrade kako ovog rada, tako i brojnih prethodnih projekata.

Hvala Klinici za urologiju KBC-a Rijeka pod vodstvom prof. dr. sc. Josipa Španjola na ustupljenim podacima.

# Sadržaj

# 1. UVOD

U zadnjih je nekoliko desetljeća brzorastuća digitalna tehnologija postala sastavnica gotovo svakog područja ljudskih života. U medicini se računala koriste za čitanje, prikaz, pohranu i obradu medicinskih podataka prikupljenih iz različitih dijagnostičkih uređaja te kao pomoć pri dijagnosticiranju raznih zdravstvenih stanja pa se može govoriti o računalno potpomognutoj dijagnostici (*Computer Aided Diagnostics,* CAD) [1]. Računalno potpomognuta dijagnostika obuhvaća obradu slika, strojno i duboko učenje, računalni vid, matematiku i fiziku i koristi ih za pomoć liječnicima pri dijagnosticiranju bolesti, određivanju terapije i donošenju odluka [2].

U ovom radu naglasak je na korištenju računalnog vida u obliku YOLO (*You Only Look Once*) algoritma za detekciju objekata pri detekciji karcinoma mokraćnog mjehura.

Karcinom mokraćnog mjehura deveti je najčešći karcinom u svijetu i drugi najčešći karcinom genitourinarnog sustava [3]. Drugo poglavlje pobliže opisuje kako i gdje on nastaje, koji su simptomi bolesti, stadiji, prognoze i metode dijagnosticiranja. Izdvojene su metode dijagnosticiranja kompjutorizirana tomografija i magnetska rezonanca na čijim će slikovnim prikazima u šestom poglavlju biti izvršena detekcija karcinoma mokraćnog mjehura.

Prije no što se prijeđe na samu detekciju karcinoma, potrebno je uvesti pojmove umjetne inteligencije i strojnog učenja te njihovu ulogu u medicini što je učinjeno u trećem poglavlju.

U četvrtom poglavlju izdvojeno je posebno područje umjetne inteligencije - računalni vid i problematika detekcije objekata. Predstavljene su umjetne neuronske mreže s naglaskom na konvolucijske neuronske mreže koje se najčešće koriste pri detekciji objekata. Opisana je njihova struktura, princip rada i učenja te parametri vrednovanja detekcije objekata i skupovi podataka na kojima konvolucijske mreže uče. Na samom kraju poglavlja dan je pregled nekih od algoritama koji se koriste u detekciji objekata.

Peto poglavlje donosi opis YOLO (*You Only Look Once*) algoritma za detekciju objekata. Izdvojena su izdanja algoritma YOLO (prvo izdanje), YOLOv2 i YOLO9000, YOLOv3, YOLOv4, skalirani YOLOv4 i YOLOv7 te YOLO-R (*You Only Learn One Representation*) algoritam. Poseban je naglasak pritom na najnovijem YOLOv7 algoritmu.

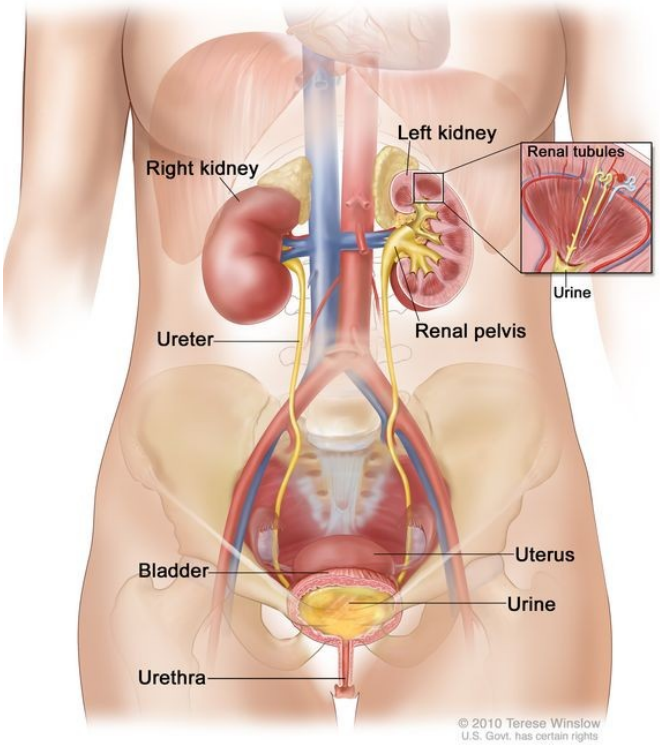
Praktična primjena YOLOv7 algoritma dana je u šestom poglavlju gdje se koristi pri detekciji karcinoma mokraćnog mjehura na slikama presjeka trbušne šupljine dobivenima kompjutoriziranom tomografijom i magnetskom rezonancom. Opisan je skup podataka na kojem je algoritam učio, priprema podataka i proces učenja, a zatim su prikazani i komentirani rezultati ispitivanja algoritma.

Prilog A sadrži korišteni kod za učenje i ispitivanje algoritma, a Prilog B donosi grafove koji daju prošireni uvid u tijek učenja.

# 2. KARCINOM MOKRAĆNOG MJEHURA

Karcinom (rak) naziv je za skupinu oboljenja koja nastaju uslijed nekontroliranog rasta tumorskih stanica. Tumorske stanice nastaju kada stanice tkiva razviju genetske mutacije u svojoj DNA. Dolazi do strukturalnih promjena dviju vrsta gena koji kontroliraju dijeljenje i umiranje stanica. Mutacijom protoonkogena nastaju onkogeni koji potiču nekontrolirani rast i umnažanje stanica. Istovremeno mutacijom na tumorsurpresorskim genima ili antionkogenima, koji reguliraju programiranu staničnu smrt (apoptozu), može doći do njihove smanjene aktivnosti ili potpune deaktivacije, što će rezultirati produljenim životnim vijekom mutiranih stanica. Mutirane stanice oblikuju abnormalne nakupine tkiva - tumore. Tumori mogu biti dobroćudni (benigni) i zloćudni (maligni). Zloćudni tumori uzročnici su karcinoma.

Mokraćni mjehur šuplji je organ u donjem predjelu trbušne šupljine, čija je prvotna zadaća skladištenje mokraće iz bubrega, kako je i prikazano na primjeru mokraćnog sustava žene na Slici 2.1.




*Slika 2.1. Organi mokraćnog sustava žene [4]*

Karcinom mokraćnog mjehura jedan je od najčešćih tipova karcinoma i drugi najčešći karcinom genitourinarnog sustava [3]. S trostruko većom vjerojatnošću javlja se kod muškaraca nego kod

žena, vjerojatnost pojave raste s dobi, a istraživanja pokazuju da je češći u razvijenim i industrijaliziranim zemljama nego u onima manje razvijenima [3, 5].

Značajan utjecaj u razvoju karcinoma mokraćnog mjehura ima naslijeđe pa je tako dvostruko veća vjerojatnost razvoja u osoba čiji su roditelji imali ovu vrstu karcinoma nego kod onih čiju ju roditelji nisu imali [3]. Osim naslijeđa, na pojavu ove vrste karcinoma utječe i niz drugih uzročnika. Jedan je od najvećih uzročnika pojave karcinoma mokraćnog mjehura pušenje [3, 5, 6]. Metaboliti proizvedeni pušenjem iz tijela se izlučuju mokraćom. Produljeni dodir štetnih tvari u mokraći sa stijenkom mokraćnog mjehura doprinosi pojavi mutacija, a posljedično tome i karcinoma [6]. Pušači imaju od dva do šest puta veću vjerojatnost razvoja karcinoma od nepušača [3]. Osim pušenja, veliku ulogu igra i izloženost nizu vanjskih čimbenika među kojima su prvi anilinske boje, guma, koža i proizvodi od kože i organske kemikalije [3, 5, 6]. Ove tvari ulaze u tijelo udisanjem ili kroz kožu te izlaze iz tijela mokraćom. Od ostalih čimbenika koji uzrokuju karcinom mokraćnog mjehura valja spomenuti lijekove (analgetici - fenacetin, kemoterapeutici - ciklofosfamid), zračenje, kronične infekcije mokraćnog sustava (bakterijske, *Schistosoma haematobium*), kamenac u mokraćnom sustavu, lošu prehranu i nedovoljan unos tekućine [3, 5, 6].

Karcinom mokraćnog mjehura u 98% slučajeva nastaje na epitelnom tkivu [5]. Karcinom epitelnog tkiva u više je od 90% slučajeva karcinom prijelaznog epitela (urotelni karcinom), a rijetko je karcinom pločastih stanica (planocelularni karcinom) ili adenokarcinom. Neepitelni tumori iznimno su rijetki i tada je riječ o rabdosarkomima, miosarkomima, fibromima, lejomiomima i hemangiomima [5, 6].

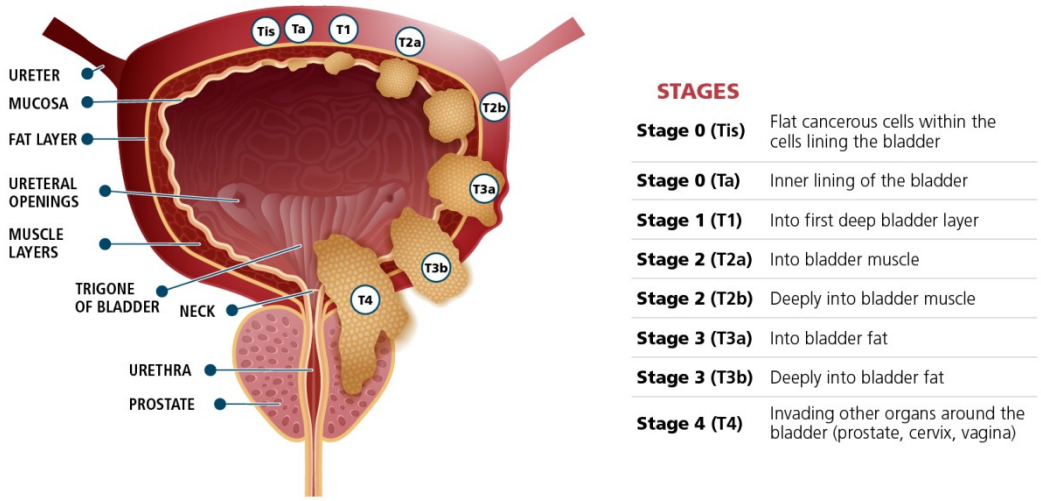
Kada se spomene karcinom mokraćnog mjehura, najčešće se misli na karcinom urotela. Urotel (prijelazni epitel) vrsta je elastičnog epitelnog tkiva koje s unutarnje strane oblaže mokraćni mjehur, mokraćovode i kolektorski sustav bubrega. Na urotel se nastavljaju suburotelno vezivno tkivo (lamina propria) i mišićno tkivo [6].

Razvoj karcinoma mokraćnog mjehura započinje nastankom baze tumorskih stanica u urotelu [3, 6]. Tumorska izraslina može dalje rasti u šupljinu mokraćnog mjehura, međutim, pri određivanju stadija važno je samo koliko je duboko baza tumora prodrla u tkivo [6]. Ovisno o tome tumori se u grubo dijele na mišićno neinvazivne (površinske, superficijalne) i mišićno invazivne [3, 5, 6].

Mišićno neinvazivni tumori javljaju se u 75%-85% slučajeva [5]. Među njih pripadaju stadiji Ta, T1 i TIS/CIS (*tumor in situ/carcinoma in situ*) [3, 5, 6]. Oko 70% površinskih tumora je u Ta stadiju u kojem se baza tumora nalazi u urotelnom tkivu s izraslinom koja raste u šupljinu mjehura [5, 6]. Stadij T1 obuhvaća oko 20% slučajeva i u njemu baza prodire u laminu propriju dok tumorska izraslina raste u šupljinu mjehura [5, 6]. Preostalih 10% slučajeva je TIS/CIS stadij u kojem je baza tumora u urotelnom tkivu, ali izrasline nema već je tumor u obliku crvenkaste lezije u razini s urotelom ili tek blago izdignute [6]. TIS nije opasan, međutim može biti prekursor invazivnog karcinoma [5].

Mišićno invazivni tumori javljaju se u 10%-15% slučajeva [6]. Među njih pripadaju stadiji T2, T3 i T4 [3, 5, 6]. U stadiju T2 baza tumora podire u mišićno tkivo, u stadiju T3 u perivezikalno masno tkivo, koje se nalazi oko mokraćnog mjehura, a u stadiju T4 prodire u okolne organe (prostata, maternica, zdjelična stijenka) [3, 5, 6]. Mišićno invazivni tumori imaju sklonost metastaziranju u limfne čvorove i druge organe (pluća, jetra, nadbubreg, kosti) [5].

Slika 2.2 daje vizualni prikaz opisanih stadija tumora mokraćnog mjehura.




*Slika 2.2. Stadiji tumora mokraćnog mjehura* [7]

Mišićno neinvazivni i mišićno invazni tumori jako se razlikuju po svom malignom potencijalu i daljnjoj prognozi te zahtijevaju različit pristup liječenju.

Mišićno neinvazivne tumore uglavnom tretira urolog i izraslinu odstranjuje mehanički transuretralnom resekcijom [5, 6]. Ako je potrebno može se pristupiti i kemoterapiji i/ili imunoterapiji, a u rijetkim slučajevima dolazi do kirurškog odstranivanja (cistektomije)

mokraćnog mjehura [5]. Postoji sklonost redicivima (povratak tumora) pa je, osim otklanjanja tumora, važno primijeniti odgovarajuću terapiju i redovito pratiti stanje pacijenta čak i ako je tumor odstranjen [5]. Prognoza je vrlo dobra i vjerojatnost petogodišnjeg preživljavanja veća je od 90% [5].

Mišićno invazivni tumori imaju značajno lošiju kliničku sliku i složeniju terapiju. Budući da su prodrli u mišićno tkivo i/ili dublje, mehaničko odstranjivanje izrasline nije opcija [6]. Teži se maksimizirati učinak terapije i poboljšati kliničku sliku. Metode kojima se pristupa uključuju kemoterapiju, zračenje, radikalnu cistektomiju i parcijalnu cistektomiju u kombinaciji s kemoterapijom i zračenjem [3, 5, 6]. Multimodalna terapija pokazala se kao najbolji pristup u tretiranju mišićno invazivnih tumora i uključuje zajednički rad urologa, onkologa i radiologa [3]. Vjerojatnost petogodišnjeg preživljavanja je 66%, a desetogodišnjeg 43% [5]. U slučaju da je karcinom metastazirao, vjerojatnost izlječenja iznimno je mala. Ipak, pristupa se terapiji kako bi se ublažili simptomi, poboljšala kvaliteta života i produljio život pacijenta.

Karcinom mokraćnog mjehura uglavnom se otkrije u ranim stadijima. Razlog tome je da su tumorske izrasline nestabilne i lako otpadnu što uzrokuje pojavu krvi u mokraći (hematuriju) [6]. Govori se o makrohematuriji, koja je vidljiva golim okom, i mikrohematuriji, koju se otkriva mikroskopskom analizom. Ponekad se uz krvarenje javlja i bol pri mokrenju (dizurija), neodgodiva potreba za mokrenjem (urgentnost) i učestalo mokrenje [3].

Kada se pojave simptomi, nekoliko je načina dijagnosticiranja karcinoma mokraćnog mjehura. Najčešće korišteni su citoskopija, biopsija i citologija urina, a koriste se i kompjutorizirana tomografija (CT, *Computed Tomography*), magnetska rezonanca (MRI, *Magnetic Resonance Imaging*) i ultrazvuk [8]. Nakon utvrđivanja prisutnosti tumora, pristupa se određivanju stadija karcinoma. Ono se može izvršiti kompjutoriziranom tomografijom, magnetskom rezonancom, pozitronskom emisijskom tomografijom (PET), skeniranjem kostiju i rendgenom prsnog koša [9].

Ovaj rad izdvaja dvije metode, kompjutoriziranu tomografiju i magnetsku rezonancu.

Kompjutorizirana tomografija radiološka je metoda koja koristi rendgenske zrake kako bi se dobio slikovni prikaz presjeka tijela ili organa. Slike se generiraju u više slojeva pa se često metodu naziva i višeslojnom kompjutoriziranom tomografijom (MSCT, *Multislice Computed Tomography*) [10]. Višeslojne slike dobivaju se iz više osi - frontalne, horizontalne i sagitalne, a

moguća je i integracija slika iz svih osi kako bi se dobio trodimenzionalni prikaz. Kompjutorizirana se tomografija može raditi uz upotrebu kontrastnog bojila, koje se primjenjuje oralno ili intravenski, kako bi se na dobivenoj slici istaknuli pojedini organi ili dijelovi organa [8]. Ova se metoda koristi kod detekcije i utvrđivanja stadija tumora mokraćnog mjehura.

Magnetska rezonanca radiološka je metoda koja za dobivanje slika presjeka tijela ili organa koristi jako magnetsko polje i radiovalove [11]. Kao i kod kompjutorizirane tomografije, slike se dobivaju u više slojeva i iz frontalne, horizontalne i sagitalne osi te je iz njih moguće dobiti trodimenzionalni prikaz. Magnetska se rezonanca također može raditi uz primjenu kontrastnog bojila koje se često koristi pri utvrđivanju prisutnosti i određivanju stadija tumora na mokraćnom mjehuru [8].

Kompjutorizirana tomografija i magnetska rezonanca pokazuju približno jednaku točnost pri utvrđivanju prisutnosti tumora. no utvrđivanje stadija preciznije je korištenjem magnetske rezonance [8].

U novije se vrijeme za otkrivanje karcinoma mokraćnog mjehura i određivanje njegovoga stadija koristi virtualna citoskopija. U suštini se radi o integraciji CT ili MRI slika kako bi se dobio trodimenzionalni prikaz mokraćnog mjehura pa se govori o CT virtualnoj citoskopiji (CT VC) ili MR virtualnoj citoskopiji (MR VC) [8]..

U nastavku rada bit će korištene CT i MRI slike frontalne, horizontalne i sagitalne osi trbušne šupljine kako bi se naučio model umjetne neuronske mreže da prepoznaje tumorske nakupine.

# 3. PRIMJENA UMJETNE INTELIGENCIJE U MEDICINI

Umjetna inteligencija (AI, *Artificial Intelligence*) u posljednjih je nekoliko desetljeća postala nezaobilazna sastavnica života modernog društva. Može se opisati kao sposobnost strojeva da donose odluke i uče na način sličan ljudima [12]. Široko je primjenjena grana umjetne inteligencije strojno učenje (ML, *Machine Learning*) koje ima za cilj razumjeti i izgraditi statistički utemeljene algoritme koji imaju sposobnost učiti [12].

Opseg područja primjene umjetne inteligencije eksponencijalno raste. Umjetna se inteligencija danas koristi u proizvodnji, prodaji, industriji, transportu i brojnim drugim djelatnostima, no možda je najvažnija i najplemenitija njezina uporaba u medicini.

Zdravstveni je sektor jedan od onih koji su najviše prisvojili umjetnu inteligenciju i iskoristili njene prednosti.

Dostupnost velikih baza podataka i razvoj naprednih algoritama omogućili su uporabu umjetne inteligencije u medicini za analizu podataka dobivenih slikovnim metodama snimanja, preliminarnu dijagnozu, predlaganje terapije, predviđanje pojave bolesti, otkrivanje novih lijekova, stvaranje, pohranu i analizu dokumentacije, a razvojem mobilnih aplikacija zaživjeli su i brojni virtualni asistenti [12, 13].

Jedna od najčešće korištenih primjena umjetne inteligencije u medicini već je spomenuta analiza podataka dobivenih slikovnim metodama snimanja. Na temelju tih podataka računalo može dati dijagnozu, predvidjeti prognozu i predložiti terapiju. Pritom liječnik iste može prihvatiti, odbaciti ili izmijeniti na temelju okolnosti, vlastitog iskustva i procjene [12, 13].

Uloga liječnika može se opisati trima sastavnicama - percepijom (tumačenjem vizualnih slika i integriranom analizom podataka), spoznajom (kreativnim rješavanjem problema i donošenjem složenih odluka) i zahvatima [13]. Iako se razvija uporaba umjetne inteligencije u sve tri navedene sastavnice i računala su se u zadatku percepcije pokazala višestruko bržima i učinkovitijima od liječnika, liječnici i dalje imaju prednost kada je riječ o spoznaji i zahvatima [13]. Točnost rezultata umjetne inteligencije visoka je, ali nije apsolutna i konačna je odluka i dalje u rukama liječnika.

Cilj je moderne medicine iskoristiti prednosti i potencijale umjetne inteligencije i integrirati ih s radom liječnika kako bi se povećala točnost i učinkovitost dijagnoza, odluka, zahvata, terapija i prognoza.

# 4. PRIMJENA KONVOLUCIJSKIH MREŽA U DETEKCIJI OBJEKATA

Kako računala "vide"? Vid je za ljude najvažnije osjetilo kojim primaju 90% svih informacija iz svoje okoline. Ljudi s lakoćom percipiraju trodimenzionalni svijet oko sebe prepoznajući živa bića i objekte i razlikujući ih od njihove pozadine. Pritom promjena točke gledišta i osvjetljenosti, deformacija objekta, sličnost objekta s pozadinom i različite varijacije objekta u istoj vizualnoj okolini za čovjeka uglavom ne predstavljaju problem u interpretaciji vizualnog podražaja [14, 15]. Međutim, kada je riječ o računalima, proces prepoznavanja objekata daleko je složeniji i prethodno nabrojeni problemi stvaraju veliku prepreku [14, 15]. Ono što je čovjeku vizualni podržaj, za računalo je samo niz brojčanih vrijednosti (piksela).

Ovo poglavlje donosi problematiku računalnog vida i detekcije objekata i opisuje konvolucijske neuronske mreže - u novije vrijeme korišteni alat kada je riječ o klasifikaciji, lokalizaciji i detekciji objekata. Zatim su opisani parametri vrednovanja detekcije objekata konvolucijskim mrežama te skupovi podataka na kojima one uče, a na samom su kraju poglavlja predstavljeni neki od algoritama koji se koriste pri detekciji objekata.

## 4.1. Računalni vid

Računalni vid (CV, *Computer Vision*) interdisciplinarno je područje umjetne inteligencije koje se bavi prikupljanjem, obradom, analiziranjem i razumijevanjem višedimenzionalnih podataka (slika, videozapisa) s ciljem izvlačenja korisnih numeričkih ili simboličkih podataka [16]. Pokriva područja poput detekcije, prepoznavanja i praćenja objekata, semantičke segmentacije i mnogih drugih [16, 17, 18].

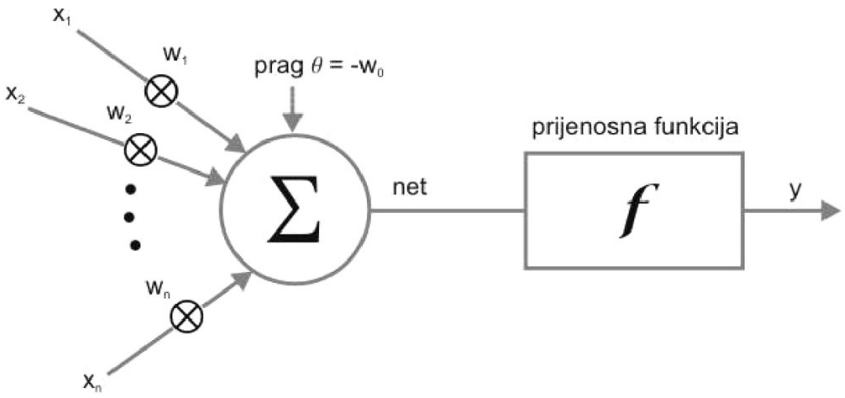
Razvoj ovog područja počeo je u 20. stoljeću, no prepreku su predstavljali malena dostupnost podataka i ograničene sposobnosti računala [14]. Zahvaljujući galopirajućem razvoju tehnologije i Interneta u 21. stoljeću, računalni vid doživio je značajan napredak. Veliki široko dostupni skupovi podataka i razvoj grafičkih procesora (GPU, *Graphics Procesing Unit*) u sinergiji s razvojem programskih jezika i dostupnošću modela dubokog učenja doprinijeli su drastičnom rastu ovog područja [15, 19].

Duboko učenje dio je strojnog učenja nadahnut načinom na koji djeluje ljudski mozak koji koristi neurone i njihove međusobne veze za izvršavanje složenih aktivnosti kao što su govor,

kretanje, razmišljanje, gledanje. Duboko učenje uglavnom podrazumijeva uporabu umjetnih neuronskih mreža.

Umjetne neuronske mreže, ispirirane biologijom, sastavljene su od više slojeva kojima je osnovni element neuron. Biološki se neuron sastoji od dendrita, tijela i aksona, a umjetni, po uzoru na njega, od ulaza, sumatora i aktivacijske funkcije (prijenosne funkcije) i izlaza. Dok su akson jednog biološkog neurona i dendriti drugog povezani sinapsom, izlaz jednog umjetnog neurona i ulaz drugog povezani su težinskim faktorima (utezima, *weights*) koji skaliraju vrijednosti pojedinih ulaza. Kako biološki neuron preko dendrita prima podražaj, odgovara na njega i šalje odgovor preko aksona idućem neuronu u nizu, tako i umjetni neuron na svojim ulazima prima neke numeričke vrijednosti skalirane utezima, zbraja ih i dodaje im pomak (prag, *bias*), a aktivacijska funkcija na to odgovara nekim izlazom koji se prosljeđuje sljedećem neuronu.

Građa jednog umjetnog neurona s $n$ ulaza dana je Slikom 4.1, gdje su $x_1, x_2, \dots, x_n$ ulazi, $w_1, w_2, \dots, w_n$ njima pridruženi utezi, $\theta$ vrijednost praga, $f$ prijenosna funkcija, a $y$ izlaz iz neurona.




*Slika 4.1. Građa umjetnog neurona [20]*

Neuroni istog sloja imaju istu funkciju, ali svaki uči druge parametre. Niz neuronskih slojeva kontinuirano transformira ulazne podatke i preslikava ih u izlazne koristeći unaprijedni proces. Utezi i pomaci pojedinih neurona zatim su optimizirani u procesu unazadne propagacije. Tijekom ovog procesa izračunava se pogreška, razlika između stvarnog i predviđenog izlaza, i propagira se unatrag kroz mrežu pomoću gradijenata koji služe za ažuriranje utega i pomaka. Ovaj se postupak ponavlja iterativno čime se pogreška minimizira i parametri mrežne arhitekture konvergiraju. Na taj način mreža uči optimalne parametre za neurone koji su potrebni za

smisleno predviđanje izlaza. U novije se vrijeme tehnike dubokog učenja primjenjuju i u području računalnog vida u detekciji objekata u područjima autonomne vožnje, detekcije pješaka, medicinskog snimanja, robotskog vida, pametnog video nadzora i drugima.

### 4.2. Detekcija objekata

Detekcija objekata dio je računalnog vida čija je svrha prepoznavanje i klasificiranje objekata ili njihovih dijelova na slikama ili videozapisima [21]. Nastoji obraditi slikovne podatke koristeći razne metode izlučivanja značajki iz slike i klasifikacijske algoritme s ciljem točnog, preciznog i pouzdanog dobivanja korisnih semantičkih i položajnih informacija o slici i otkrivanja korisnih značajki [18].

Na proces detekcije uvelike utječu mnogi vanjski čimbenici kao što su kut slikanja, svjetlost, sjene i oprema što može uzrokovati pojavu izobličenja i šuma na slici. Kako bi se riješio taj problem, posljednjih se godina istražuju različiti pristupi i algoritmi [18].

Tradicionalni algoritmi za detekciju objekata na slikama temelje se na takozvanoj metodi kliznih prozora (*sliding window method*) [14, 15, 18, 22]. Postupak detekcije može se podijeliti u tri koraka. U prvom se koraku generiraju pravokutni klizni prozori različitih veličina i odnosa visine i širine okvira [22]. Prozori se pomiču po slici s lijeva na desno i odozgo prema dolje i na taj se način pokrije većina lokacija na slici [15]. U drugom se koraku izlučuju značajke pojedine regije slike na kojoj se zaustavio klizni prozor [15]. Za izlučivanja značajki često su korištene metode histograma orijentiranog gradijenta (HOG, *Histogram of Oriented Gradient*), transformacije značajki nepromjenjivog mjerila (SIFT, *Scale Invariant Feature Transform*) i lokalnih binarnih uzoraka (LBP, *Local Binary Patterns*) [15, 18]. Izlučene je značajke moguće klasificirati pomoću unaprijed naučenog klasifikatora [22]. U tu svrhu najčešće su korišteni metoda potpornih vektora (SVM, *Support Vector Machines*), AdaBoost (*Adaptive Boosting*) klasifikator i modeli s deformabilnim dijelovima (DPM, *Deformable Part Model*) [15, 18]. Nakon izvršene klasifikacije miču se redundantni prozori i optimiziraju rezultati korištenjem nemaksimalne supresije (NMS, *Non-Maximum Supression*) [15].

Tradicionalne metode detekcije zahtijevaju previše vremena i prostora za pohranu, a rezultiraju niskom robusnošću i točnošću [15]. Iz tog razloga neće biti detaljnije opisivane u ovom radu, već će pažnja biti usmjerena k novijim, bržim, točnijim i pouzdanijim metodama.

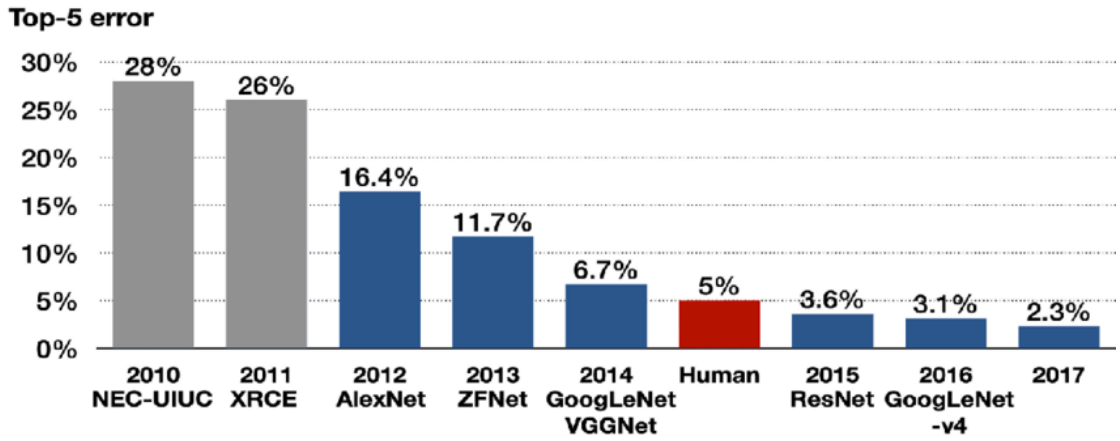
U usporedbi s tradicionalnim algoritmima za detekciju, algoritmi dubokog učenja rezultiraju većom robusnošću, točnošću i brzinom, a posebice kada se radi o detekciji više klasa objekata [23].

### 4.3. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (CNN, *Convolutional Neural Network*) posebna su vrsta umjetnih unaprijednih neuronskih mreža čija je svrha pretprocesiranje nestrukturiranih podataka kao što su pikseli, audio podaci, voxel podaci i slično [24]. Inspirirane su načinom na koji se u vizualnom korteksu mozga obrađuju značajke vizualnog podražaja [14, 25]. Danas je to jedina korištena metoda kada je riječ u klasifikaciji i semantičkoj segmentaciji slika, detekciji objekata, prepoznavanju aktivnosti i drugim zadacima računalnog vida [19].

Prve konvolucijske neuronske mreže nastale su 90-ih godina 20. stoljeća [14, 25]. Prva poznata konvolucijska mreža LeNet autora Yanna LeCuna nastala je 1998. i služila je prepoznavanju ručno pisanih slova i znamenki [14, 22, 25, 26]. Nakon toga je, zbog nedovoljnih računalnih sposobnosti kao i nedostatka podataka za učenje, razvoj konvolucijskih neuronskih mreža neko vrijeme stagnirao sve do 2012. kada su natjecatelji koji su koristili konvolucijsku neuronsku mrežu pobijedili na ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) natjecanju [14, 15, 19, 25, 27]. ILSVRC je natjecanje u kojem se ocjenjuju različiti algoritmi detekcije objekata i klasifikacije slika učeni na istom skupu podataka - ImageNetu [14, 15, 19, 25, 27, 28]. Prvih godina održavanja izazova kvaliteta klasifikacije nije se značajno povećavala sve dok 2012. godine Alex Krizhevsky, Ilya Sutskever i Geoffrey Hinton nisu upotrijebili konvolucijsku neuronsku mrežu SuperVision, danas poznatiju kao AlexNet [14, 15, 19, 25, 27]. AlexNet je rezultirao oko 10% manjom pogreškom od prošlogodišnjeg pobjednika i nakon njegove pojave kreće ubrzani razvoj konvolucijskih neuronskih mreža [14, 29]. Već sljedeće 2013. godine pet najboljih natjecatelja koristilo je konvolucijsku neuronsku mrežu, a od 2014. godine konvolucijsku neuronsku mrežu koristili su svi natjecatelji [24]. Slika 4.2 prikazuje usporedbu rezultata pogreške pobjedničkih rješenja ILSVRC natjecanja od 2010. do 2017. godine međusobno i s pogreškom čovjeka.

U nastavku potpoglavlja detaljnije će biti opisana struktura konvolucijske neuronske mreže, proces učenja, parametri vrednovanja detekcije objekata korištenjem konvolucijske neuronske kao i najčešće korišteni skupovi podataka u postupku učenja mreže.

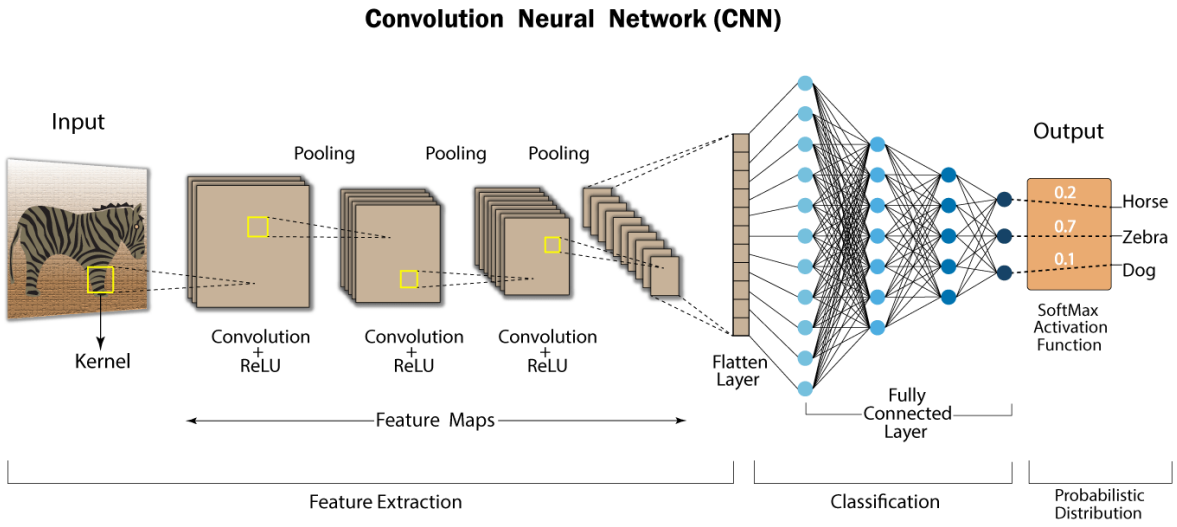



*Slika 4.2. Rezultati pogreške pobjedničkih rješenja ILSVRC natjecanja od 2010. do 2017. godine u usporedbi s pogreškom čovjeka [29]*

### 4.3.1. Struktura konvolucijske neuronske mreže

Konvolucijska neuronska mreža sastoji se od nekoliko vrsta slojeva, a osnovne su vrste ulazni sloj, konvolucijski sloj, aktivacijski (nelinearni) sloj, sloj sažimanja i potpuno povezani sloj [14, 22, 27]. Najčešće se nakon ulaznog sloja nekoliko puta izmjenjuju konvolucijski sloj, aktivacijski sloj i sloj sažimanja, a zatim slijede potpuno povezani slojevi koji daju izlaz iz mreže [15, 27].

Primjer strukture jednostavne konvolucijske neuronske mreže koja služi za klasifikaciju objekata dan je Slikom 4.3.




*Slika 4.3. Struktura konvolucijske neuronske mreže za klasifikaciju objekata [30]*

Slijedi opis svakog od prije spomenutih slojeva konvolucijske neuronske mreže.

Ulazni sloj (*Input Layer*) predstavlja samu sliku koju mreža vidi kao niz piksela [14]. Ukoliko se radi o monokromatskoj slici, dimenzije slike će biti $N \times M \times 1$, ili jednostavnije $N \times M$, gdje $N$ predstavlja visinu slike, a $M$ širinu slike. Budući da se radi o jednokanalnoj slici, treću dimenziju nije potrebno pisati već se slika može prikazati kao matrica piksela (tenzor drugog reda). Ukoliko je riječ o višekanalnoj slici $N \times M \times D$, gdje $N$ predstavlja visinu slike, $M$ širinu slike, a D dubinu slike (broj kanala), slika je tenzor višeg reda. Kada je riječ o slici u boji, odnosno trokanalnoj slici, njene će se dimenzije zapisivati kao $N \times M \times 3$, gdje broj 3 označava tri kanala slike - crveni, zeleni i plavi, odnosno RGB (*Red, Green, Blue*). Ovako zapisani pikseli mogu se promatrati kao tenzor trećeg reda.

Konvolucijski sloj (*convolutional layer*) osnovni je gradivni blok konvolucijske neuronske mreže u kojem se izvršava konvolucija ulazne slike i nekog filtra (kernela) kako bi se dobila mapa značajki (*feature map*) [14, 22, 27]. Dio slike na koji se primjenjuje filtar naziva se receptivnim poljem (*receptive field*).

Filtar je obično manje visine i širine nego sama slika, dok mu je dubina jednaka dubini slike. Pomiče se po slici s određenim korakom (*stride*) i pritom se računa skalarni produkt slike i filtra [14, 24].

Dimenzije dobivene mape značajki dane su izrazom 4.1

$$Z_H \times Z_W = \left(\frac{N - F_H}{S} + 1\right) \times \left(\frac{M - F_W}{S} + 1\right), \qquad (4.1)$$

gdje je $Z_H$ visina mape značajki, $Z_W$ širina mape značajki, $N$ visina slike, $M$ širina slike, $F_H$ visina filtra, $F_W$ širina filtra, a $S$ korak. Dubina jedne mape značajki uvijek je jednaka 1.

Budući da je filtar najčešće kvadratnog oblika, dimenzija $F \times F$, izraz 4.1 može se zapisat izrazom 4.2

$$Z_H \times Z_W = \left(\frac{N - F}{S} + 1\right) \times \left(\frac{M - F}{S} + 1\right). \qquad (4.2)$$

Visina i širina mape značajki uvijek su manje od visine i širine ulazne slike [14]. Kroz nekoliko konvolucijskih slojeva na taj se način mape značajki sve više smanjuju i moguće je izgubiti vrijedne informacije o slici. Stoga se nerijetko upotrebljava tzv. popunjavanje (*padding*) slike [14]. Najčešće je korišteno popunjavanje nulama (*zero padding*) kojim se na rubove slike dodaju nule [14]. Na taj način sprječava se naglo smanjivanje dimenzija mapa značajki kroz konvolucijske slojeve i čuvaju se potencijalno korisni podaci na rubovima slika.

Uključujući popunjavanje, izraz 4.2 može se proširiti u izraz 4.3

$$Z_H \times Z_W = \left(\frac{N - F + 2P}{S} + 1\right) \times \left(\frac{M - F + 2P}{S} + 1\right), \qquad (4.3)$$

gdje $P$ predstavlja broj slojeva nula koji se dodaju na rubove slike.

Monokromatska ulazna slika dimenzija $N \times M$ može se zapisati u obliku matrice $\boldsymbol{X}$ dane izrazom 4.4

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,M} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,M} \end{bmatrix}, \qquad (4.4)$$

gdje su $x_{1,1}, x_{1,2}, \ldots, x_{N,M}$ vrijednosti piksela slike.

Filtar dimenzija $F_H \times F_w$ koji će se primijeniti na ulaznoj slici može se zapisati u obliku matrice $\boldsymbol{F}$ dane izrazom 4.5

$$\boldsymbol{F} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,F_H} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,F_H} \\ \vdots & \vdots & \ddots & \vdots \\ w_{F_W,1} & w_{F_W,2} & \cdots & w_{F_W,F_H} \end{bmatrix}, \qquad (4.5)$$

gdje su $w_{1,1}$, $w_{1,2}, \ldots, w_{F_W,F_H}$ vrijednosti težinskih faktora. Vrijednosti težinskih faktora filtra optimiziraju se procesom učenja mreže o čemu će više riječi biti kasnije.

Kako je već rečeno, konvolucija ulazne slike i filtra rezultira mapom značajki $\boldsymbol{Z}$ dimenzija $Z_H \times Z_W$ danom izrazom 4.6

$$\boldsymbol{Z} = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,Z_H} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,Z_H} \\ \vdots & \vdots & \ddots & \vdots \\ z_{Z_W,1} & z_{Z_W,2} & \cdots & z_{Z_W,Z_H} \end{bmatrix}, \qquad (4.6)$$

gdje su $z_{1,1},\ z_{1,2}, \ldots, z_{Z_W,Z_H}$ vrijednosti mape značajki.

Svaki korak konvolucije rezultira jednom vrijednosti u mapi značajki. Konvolucija se može zapisati izrazom 4.7.

$$z_{i,j} = \sum_{a=0}^{a=F_H-1} \sum_{b=0}^{b=F_W-1} w_{a+1,b+1} x_{i+a,j+b}, \qquad (4.7)$$

gdje je $z_{i,j}$ element mape značajki u $i$-tom retku i $j$-tom stupcu.

Produkt filtra i ulazne slike moguće je po potrebi uvećati za neku vrijednost pomaka (*bias*). Kao i težinski faktori, vrijednost pomaka optimizira se u postupku učenja mreže [14].

Ako se na ulaznu sliku u jednom konvolucijskom sloju primjenjuje više filtara, dobit će se onoliko mapa značajki koliko je filtara bilo primijenjeno. Broj mapa značajki predstavlja njihovu dubinu [14].

Svaki je primjenjeni filtar zaslužan za izlučivanje drugih značajki slike. Početni filtri uglavnom razlučuju jednostavnije uzorke kao što su rubovi, filtri u sredini konvolucijske mreže razlučuju uzorke kao što su kutovi, a krajnji filtri služe za razlučivanje kompleksnih uzoraka [14, 25].

Zaključno, konvolucijski sloj zahtijeva četiri hiperparametra - broj filtara $K$, prostornu dimenziju filtra $F$, korak (*stride*) $S$ i količinu popunjavanja nulama (*zero padding*) $P$ [14].

Broj parametara konvolucijskog sloja koje je potrebno naučiti može se izračunati izrazom 4.18

$$N_{CL\_parameters} = K(F^2 \cdot D + 1), \qquad (4.18)$$

gdje $K$ predstavlja broj filtara, $F^2 \cdot D$ predstavlja broj težinskih faktora u svakom filtru, a pribrojena jedinica pomak koji se dodaje svakom filtru.

Konačan broj parametara dobiven u mapama značajki dan je izrazom 4.9

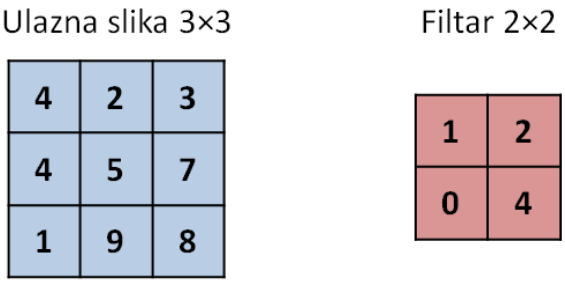
$$N_{fm\_parameters} = K \cdot Z_H \cdot Z_W, \quad (4.9)$$

gdje je $N_{fm\_parameters}$ broj parametara mapa značajki, $K$ broj filtara primijenjenih na sliku, $Z_H$, a $Z_W$ širina mape značajki.

Slijedi jednostavan numerički primjer u kojem je prikazana konvolucija monokromatske slike i filtra.

**Primjer 4.1.** ***Konvolucija proizvoljne monokromatske slike dimenzija*** $\boldsymbol{N \times M = 3 \times 3}$ ***i proizvoljnog filtra dimenzija*** $\boldsymbol{F_H \times F_w = 2 \times 2}$ ***s korakom*** $\boldsymbol{S = 2}$ ***i bez popunjavanja nulama.***

*Slika 4.4 prikazuje zadanu monokromatsku sliku dimenzija 3×3 proizvoljnih vrijednosti piksela, i proizvoljni filtar s kojim će se vršiti konvolucija slike.*




*a)* *b)*

*Slika 4.4. a) proizvoljna monokromatska slika; b) proizvoljni filtar*

*Dimenzije izlazne mape značajki prema izrazu 4.2 bit će*

$$Z_H \times Z_W = \left(\frac{3-2}{1}+1\right) \times \left(\frac{3-2}{1}+1\right) = 2 \times 2. \quad (4.10)$$

*Pojedine elemente mape značajki računa se korištenjem izraza 4.7. Izračun elemenata dan je redom izrazima 4.11*

$$z_{1,1} = \sum_{a=0}^{a=F_H-1} \sum_{b=0}^{b=F_W-1} w_{a+1,b+1} x_{1+a,1+b} = 4 \cdot 1 + 2 \cdot 2 + 4 \cdot 0 + 5 \cdot 4 = 28\,, \quad (4.11)$$

*4.12*

$$z_{1,2} = \sum_{a=0}^{a=F_H-1} \sum_{b=0}^{b=F_W-1} w_{a+1,b+1} x_{1+a,2+b} = 2 \cdot 1 + 3 \cdot 2 + 5 \cdot 0 + 4 \cdot 4 = 36, \qquad (4.12)$$

*4.13*

$$z_{2,1} = \sum_{a=0}^{a=F_H-1} \sum_{b=0}^{b=F_W-1} w_{a+1,b+1} x_{2+a,1+b} = 4 \cdot 1 + 5 \cdot 2 + 1 \cdot 0 + 9 \cdot 4 = 50 \qquad (4.13)$$

*i 4.14*

$$z_{2,2} = \sum_{a=0}^{a=F_H-1} \sum_{b=0}^{b=F_W-1} w_{a+1,b+1} x_{2+a,2+b} = 5 \cdot 1 + 7 \cdot 2 + 9 \cdot 0 + 8 \cdot 4 = 51. \qquad (4.14)$$

*Vizualani prikaz izračuna spomenutih članova dan je Slikom 4.5.*

Nelinearni ili aktivacijski sloj slijedi nakon konvolucijskog sloja. Transformira ulazni signal, u ovom slučaju mape značajki, u neku izlaznu vrijednost koja se može koristiti u sljedećem sloju [22]. Pritom volumne značajke (dimenzije) ostaju očuvane [27]. Najčešće korištene aktivacijske funkcije su sigmoidalna, tangens hiperbolna (tanh), ReLU, LReLU, ELU, SiLU i slično [27].

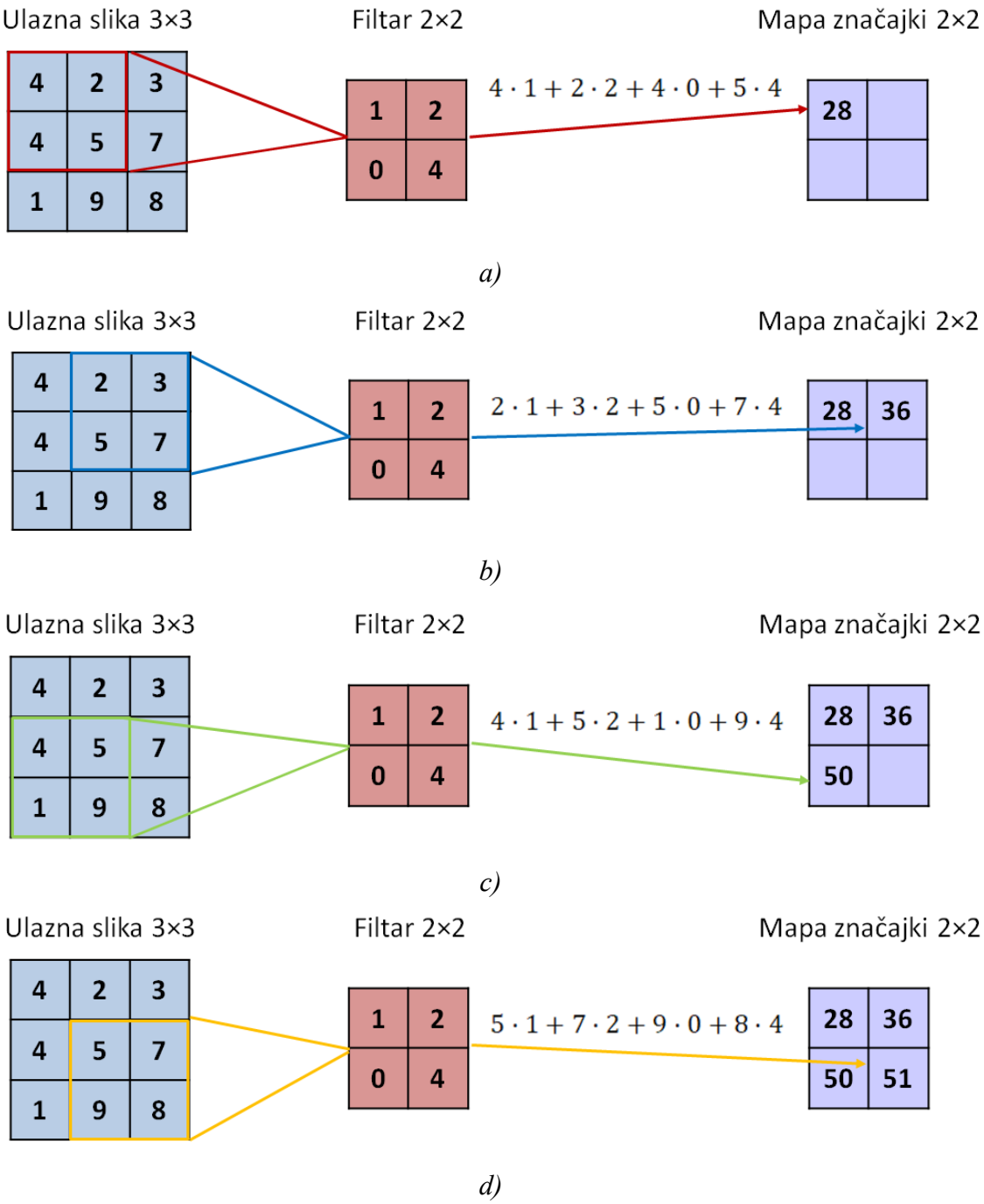
Sigmoidalna aktivacijska funkcija logistička je funkcija koja se često koristi u postupcima učenja neuronske mreže. Dana je Slikom 4.6 i izrazom 4.15

$$f(x) = \frac{1}{1 + e^{-x}}, \qquad (4.15)$$

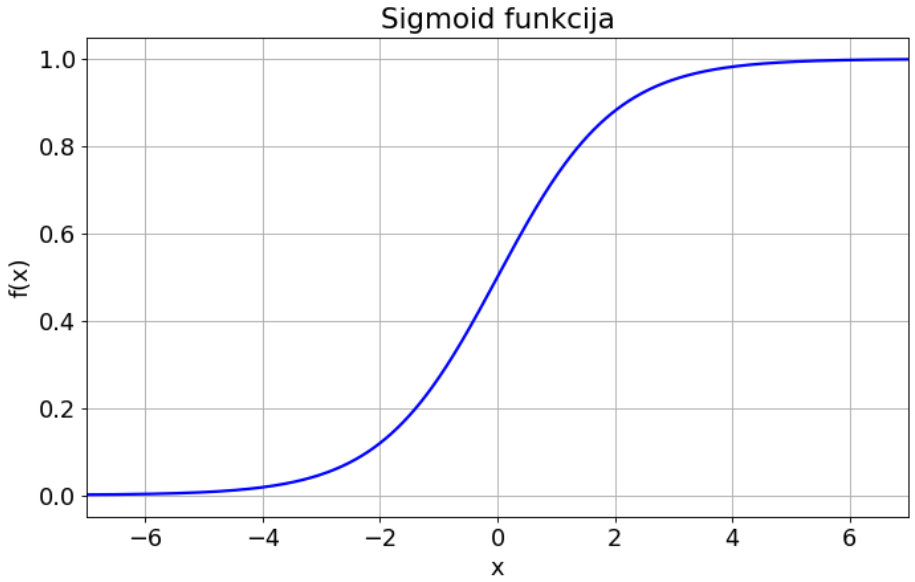
gdje je $x$ ulaz, a $f(x)$ aktivacijska funkcija.

Sigmoidalna funkcija derivabilna je na području cijele domene i na svom izlazu daje vrijednost iz intervala $[0, 1]$. Nedostatak sigmoidalne funkcije je u tome što može ući u zasićenje pa joj je u tim područjima vrijednost derivacije (gradijent) jednaka nuli [22, 24].

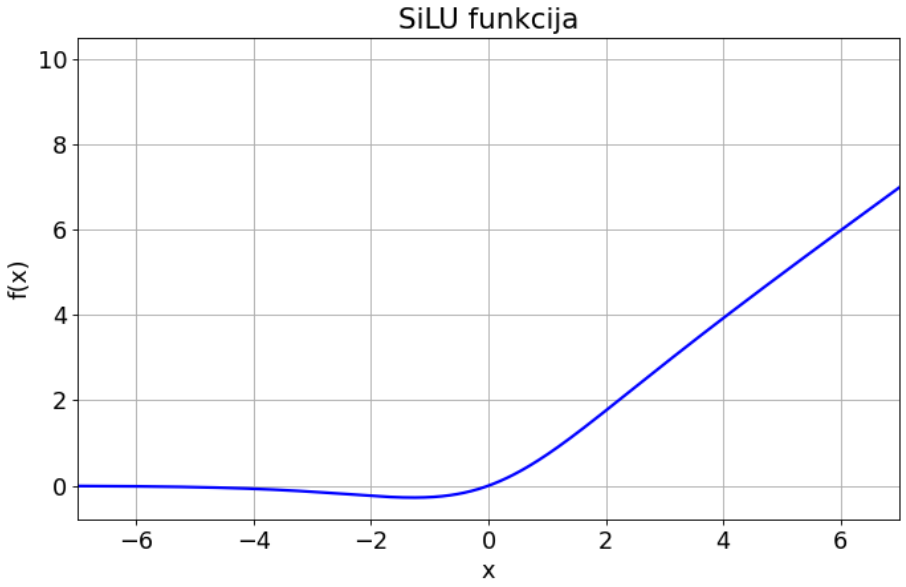
Posebna je modifikacija ove funkcije sigmoidalna linearna jedinična funkcija ili SiLU (*Sigmoidal Linear Unit*) dana Slikom 4.7.




*Slika 4.5. Izračun elementa: a) $z_{1,1}$; b) $z_{1,2}$; c) $z_{2,1}$; d) $z_{2,2}$*




*Slika 4.6. Sigmoidalna funkcija*

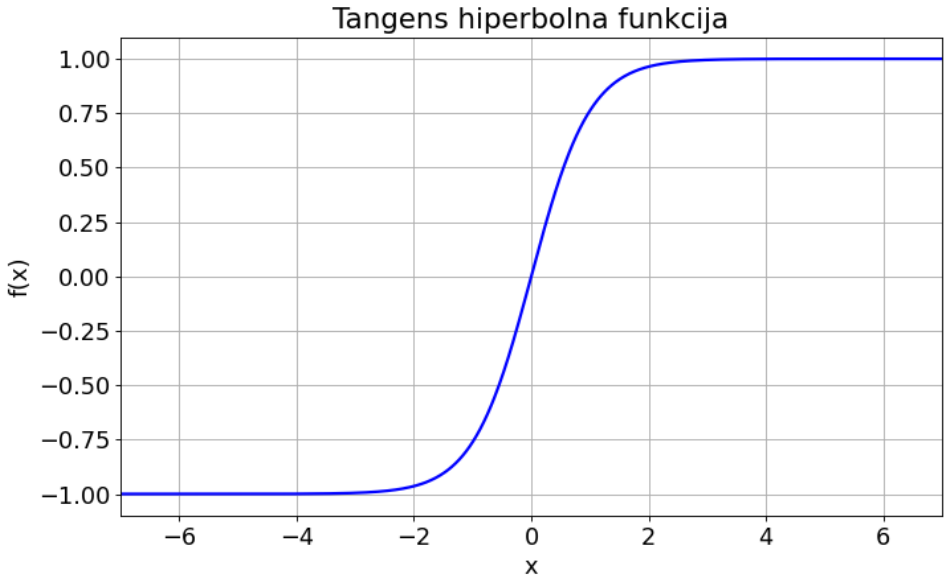



*Slika 4.7. SiLU funkcija*

Tangens hiperbolna funkcija vrlo je slična sigmoidalnoj funkciji. Dana je Slikom 4.8 i izrazom 4.16

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1, \qquad (4.16)$$

gdje je $x$ ulaz, a $f(x)$ aktivacijska funkcija.




*Slika 4.8. Tangens hiperbolna funkcija*

Na izlazu daje vrijednost iz intervala $[-1, 1]$, a najčešće se koristi za klasifikaciju između dvije klase. Ima isti nedostatak kao i sigmoidalna funkcija, iznos derivacije u području zasićenja jednak joj je 0 [24].

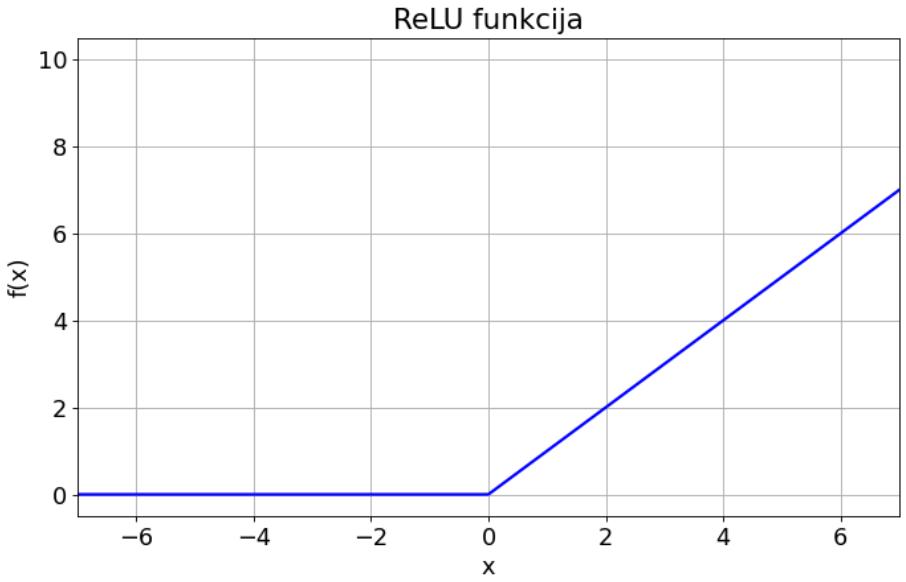
Ispravljena linearna jedinična funkcija, odnosno ReLU (*Rectifier Linear Unit*) zbog svoje se nelinearnosti često koristi kao aktivacijska funkcija u neuronskim mrežama [24]. Vrijednost njezinog gradijenta uvijek je u intervalu $[0, 1]$, a dana je Slikom 4.9 i izrazom 4.17

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}, \qquad (4.17)$$

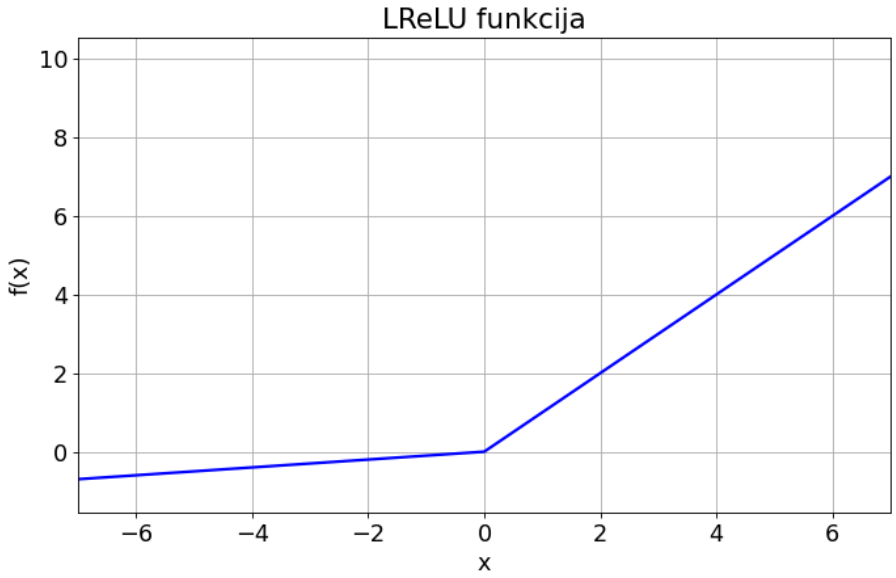
gdje je $x$ ulaz, a $f(x)$ aktivacijska funkcija.

ReLU funkcija dodatno je modificirana na više načina u svrhu boljeg učenja i detekcije pa tako nastaju funkcije kao što su *leaky* ReLU ili LReLU (Slika 4.10) i eksponencijalna linearna jedinična funkcija ili ELU (Slika 4.11).

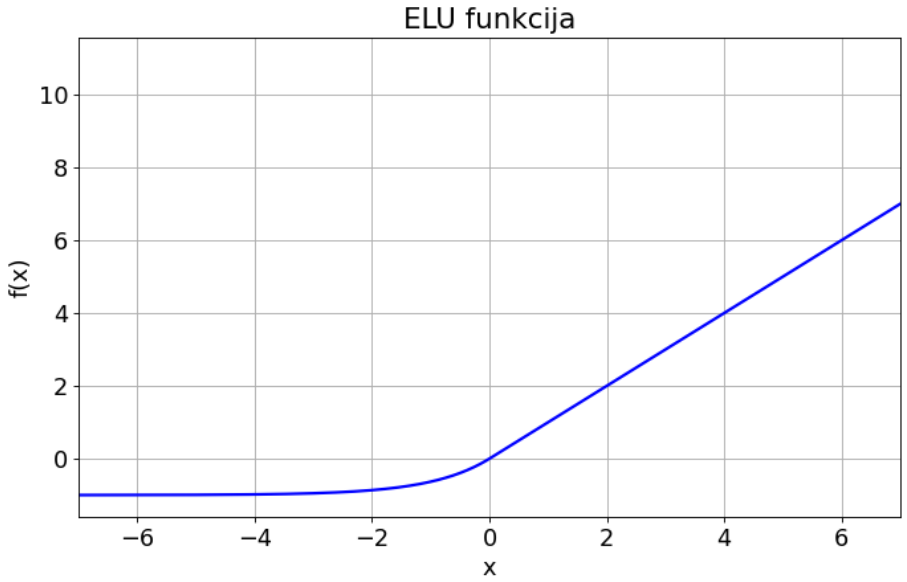
Sloj sažimanja (*pooling layer*) sljedeći je sloj u konvolucijskoj neuronskoj mreži [15]. Sažimanje se koristi da bi se spriječilo preučenje (*overfitting*) mreže, kao i kod podataka visokih rezolucija kako bi se smanjio broj parametara, a time i izračuna, u mreži uz zadržavanje svih bitnih informacija [15, 22, 27].




*Slika 4.9. ReLU funkcija*

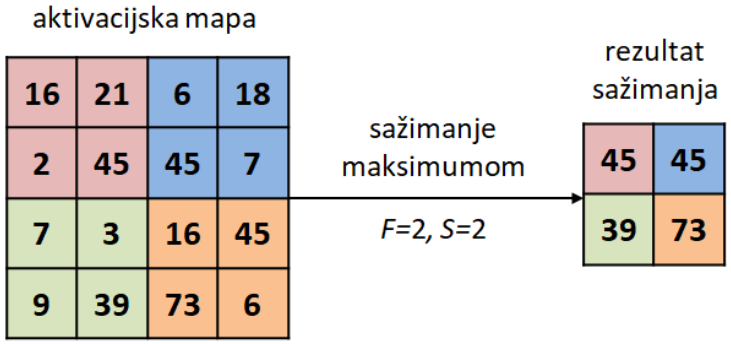



*Slika 4.10. LReLU funkcija*

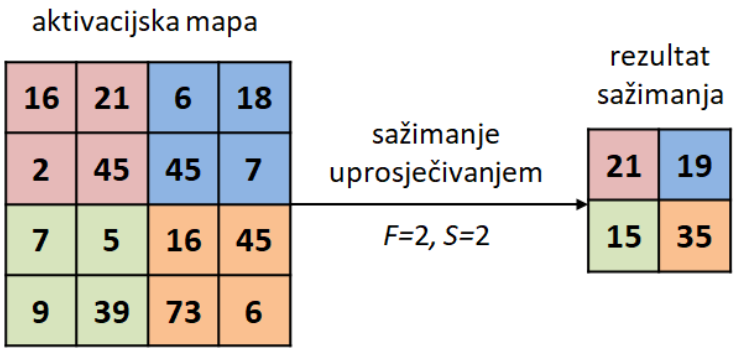



*Slika 4.11. ELU funkcija*

Ulaz u sloj sažimanja aktivacijska je mapa značajki iz prethodnog sloja. Sloj sažimanja također je jedna vrsta filtra koja izvršava sažimanje na receptivnom polju mape [14]. Najčešće primjenjivane metode sažimanja su sažimanje maksimumom (*max pooling*) i sažimanje uprosječivanjem (*average pooling*) prikazani na Slici 4.12 [24].




*a)*




*b)*

*Slika 4.12. Sažimanje: a) maksimumom; b) uprosječivanjem*

Potpuno povezani sloj (*Fully Connected Layer*) zadnji je sloj u mreži koji služi za klasifikaciju objekata i regresiju okvira [22]. On u suštini predstavlja potpuno povezanu neuronsku mrežu u kojoj su svi neuroni povezani na sve neurone iz prethodnog sloja što rezultira velikim brojem parametara - težinskih faktora i pomaka [22].

Potpuno povezani slojevi na svom ulazu traže jednodimenzionalni vektor pa se aktivacijske mape iz prethodnih slojeva pretvaraju u jedan vektor (npr. skup mapa značajki dimenzija $32 \times 32 \times 3$ pretvara se u vektor dimenzija $3072 \times 1$) [14]. Utezi potpuno povezanog sloja matrično se množe s vrijednostima ulaznog vektora i dodaje im se pomak kao što je dano izrazom 4.18

$$\mathbf{s} = \mathbf{W}^{\mathrm{T}}\mathbf{x} + \mathbf{b}, \qquad (4.18)$$

gdje je **s** vektor nenormaliziranih izlaznih vrijednosti, **W** matrica utega potpuno povezanog sloja, **x** ulazni vektor u potpuno povezani sloj, a **b** vektor pomaka potpuno povezanog sloja.

Primjerice, u klasifikaciji objekata za ulazni vektor dimenzija $3072 \times 1$, matrica **W** bit će dimenzija $3072 \times C$, pri čemu je $C$ broj klasa objekata koje model može detektirati, a vektor pomaka **b** i izlaz **s** bit će dimenzija $C \times 1$. Ovako dobiveni rezultati reprezentiraju linearni klasifikator koji izvršava klasifikaciju uglavnom na način da pretpostavi onu klasu kojoj pripada najveća vrijednost u vektoru **s**.

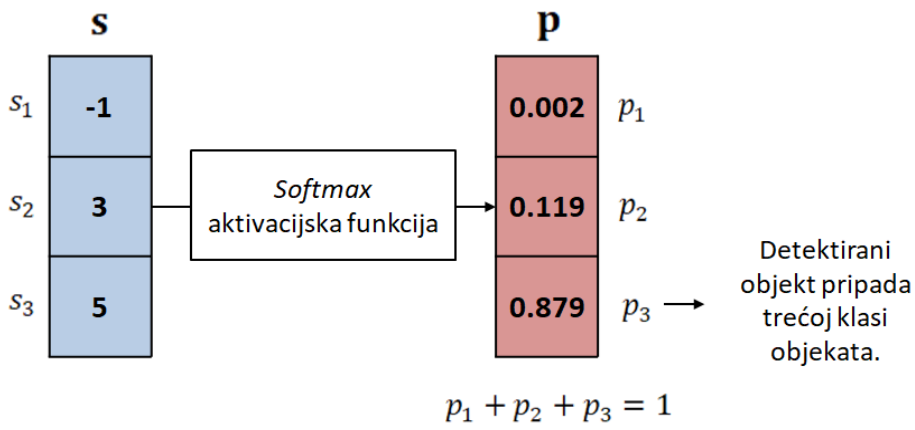
Ipak, u konvolucijskim se neuronskim mrežama češće za klasifikaciju koristi aktivacijska funkcija. Za binarnu klasifikaciju koristi se sigmoidalna aktivacijska funkcija dok se za višeklasnu klasifikaciju koristi normalizirana eksponencijalna aktivacijska funkcija poznatija kao *softmax* funkcija [22, 27].

*Softmax* funkcija detektiranom objektu pridružuje vjerojatnost pripadnosti pojedinoj klasi objekata u obliku vektora dimenzija $C \times 1$, a ukupan zbroj svih vjerojatnosti jednak je jedan. *Softmax* aktivacijska funkcija dana je izrazom 4.19

$$\sigma(\mathbf{s})_{\mathrm{i}} = \frac{\mathrm{e}^{s_{\mathrm{i}}}}{\sum_{j=1}^{C} \mathrm{e}^{s_{\mathrm{j}}}}, \qquad (4.19)$$

gdje je **s** vektor nenormaliziranih izlaznih vrijednosti, $s_i$ je vrijednost pridružena $i$-toj klasi objekata, $s_j$ je vrijednost pridružena $j$-toj klasi objekata, a $C$ je ukupan broj klasa objekata [14, 22].

Primjer kako *softmax* aktivacijska funkcija određuje vjerojatnost pripadnosti pojedinim klasama objekata dan je Slikom 4.13, gdje za ukupno tri moguće klase objekata $s_1$, $s_2$ i $s_3$ predstavljaju vrijednosti pridružene prvoj, drugoj i trećoj klasi objekata, redom, a $p_1$, $p_2$ i $p_3$ vjerojatnost pripadnosti detektiranog objekta prvoj, drugoj i trećoj klasi objekata, redom. I dalje vrijedi da će detektiranom objektu biti pridružena ona klasa s najvećom pripadnom vrijednosti, no ovaj put je ta vrijednost izražena kao vjerojatnost.




*Slika 4.13. Određivanje pripadnosti detektiranog objekta pojedinoj klasi korištenjem softmax aktivacijske funkcije*

*Softmax* funkcija ne služi samo za klasifikaciju, već igra i značajnu ulogu u procesu učenja mreže, o čemu će više riječi biti u nastavku.

### 4.3.2. Učenje konvolucijske neuronske mreže

Svojstvo konvolucijski neuronskih mreža koje ih izdvaja od tradicionalnih metoda detekcije objekata sposobnost je učenja parametara mreže kako bi se optimizirao učinak. Konvolucijske neuronske mreže uče pod nadzorom.

Učenje pod nadzorom (*supervised learning*), poznato i kao učenje s učiteljem, klasičan je postupak učenja umjetnih neuronskih mreža koji se odvija na način da se mreži daju parovi ulaza i pripadajućih izlaza. Mreža postupkom unaprijedne propagacije (*forward propagation*) za dobiveni ulaz predviđa neki izlaz [14]. Potom uspoređuje dobiveni izlaz sa stvarnim izlazom i

procesom unazadne propagacije (*backward propagation*) osvježava svoje parametre, utege i pomake [14, 15, 24]. Proces se iterativno ponavlja sve dok mreža ne nauči one parametre za koje je razlika između predviđenog i stvarnog izlaza minimalna.

Za praćenje koliko dobre predikcije daje mreža tijekom treninga koristi se funkcija gubitka (*loss function*) $L$ [14]. Nekad se naziva i funkcijom pogreške, rizika ili troška.

Gubitak na cijelom skupu podataka općenito se definira izazom 4.20

$$L = \frac{1}{N}\sum_{i} L_i(f(x_i, W), y_i) + \lambda R(W), \quad (4.20)$$

gdje $N$ predstavlja ukupan broj primjera za učenje koje je dobila mreža, $L_i$ predstavlja gubitak predikcije $i$-tog izlaza, $f(x_i, W)$ je funkcija koja daje predikciju klase objekata na temelju $i$-tog ulaza $x_i$ i trenutnih parametara mreže $W$, a $y_i$ predstavlja stvarnu vrijednost pridruženu ulazu $x_i$. Funkcija $R(W)$ je funkcija regularizacije skalirana konstantom $\lambda$.

Funkcija regularizacije dodaje se gubitku kako bi se mreža "kaznila" za korištenje složenijih modela [14]. Uvođenje složenijih parametara i funkcija u model može davati dobar rezultat na skupu podataka za učenje, ali loš kada se mreža primjenjuje na nekom drugom skupu podataka. Stoga, ako mreža želi smanjiti vrijednost funkcije gubitka, morat će smanjiti i vrijednost funkcije $R(W)$ tako što će koristiti jednostavnije parametre.

Postoje različite funkcije gubitaka koje se mogu koristiti pri učenju konvolucijske neuronske mreže, a najčešće korištene su srednja kvadratna pogreška i unakrsna entropija [24].

Srednja kvadratna pogreška ili MSE (*Mean Square Error*) jednostavna je i često korištena funkcija gubitka dana izrazom 4.21

$$L_{MSE}(W, b) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \quad (4.21)$$

gdje su $W$ težinski faktori mreže, $b$ pomaci, $N$ broj parova ulaz-izlaz koje je mreža primila, $\hat{y}_i$ vrijednost predikcije $i$-tog izlaza, a $y_i$ stvarna vrijednost $i$-tog izlaza.

Srednja kvadratna pogreška koristi se uglavnom u regresijskim problemima i kada ciljana varijabla dolazi iz normalne razdiobe. U klasifikacijskim problemima koristi se uglavnom za binarnu klasifikaciju. Kod višeklasne klasifikacije nije dobro mjerilo budući da ne podržava vjerojatnosnu razdiobu [14, 15, 24].

Unakrsna entropija ili CE (*Cross Entropy*) računa se pomoću negativne logaritamske izglednosti ili NLL-a (*Negative Log Likelihood*). Najčešće je korištena u problemima klasifikacije, a posebice kada je riječ o višeklasnoj klasifikaciji. Primjenjuje se na rezultatu koji je zapisan u obliku vrijednosti vjerojatnosne razdiobe [14, 15, 22, 24]. Računa se pomoću pojednostavljene formule dane izrazom 4.22

$$L_{CE}(W, b) = -\sum_{i=1}^{C} \log\left(P(\hat{y}_i = y_i | x_i)\right), \quad (4.22)$$

gdje su $W$ težinski faktori mreže, $b$ pomaci, $C$ broj klasifikacijskih kategorija, $x_i$ je ulaz $i$-te klase, $\hat{y}_i$ vrijednost predikcije $i$-te klase, $y_i$ stvarna vrijednost $i$-te klase, a $P(\hat{y}_i = y_i | x_i)$ je vjerojatnost da je predviđeni izlaz jednak stvarnom izlazu.

Ako se za klasifikaciju koristila *softmax* funkcija, 4.22 se može zapisati izrazom 4.23

$$L_{CE}(W, b) = -\sum_{i=1}^{C} \log\left(\frac{e^{s_{y_i}}}{\sum_{j=1}^{C} e^{s_j}}\right), \quad (4.23)$$

gdje su $W$ težinski faktori mreže, $b$ pomaci, $C$ broj klasifikacijskih kategorija koje je mreža primila, $s_{y_i}$ je vrijednost pridružena predikciji stvarnog izlaza $i$-te klase $y_i$, a $s_j$ je vrijednost pridružena predikciji izlaza $j$-te klase.

Cilj je učenja minimizirati vrijednosti funkcije gubitaka što mreža radi metodom unazadne propagacije i primjenom nekog od optimizacijskih algoritama.

Unazadna propagacija (*backpropagation*) omogućuje rekurzivnu propagaciju gubitka unatrag kroz mrežu i iterativno osvježavanje parametara mreže kako bi oni postali optimalni. Za konvolucijske neuronske mreže najvažniji su parametri konvolucijskih filtara, sloja sažimanja i potpuno povezanog sloja te je cilj mreže naučiti poveznicu između dobivene pogreške i pojedinih parametara u mreži i optimizirati ih kako bi se pogreška minimizirala [15]. Za to se u konvolucijskim neuronskim mrežama najčešće koristi optimizacijski algoritam gradijentni spust.

Gradijentni spust (*gradient descent*) optimizacijski je algoritam koji se svodi na pronalaženje minimuma funkcije gubitka korištenjem parcijalnih derivacija po varijablama težinskih faktora i pomaka te se koristi u unazadnoj propagaciji za optimiziranje istih [14, 15, 24].

Osvježavanje utega i pomaka izvršava se kroz unazadnu propagaciju i pritom značajnu ulogu igra stopa učenja $\eta$. Stopa učenja (*learning rate*) važan je parametar u neuronskim mrežama. Ona nadzire snagu unazadne propagacije te time upravlja brzinom učenja - konvergencijom modela [15]. Odabir optimalne stope učenja od ključne je važnosti kako bi mreža brzo učila uz niske gubitke. Ukoliko je odabrana stopa učenja prevelika, model će učiti prebrzo i na kraju divergirati, a ukoliko je premalena konvergirat će jako sporo.

Utezi i pomaci osvježavaju se pomoću izraza 4.24 i 4.25, redom

$$W_i = W_{i(old)} - \eta \frac{\partial L(W,b)}{\partial W_i}, \quad (4.24)$$

$$b_i = b_{i(old)} - \eta \frac{\partial L(W,b)}{\partial b_i}, \quad (4.25)$$

gdje $W_i$ predstavlja težinske faktore $i$-tog sloja, a $W_{i(old)}$ njihove stare vrijednosti, $\frac{\partial L(W,b)}{\partial W_i}$ je gradijent funkcije gubitka u ovisnosti o težinskim parametrima $W_i$, $b_i$ predstavlja pomake $i$-tog sloja, a $b_{i(old)}$ njihove stare vrijednosti, $\frac{\partial L(W,b)}{\partial b_i}$ je gradijent funkcije gubitka u ovisnosti o parametrima pomaka $b_i$, a $\eta$ je stopa učenja.

Kod višeslojne mreže javlja se problem kod izračuna gradijenta budući da je funkcija gubitka rezultat dobiven unaprijednom propagacijom ulaznih podataka kroz mnoge transformacijske slojeve od kojih svaki ima svoje parametre. Za učenje tih parametara gradijent se računa lančanim pravilom kao što je dano izrazom 4.26

$$\frac{\partial L}{\partial w_{ab}} = \sum_i \sum_j \frac{\partial L}{\partial z_{ij}} \frac{\partial z_{ij}}{\partial w_{ab}}, \quad (4.26)$$

gdje je $\frac{\partial L}{\partial w_{ab}}$ parcijalna derivacija funkcije gubitka po težinskom faktoru $a$-tog retka i $b$-tog stupca matrice parametara, a $z_{ij}$ element mape značajki u $i$-tom retku i $j$-tom stupcu [14, 24].

Ako se izraz 4.7 uvrsti u izraz 4.26, on postaje

$$\frac{\partial L}{\partial w_{ab}} = \sum_i \sum_j \frac{\partial L}{\partial z_{ij}} \, x_{i+a,j+b} \quad (4.27)$$

gdje je $x_{i+a,j+b}$ element ulaza u filtar u $(i+a)$-tom retku i $(j+b)$-tom stupcu.

Iterativnim ponavljanjem opisanog postupka unazadne propagacije, mreža uči optimalne parametre.

Postupak traje sve dok model ne uđe u stagnaciju, zadovolji granične vrijednosti unutar kojih mora biti rješenje ili se izvrši unaprijed zadan broj iteracija.

U postupku učenja potrebno je obratiti pažnju na podučenje (*underfitting*) i preučenje (*overfitting*).

Podučenje mreže (*underfitting*) pojam je koji opisuje model koji ne daje dovoljno dobre rezultate niti na skupu podataka za učenje, niti na skupu podataka za ispitivanje. Podučenje se javlja iz više razloga, a neki su od mogućih prisutnost šuma u podacima za učenje, nedovoljan broj podataka za učenje, prekratko učenje i previše jednostavan model [31]. Podučenje uglavnom ne predstavlja problem jer se lako uočava i ispravlja uklanjanjem šuma iz podataka, povećanjem broja i raznolikosti podataka za učenje, povećavanjem trajanja učenja i odabirom složenijeg modela.

Preučenje mreže (*overfitting*) pojava je koja rezultira vrlo visokim učinkom mreže na skupu podataka za učenje, a značajno lošijim na skupu podataka za ispitivanje koje mreža prije nije "vidjela". Postoji više uzroka preučenja, a neki od njih su šum u slici, nedovoljna raznolikost podataka za učenje, predugo trajanje učenja i presložen model koji se ne može poopćiti na primjenu na skupu podataka za ispitivanje [31]. Problem preučenja rješava se uklanjanjem šuma iz podataka, jer će u protivnom model učiti i šum, odabirom dovoljno raznolikih podataka za učenje, smanjenjem trajanja učenja i poticanjem mreže da bira jednostavniji model. Kako bi se spriječilo preučenje, za vrijeme učenja može se, uz skup podataka za učenje, koristiti i skup podataka za validaciju. Skup podataka za validaciju ne sudjeluje u samom procesu učenja mreže, već služi za iterativno nadziranje učinka trenutnog modela.

Za poboljšanje samog učenja, a time i učinkovitosti modela pri detekciji objekata, poželjno je koristiti augmentaciju ulaznog skupa podataka i unaprijed naučene modele u procesu prenesenog učenja.

Augmentacija ulaznog skupa podataka proces je stvaranja novih podataka od već postojećih. Često se koristi kada je nedovoljan broj dostupnih podataka za učenje i kako bi se povećala raznolikost. U pogledu slikovnih podataka, augmentacija se odnosi na postupke kao što su izrezivanje, zrcaljenje, translacija, skaliranje i rotacija ulaznih slika [24]. Mnogi današnji algoritmi za detekciju objekata augmentaciju vrše sami.

Većina aktualnih algoritama za detekciju objekata dolazi i s mogućnošću korištenja već naučenih utega pa se govori o prenesenom učenju. Preneseno učenje (*transfer learning*) moćna je tehnika u području dubokog učenja u kojoj prethodno naučeni modeli za izvršavanje jednog zadatka mogu biti prenamijenjeni za izvršavanje nekog drugog zadatka [22, 27]. Jednostavno se težinski parametri i pomaci prethodno naučene mreže implementiraju kao početni parametri u učenju mreže izvršavanju nekog drugog zadatka.

Primjerice, naučeni parametri modela za detekciju mačaka, mogu se koristiti kao inicijalni parametri u učenju modela za detekciju pasa. Na taj je način proces učenja novog modela višestruko kraći, potrebno je manje podataka za kvalitetno učenje te se za učenje i primjenu mreže može koristiti i centralna procesorska jedinica (CPU, *Central Processing Unit*), a ne samo GPU [27].

Većina današnjih algoritama za detekciju objekata učena je na ImageNet i Ms COCO skupovima podataka. Kada se ti algoritmi koriste pri učenju detekcije novih objekata, koji se ne nalaze u tim skupovima podataka, u inicijalnoj se fazi učenja koriste parametri mreže naučeni na spomenutim skupovima podataka.

### 4.4. Parametri vrednovanja detekcije objekata temeljene na korištenju konvolucijskih neuronskih mreža

U ovom potpoglavlju bit će opisani neki od korištenih pojmova i parametara vrednovanja izvršene detekcije objekata. Redom će biti opisani pojmovi i parametri presjek nad unijom, *TP*, *FP*, *TN*, *FN*, točnost, preciznost, odziv, *F1* mjera, *PR* krivulja, prosječna preciznost i srednja prosječna preciznost.

Presjek nad unijom ($IoU$, *Intersection over Union*) parametar je koji daje omjer površine presjeka
predviđenog okvira (*bounding box*) i stvarnog okvira (*ground-truth bounding box*) nekog detektiranog objekta i površine koja predstavlja uniju tih dviju površina [15]. Mjera je odstupanja predviđanja od stvarne vrijednosti i opisuje se izrazom 4.28

$$IoU = \frac{B \cap B_g}{B \cup B_g}, \quad (4.28)$$

gdje je $B$ površina predviđenog okvira, a $B_g$ površina stvarnog okvira.

Ako se $IoU$ uspoređuje s nekom vrijednošću praga, često 0.5, mogu se odrediti točno pozitivne i pogrešno pozitivne detekcije.

Ukoliko je $IoU$ predviđenog i stvarnog okvira veći ili jednak vrijednosti praga, što za vrijednost 0.5 znači da se preklapaju barem u polovici svojih površina, ta će se detekcija smatrati točno pozitivnom (*TP, True Positive*) [15].

Ukoliko je $IoU$ predviđenog i stvarnog okvira manji od vrijednosti praga, što za vrijednost 0.5 znači da se ne preklapaju niti u polovici svojih površina, ta će se detekcija smatrati pogrešno pozitivnom (*FP, False Positive*) [15].

Ukoliko mreža nije generirala niti jedan okvir i u stvarnosti slika ne sadrži traženi objekt, takva se detekcija smatra točno negativnom (*TN, True Negative*) [15].

Ukoliko pak mreža nije generirala niti jedan okvir iako u svarnosti na slici postoji traženi objekt, ta će se detekcija smatrati pogrešno negativnom (*FN, False Negative*) [15].

Sada je važno uvesti pojmove točnosti, preciznosti, odziva, *F1* mjere, $PR$ krivulje, prosječne točnosti i srednje prosječne točnosti.

Točnost (*Acc, Accuracy*) predstavlja omjer broja točnih predikcija i broja svih napravljenih predikcija. Računa se prema izrazu 4.29

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}. \quad (4.29)$$

Preciznost (*P, Precision*) predstavlja omjer broja točno detektiranih objekata i broja ukupno detektiranih objekata [15]. U suštini govori koliko je točna detekcija. Računa se prema izrazu 4.30

$$P = \frac{TP}{TP + FP}. \quad (4.30)$$

Odziv (*R, Recall, True Positive Rate, Sensitivity*) predstavlja omjer broja točno detektiranih objekata i broja svih prisutnih objekata [15]. Računa se prema izrazu 4.31

$$R = \frac{TP}{TP + FN}. \quad (4.31)$$

$F_1$-mjera ($F_1$-*score*) predstavlja harmonijsku sredinu preciznosti i odziva [18]. Računa se prema izrazu 4.32

$$F_1 = \frac{2PR}{P + R}. \quad (4.32)$$

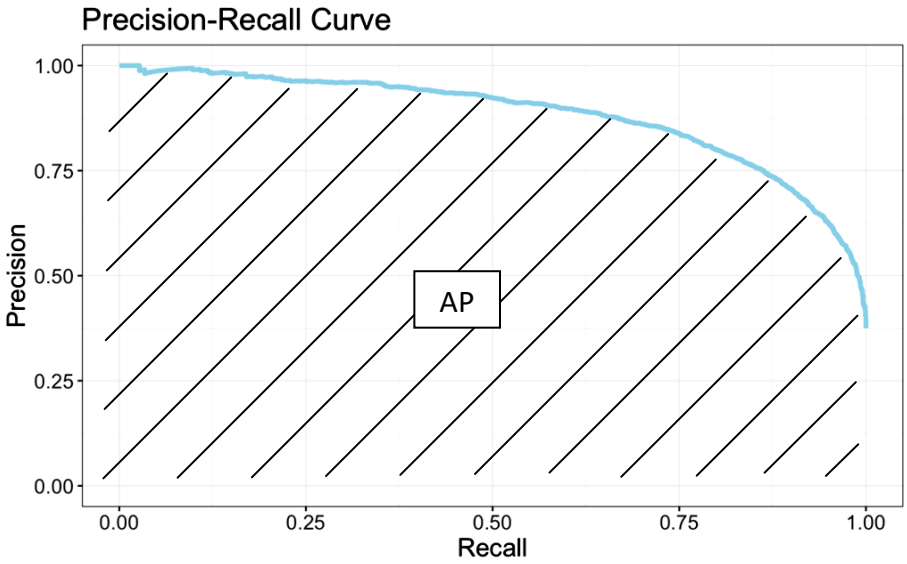
Pomoću mjera preciznosti i odziva crta se $PR$ krivulja (*PR curve*), a površina ispod krivulje daje prosječnu preciznost (*AP, average precision*), koja se često koristi kao mjera uspješnosti modela za detekciju i daje vrijednost iz intervala $[0, 1]$. Detekcija je bolja što je površina ispod krivulje veća.

Primjer $PR$ krivulje dan je slikom 4.14 na kojoj iscrtano područje predstavlja površinu ispod krivulje - $AP$ [18].

U slučaju detekcije više klasa objekata govori se o srednjoj prosječnoj preciznosti ($mAP$, *mean Average Precision*) koja se izračuna izrazom 4.32

$$mAP = \frac{1}{C}\sum_{i=1}^{N} AP_i, \quad (4.32)$$

gdje $C$ broj klasa objekata, a $AP_i$ prosječna preciznost za $i$-tu klasu objekata [15, 18].




*Slika 4.13. PR krivulja i AP*

### 4.5. Skupovi podataka za učenje konvolucijskih neuronskih mreža

Algoritmi za klasifikaciju slika, semantičku segmentaciju, detekciju objekata i slične zadatke računalnog vida trenirani su na velikim skupovima podataka od kojih su najčešći ImageNet, Pascal VOC i COCO.

ImageNet je velika baza podataka koja sadrži više od 14 milijuna slika i njima pripadajućih anotacija te više od 20000 klasa objekata. Upravo je na ImageNetu odrađeno ILVSRC natjecanje koje je odigralo ključnu ulogu u ulasku konvolucijskih neuronskih mreža u algoritme računalnog vida. Natjecanje se prestalo održavati nakon 2017. godine [28].

Pascal VOC (*Visual Object Classes*) još je jedna velika baza podataka koja se razvijala svake godine od 2005. do 2012. u svrhu održavanja Pascal VOC izazova - još jednog natjecanja algoritama računalnog vida. Posljednja baza podataka sadržava preko 11000 slika za 20 klasa objekata [33].

MS COCO (*Microsoft Common Objects in Context*) trenutno je najčešće korištena baza podataka u učenju i međusobnoj usporedbi aktualnih algoritama za izvršavanje zadataka računalnog vida. Koristi se za zadatke klasifikacije, semantičke segmentacije, segmentacije instanci, prepoznavanja konteksta, detekcije objekata, prepoznavanja položaja tijela i drugih. Sadrži 80 kategorija objekata i preko 330000 slika u visokoj rezoluciji [34].

## 4.6. Algoritmi za detekciju objekata temeljeni na konvolucijskim neuronskim mrežama

Algoritmi za detekciju objekata temeljeni na konvolucijskim neuronskim mrežama mogu se podijeliti na algoritme s pristupom u dva koraka (*two-stage approach*) i algoritme s pristupom u jednom koraku (*one-stage approach*). Algoritmi s pristupom u dva koraka utemeljeni su na prijedlozima regija i uglavnom su točniji i precizniji, ali sporiji od algoritama u jednom koraku koji su utemeljeni na regresiji [15, 18, 26].

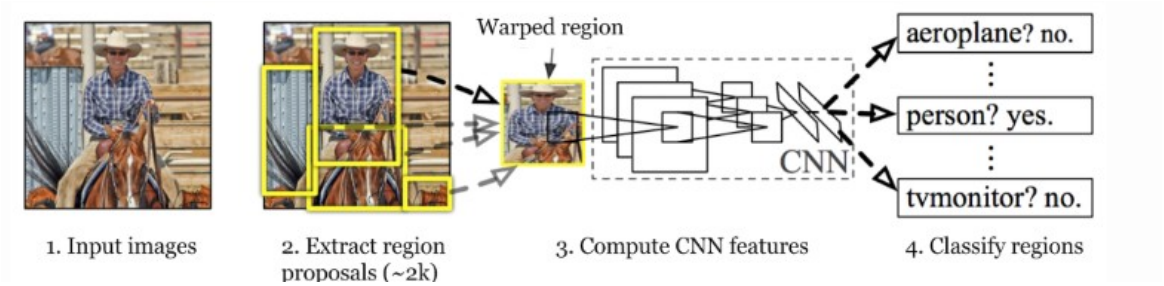
Slijedi pregled dvaju spomenutih tipova algoritama.

### 4.6.1. Algoritmi s pristupom u dva koraka

Algoritmi s pristupom u dva koraka koji će biti opisani su R-CNN, SPP, brza R-CNN, brža R-CNN i maskirana R-CNN.

R-CNN (*Region-based Convolutional Neural Network*) metoda je koju je 2013. godine predstavio Ross Girshick [35]. Koristi konvolucijsku neuronsku mrežu AlexNet arhitekture u kombinaciji s algoritmom za prijedloge regija [15, 25, 26]. Donekle je slična tradicionalnim algoritmima, ali daleko brža i točnija. Kako bi se riješio problem velikog broja redudantnih prozora koristi se algoritam za prijedloge regija koji generira 2000 regija (okvira) na slici na kojima bi se mogao nalaziti objekt [15, 26, 35]. Okviri su generirani korištenjem informacija o slici kao što su boja, tekstura i rubovi, ali neovisno o moguće prisutnim klasama objekata [15, 35]. Budući da se objekt može nalaziti bilo gdje na slici i u bilo kojoj veličini, generirane su regije također različite veličine [35]. Slične regije mogu se rekurzivno kombinirati u veće regije [25]. Nakon toga se koristi konvolucijska neuronska mreža, koja zahtijeva ulaz fiksne veličine, kojoj je onda potrebno prilagoditi regiju [15]. Mana ovog pristupa je što se takvim izobličenjem gube informacije o slici kao što su skala i odnos omjera. Konvolucijska neuronska mreža iz slike izvlači vektor značajki fiksne veličine i dalje se koristi SVM klasifikator kako bi se promatrana regija podijelila na objekt i njegovu pozadinu [15, 25, 26, 35]. Kako bi se poboljšala točnost lokalizacije autor je naučio linearan regresijski model da po potrebi mijenja koordinate prozora za detekciju [15, 25, 26, 35].

Postupak detekcije R-CNN metodom dan je Slikom 4.14.




*Slika 4.14. Detekcija objekata korištenjem R-CNN [35]*

R-CNN metoda predstavljala je revolucionarno otkriće u vremenu kada je nastala. Međutim, sa sobom je dovela i nekoliko mana. Selektivni algoritam pretraživanja regija fiksan je i u njemu se ne događa učenje [25]. Treniranje modela zahtijeva mnogo vremena i računalnog prostora za pohranu [18]. Generiranje 2000 regija čini proces detekcije relativno sporim - potrebno je oko 47 sekundi po slici [25]. Model zahtijeva ulaz fiksne veličine za konvolucijsku neuronsku mrežu. Ovi nedostaci ispravljeni su u sljedećim izdanjima R-CNN metode.

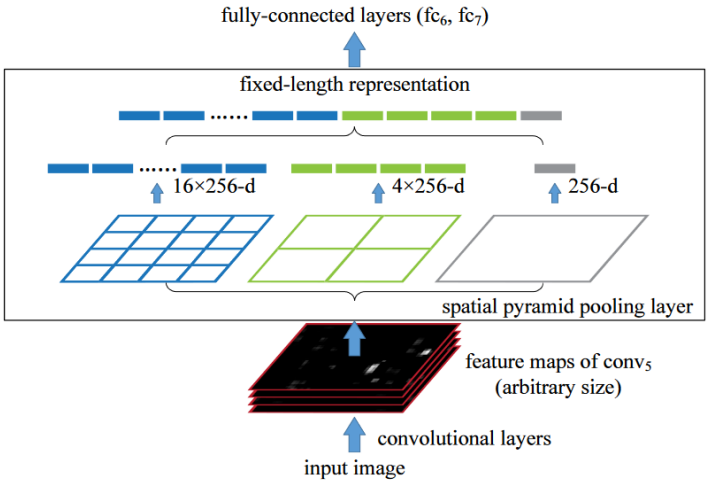
SPP-Net (*Spatial Pyramid Pooling Network*) metoda je kojom je rješen problem potrebe za fiksnom veličinom ulazne slike uvođenjem prostornog piramidalnog sažimanja [26, 36]. Polazi od činjenice da konvolucijski slojevi u mreži zapravo ne zahtijevaju fiksnu veličinu ulaza, već je ona potrebna samo potpuno povezanim slojevima [26]. Stoga u ovoj metodi cijela slika služi kao ulaz u konvolucijske slojeve, čime se dobiva mapa značajki, a zatim se generiraju prijedlozi regija [15]. Između zadnjeg konvolucijskog sloja i prije potpuno povezanog sloja ubačen je sloj prostornog piramidalnog sažimanja koji mape značajki različitih dimenzija za pojedine regije transformira u vektor fiksne duljine [15, 36]. On je zatim ulaz u potpuno povezane slojeve. Uvođenje SPP sloja u mrežu omogućuje sačuvati globalne i lokalne informacije kao što je položaj odabranih regija, odnosno njihovih značajki u odnosu na originalnu sliku [15, 36].

Pojednostavljena struktura SPP-Net-a dana je Slikom 4.15.

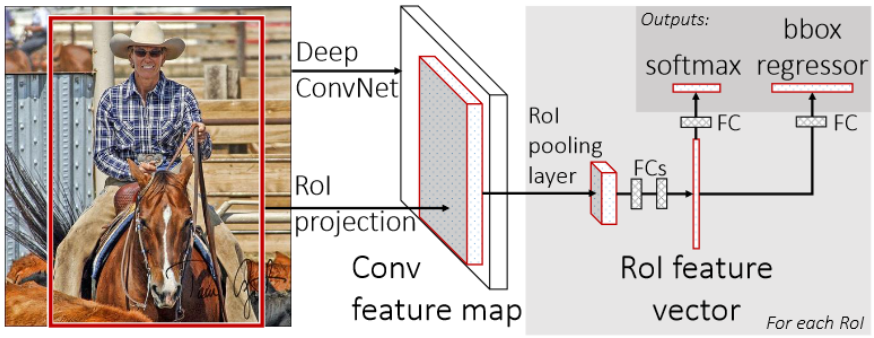
Brza R-CNN (*Fast Region-based Convolutional Neural Network*) metoda brža je i točnija od klasične R-CNN, a uvelike je inspirirana SPP-Net-om [15, 37]. Cijela slika proizvoljnih dimenzija služi kao ulaz u konvolucijsku neuronsku mrežu [26]. Na izlazu iz konvolucijskih slojeva dobiva se mapa značajki iz koje se selektivnim algoritmom pretraživanja generiraju regije interesa (RoI, *Region of Interest*) [15, 18, 26, 37]. Zatim se značajke pojedinih područja interesa sažimaju koristeći sloj sažimanja regija interesa (*RoI pooling*) i dobiva se vektor fiksne duljine [37]. Potpuno povezani slojevi primaju izlaz iz RoI sloja sažimanja, obrađuju ga i šalju u

dvije izlazne sestrinske grane mreže - jednu za klasifikaciju, a drugu za regresiju okvira [15, 26, 37].

Pojednostavljena struktura brze R-CNN dana je slikom 4.16.




*Slika 4.15. Mrežna struktura SPP-Net-a sa slojem prostornog piramidalnog sažimanja [36]*

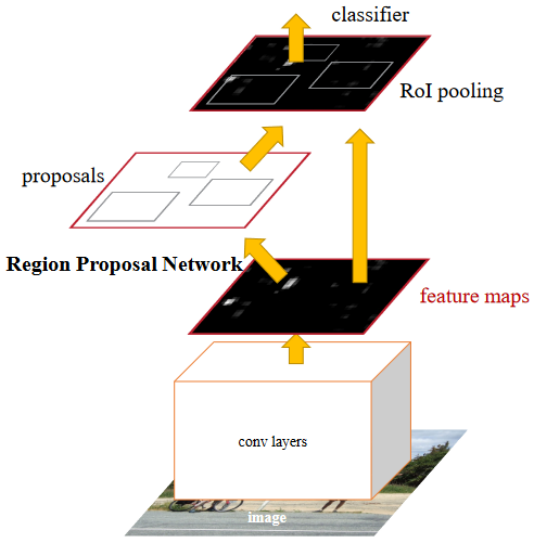



*Slika 4.16. Pojednostavljena struktura brze R-CNN [37]*

Brža R-CNN (*Faster Region-based Convolutional Neural Network*) za razliku od R-CNN i brze R-CNN ne koristi selektivni algoritam za pretraživanje, već se regije predlažu ugradnjom mreže za predlaganje regija (RPN, *Region Proposal Network*) [15, 18, 22, 25, 26, 38]. Cijela se slika najprije koristi kao ulaz u konvolucijsku neuronsku mrežu pri čemu se dobiva mapa značajki, a zatim RPN daje skup prijedloga baznih okvira (*anchor boxes*) različite veličine [26]. Svaki prijedlog ima sebi pridijeljenu vjerojatnost (*objectness score*) da se u toj regiji nalazi objekt [15, 26, 38]. Ta vrijednost se uspoređuje s nekom odabranom vrijednosti praga te se na taj način biraju samo prijedlozi koji su zadovoljili vrijednost praga [26, 38]. Predložene regije i mape značajki zatim se sažimaju u RoI sloju sažimanja (*RoI pooling layer*) koji ih pretvara u vektor fiksne duljine, a on onda služi kao ulaz u potpuno povezane slojeve [18, 26, 38]. Oni daju dva

izlaza - jedan za klasifikaciju, a drugi za regresiju baznih okvira [26, 38]. Brža R-CNN ima veću točnost od prethodno opisanih R-CNN i obrađuje i po 20 slika u sekundi, ali to i dalje nije dovoljno da bi se zadovoljio zahtijev za detekciju u realnom vremenu [15, 22].

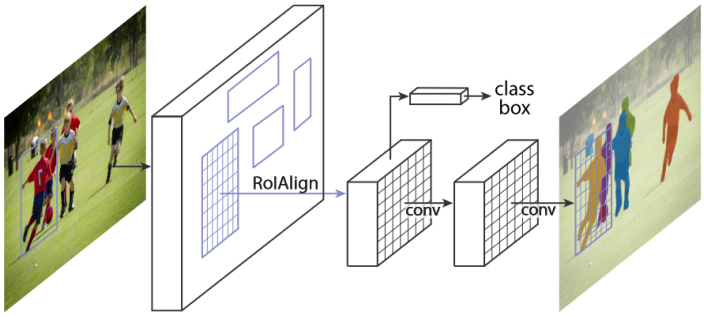
Pojednostavljena struktura brže R-CNN dana je slikom 4.17.




*Slika 4.17. Pojednostavljena struktura brže R-CNN [38]*

Maskirana R-CNN (*Mask Region-based Convolutional Neural Network*) svojevrsna je nadogradnja brže R-CNN. Metoda ima istu RPN koja se koristi i u bržoj R-CNN, no umjesto RoI sloja za sažimanje, ima RoI sloj za poravnanje (*RoI align layer*) koji služi za poravnanje izlučenih značajki s njihovim lokacijama na ulaznoj slici [26, 39]. Dok brža R-CNN ima dva izlaza, koji klasificiraju objekte i predviđaju im okvire, maskirana R-CNN ima i treći izlaz koji daje binarnu masku objekta [26, 39]. Maska objekta dobiva se na izlazu iz male potpuno povezane konvolucijske neuronske mreže segmentacijom instanci kojom se svakom pikselu u slici dodjeljuje kategorija [39]. Ukoliko postoji više objekata iste kategorije, segmentacijom instanci oni će biti razlikovani međusobno [39].

Pojednostavljena struktura maskirane R-CNN dana je slikom 4.18.




*Slika 4.18. Pojednostavljena struktura maskirane R-CNN [39]*

### 4.6.2. Algoritmi s pristupom u jednom koraku

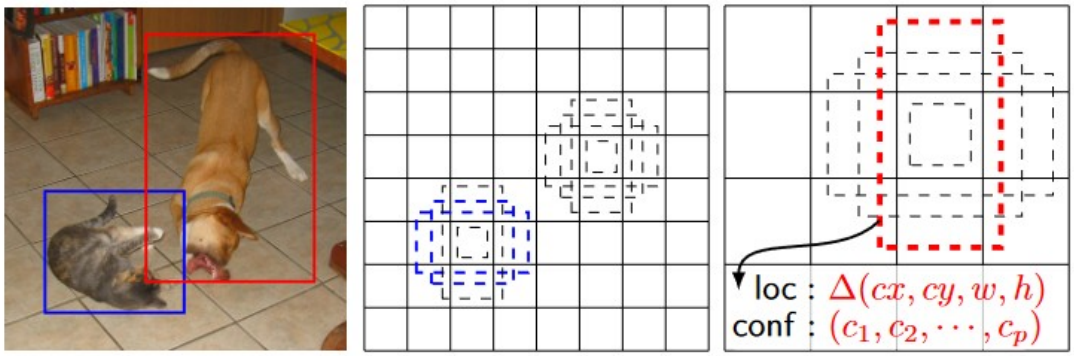
Algoritmi s pristupom u jednom koraku koji će biti opisani su YOLO, SSD i RetinaNet.

YOLO (*You Only Look Once*) algoritam predstavio je 2015. Josh Redmon uvodeći ideju regresije u detekciju objekata [40]. Ulazna se slika podijeli na mrežu dimenzija $S \times S$ od čega svaka ćelija mreže generira $N$ okvira [22, 40]. Ukoliko se središte nekog objekta nalazi u pojedinoj ćeliji, ta je ćelija odgovorna za njegovu detekciju [40]. Na ovaj se način izbjegava uporaba prijedloga regija čime se detekcija značajno ubrzava [18]. Cijela se detekcija odvija u samo jednom koraku [40]. Konvolucijska mreža predviđa klasu i poziciju objekta te pridružuje numeričku vrijednost vjerojatnosti točne predikcije [18, 40]. YOLO algoritam jedan je od najbržih, najpreciznijih i najtočnijih algoritama za detekciju i zadovoljava kriterije detekcije u realnom vremenu obrađujući 45 slika u sekundi [15]. Od 2015. do 2022. izašlo je nekoliko izdanja YOLO algoritama i poneke nadogradnje za svaki. Detaljniji opis slijedi u 5. poglavlju.

SSD (*Single-Shot MultiBox Detector*) metoda je izgrađena na VGG-16 arhitekturi u koju se umjesto završnih potpuno povezanih slojeva dodaje još konvolucijskih slojeva [18, 25, 41]. Konvolucijski slojevi različitih su dimenzija i služe za izlučivanje mapa značajki također različitih dimenzija i broja parametara, koji se progresivno smanjuju, i slažu ih jedne na druge [15, 18, 22, 41]. Početni dio izlučenih mapa značajki sadrži više parametara i koristi se za detekciju manjih objekata, dok se krajnji dio, koji sadrži manje parametara, koristi za detekciju većih objekata [15, 22, 26, 41]. Izlazi iz mapa značajki dalje se koriste u dodatnim konvolucijskim slojevima koji predviđaju bazne okvire različitih skala i omjera stranica i pridružuju im klasu i vjerojatnost točne detekcije [26, 41]. Iako je SSD algoritam jednostavan za učenje i integriranje u sustave, ima visoku točnost detekcije i omogućava detekciju u realnom

vremenu, nedostaci su mu što mu za proces učenja treba mnogo podataka i detekcija nije toliko precizna kada je riječ o manjim objektima [25, 41].

Princip rada SPP dan je slikom 4.19.

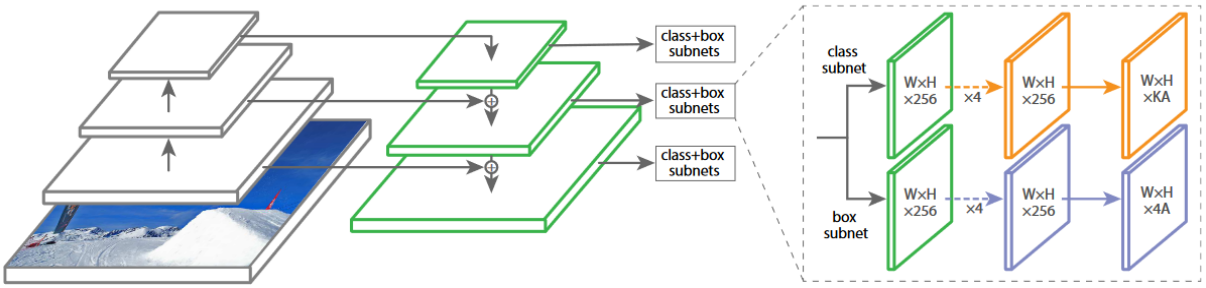



*a)* *b)* *c)*

*Slika 4.19. Princip rada SSD: a) ulazna slika s pridruženim stvarnim okvirima; b) mapa preciznijih značajki* $8 \times 8$ *za detekciju malih objekata; c) mapa grubljih značajki* $4 \times 4$ *za detekciju velikih objekata [41]*

RetinaNet algoritam građen je na ResNet arhitekturi s ciljem spajanja i unaprijeđenja metoda kao što su YOLO i SSD [22, 42]. Sastoji se od piramidalne mreže značajki FPN (*Feature Pyramid Network*) i dvije podmreže za klasifikaciju objekata i regresiju okvira [18, 22, 26, 42]. FPN iz ulazne slike stvara mape značajki različitih dimenzija i broja parametara, koji se progresivno smanjuju, i slaže ih jedne na drugu poput piramide [22, 42]. Dvije spomenute podmreže spojene su na svaku razinu FPN mreže kako bi se stvorili okviri različitih veličina i omjera stranica i njima se pridružile pripadajuće klase objekata [18, 42]. Ideja RetinaNet metode je da velika neravnoteža u broju objekata u prednjem planu i broju objekata u pozadini slike može utjecati na rezultate predikcije pa je uvedena i nova lokalna funkcija gubitka koja proces učenja usmjerava k težim uzorcima [18, 22, 42].

Pojednostavljena struktura RetinaNet-a dana je Slikom 4.20.




*Slika 4.20.Pojednostavljena struktura RetinaNet-a [42]*

# 5. YOLO ALGORITAM

YOLO (*You Only Look Once*) algoritam pristupa detekciji objekata kao regresijskom problemu. Predviđa klasifikaciju objekata i njihovih okvira pomoću jedne konvolucijske neuronske mreže. Kreira se mreža koja dijeli ulaznu sliku na ćelije. Ako se središte objekta koji se treba detektirati, tj. njegova okvira, nalazi unutar pojedine ćelije, ta će ćelija biti odgovorna za njegovu detekciju [40].

YOLO algoritam prošao je kroz mnoge modifikacije te ima pet službenih izdanja: YOLO, YOLOv2/YOLO9000, YOLOv3, YOLOv4 i YOLOv7. Ta će službena izdanja biti opisana u nastavku poglavlja. Na njima su izgrađeni mnogi drugi modeli, a biti će opisana nadogradnja YOLOv4 algoritma u skalirani YOLOv4 algoritam. Slijedi i opis YOLO-R (*You Only Learn One Representation*) algoritma, koji ne pripada YOLO (*You Only Look Once*) algoritmima, ali predstavlja svojevrsnu nadogradnju na skalirani YOLOv4 i služi kao jedan od baznih modela za YOLOv7, koji će kasnije biti primijenjen za detekciju karcinoma mokraćnog mjehura.

## 5.1. Prvi YOLO algoritam

Prije pojave YOLO algoritma detekcija objekata vršila se prenamjenom klasifikatora i njegovom evaluacijom na različitim lokacijama i u različitim skalama (veličinama) na slici. Nakon klasifikacije bilo je potrebno postprocesuiranje kako bi se popravili okviri, uklonile dvostruke detekcije te ponovno evaluirali novi okviri. Ovakav je postupak bio složen, spor i težak za optimizaciju zato što je zahtijevao zasebno učenje svakog dijela mreže.

Prvo izdanje YOLO algoritma pojavilo se 2015. godine. Predstavljeno je radom "You Only Look Once: Unified, Real-Time Object Detection" autora Josepha Redmona, Santosha Divvale, Rossa Girshicka i Alija Farhadija [40].

U nastavku je prvo YOLO izdanje opisano prema autorima [40].

Oni su predstavili problem detekcije objekata kao regresijski problem primijenjen na prostorno odvojene okvire koji u jednom prolasku slike kroz mrežu predviđa na njoj okvire i njima pripadajuće klase.

YOLO algoritam ujedinio je odvojene dijelove postupka detekcije objekata u samo jednu neuronsku mrežu. Za predviđanje okvira mreža se koristi značajkama cijele slike, a ne samo njenih dijelova, zbog čega YOLO algoritam ima dobro razumijevanje konteksta.

Ulazna se slika podijeli pomoću $S \times S$ rešetke. Ukoliko se središte objekta za detekciju nalazi u nekoj ćeliji rešetke, ta je ćelija odgovorna za njegovu detekciju. Svaka ćelija generira nekoliko okvira od kojih je svaki definiran pomoću pet vrijednosti: $x$, $y$, $w$, $h$ i mjere pouzdanosti (*confidence score*). $(x, y)$ koordinate pokazuju gdje se nalazi središte okvira relativno granicama ćelije koja ga je generirala. Širina $w$ i visina $h$ predviđene su relativno cijeloj slici. Mjera pouzdanosti (*confidence score*) pokazuje koliko je mreža sigurna da se unutar generiranog okvira nalazi objekt i koliko je točan generirani okvir. Može se izračunati kao umnožak vjerojatnosti da se u pojedinom okviru nalazi objekt $\Pr(object)$ i $IoU$ vrijednosti generiranog okvira i stvarnog okvira prema izrazu 5.1

$$confidence\ score = \Pr(Object) \cdot IoU_{pred}^{truth}. \quad (5.1)$$

Svaka ćelija rešetke predviđa i $C$ uvjetnih vjerojatnosti klasa, $\Pr(Class_i|Object)$. Ove su vjerojatnosti uvjetovane na ćeliju koja sadrži objekt. Neovisno o tome koliko okvira ćelija generira, za svaku se ćeliju predviđa samo jedan skup vjerojatnosti klasa.

U slučaju postojanja više klasa objekata, množi se uvjetna vjerojatnost neke klase i mjera pouzdanosti pojedinog okvira kako bi se dobila mjera pouzdanosti za pojedinu klasu. Računa se prema izrazu 5.2

$$confidence\ score(Class_i) = \Pr(Class_i|Object) \cdot \Pr(Object) \cdot IoU_{pred}^{truth}. \quad (5.2)$$

Predikcije na izlasku iz YOLO algoritma na kraju su dane u obliku $S \times S \times (B \cdot 5 + C)$ tenzora u kojem $S \times S$ predstavljaju dimenzije rešetke, $B$ broj okvira koje generira pojedina ćelija, $C$ broj klasa objekata koje je model učen detektirati, a broj 5 odnosi se na pet spomenutih vrijednosti kojima je opisan svaki okvir.

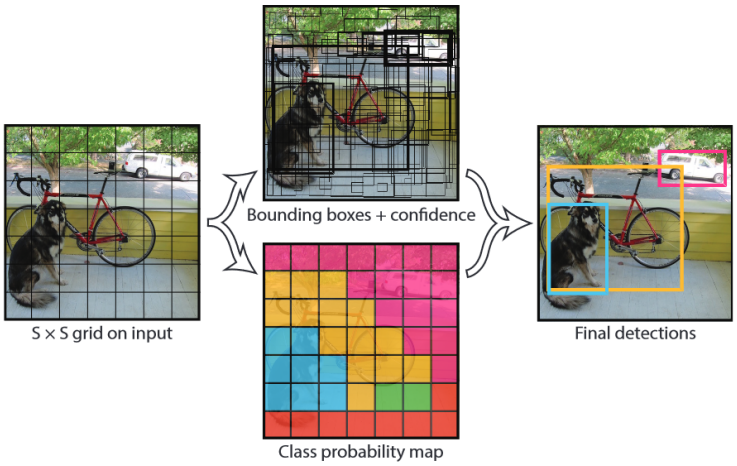
Pojednostavljeni postupak detekcije objekata YOLO algoritmom dan je Slikom 5.1.

YOLO algoritam građen je na Darknet arhitekturi. Ima 24 konvolucijska sloja, koji izlučuju značajke iz slike, i 2 potpuno povezana sloja, koji predviđaju koordinate okvira i pripadajuće

klasne vjerojatnosti. Koristi $1 \times 1$ slojeve koji smanjuju dimenzije mapa značajki iz sebi prethodnih slojeva popraćene s $3 \times 3$ konvolucijskim slojevima. Izlazni sloj koristi linearnu aktivacijsku funkciju, a svi ostali slojevi koriste *leaky* ReLU aktivacijsku funkciju danu izrazom 5.3

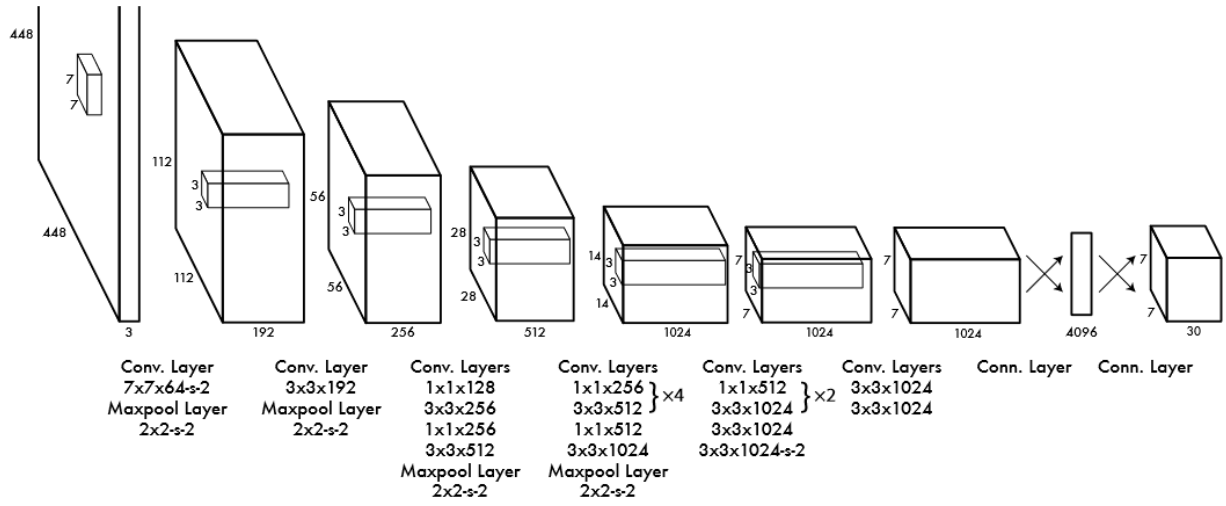
$$f(x) = \begin{cases} x, & x \geq 0 \\ 0.1x, & x < 0 \end{cases}, \qquad (5.3)$$

gdje je $x$ ulaz, a $f(x)$ aktivacijska funkcija.




*Slika 5.1. Pojednostavljeni postupak detekcije objekata YOLO algoritmom[40]*

Cijela je mrežna arhitektura dana je Slikom 5.2.




*Slika 5.2. Arhitektura YOLO algoritma [40]*

Pri učenju modela koristi se sloj ispadanja i augmentacija podataka kako bi se izbjeglo preučenje mreže. Sloj ispadanja preventira koadaptaciju između slojeva na način da pojedinim ulazima u

sloj prisilno postavi vrijednost na nulu. Za augmentaciju podataka korištene su metode nasumičnog skaliranja slika, translacije, promjena ekspozicije i zasićenosti.

Funkcija gubitka računa se samo za one okvire koji imaju najbolji $IoU$ i odgovorni su za detekciju objekta. Nadahnuta je sumom kvadratnih pogrešaka, no uvedene su pojedine modifikacije. Dana je izrazom 5.4

$$
\begin{aligned}
L(W,b) = \lambda_{\text{coord}} & \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{I}_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2] \\
& + \lambda_{\text{coord}} \sum_{\text{i=0}}^{\text{S}^2}\sum_{\text{j=0}}^{\text{B}} \mathbb{I}_{\text{ij}}^{\text{obj}}\left[\left(\sqrt{w_i}-\sqrt{\hat{\text{w}}_{\text{i}}}\right)^2+\left(\sqrt{\text{h}_{\text{i}}}-\sqrt{\hat{\text{h}}_{\text{i}}}\right)^2\right] \\
& + \sum_{\text{i=0}}^{\text{S}^2}\sum_{\text{j=0}}^{\text{B}} \mathbb{I}_{\text{ij}}^{\text{obj}}\left(C_i-\hat{C}_i\right)^2 + \lambda_{\text{noobj}} \sum_{\text{i=0}}^{\text{S}^2}\sum_{\text{j=0}}^{\text{B}} \mathbb{I}_{\text{ij}}^{\text{noobj}}\left(C_i-\hat{C}_i\right)^2 \\
& + \sum_{\text{i=0}}^{\text{S}^2} \mathbb{I}_{\text{ij}}^{\text{obj}} \sum_{c\in\text{classes}} (p_i(c)-\widehat{p_i}(c))^2, \qquad (5.4)
\end{aligned}
$$

koji se sastoji od pet dijelova.

Prvi i drugi dio izraza odnose se na gubitak zbog pogreške lokalizacije. Prvi dio funkcije gubitka odnosi se na sumu kvadratne pogreške utvrđivanja koordinata središta okvira koji sadrže objekt. Suma je pomnožena faktorom $\lambda_{\text{coord}} = 5$ čija je svrha povećati utjecaj gubitka u onim okvirima koji sadrže objekt. S istim je faktorom pomnožen i drugi dio funkcije gubitka koji zbraja kvadratne pogreške procjene širine i visine pojedinih okvira. Pritom se koristi vrijednosti korijena visine i širine. Povod takvom načinu izračuna je u tome što bi klasična kvadratna pogreška za neko fiksno odstupanje rezultirala jednakim gubitkom za velike i male okvire. Na ovaj se način pogreške u dimenzijama malih okvira jače kažnjavaju nego brojčanim iznosom jednake pogreške u dimezijama velikih. Treći i četvrti dio odnose se na gubitak zbog pogreške u mjeri pouzdanosti. Treći dio odnosi se na pogrešku u mjeri pouzdanosti okvira kada se u njemu nalazi objekt, a četvrti dio odnosi se na pogrešku mjere pouzdanosti okvira kada u njemu nema objekta. Četvrta suma množi se faktorom $\lambda_{\text{noobj}} = 0.5$ kako bi se smanjio doprinos pogrešaka u kojima je detektirana pozadina slike. Konačno, peta suma u izrazu odnosi se na pogrešku klasifikacije.

Na kraju mreže dodaje se nemaksimalna supresija kako bi se uklonile dvostruke detekcije.

Neupitna prednost YOLO algoritma pred drugim tadašnjim metodama detekcije njegova je brzina. Osnovni YOLO algoriam obrađuje 45 slika po sekundi (fps, *frames per second*) dok brža verzija Fast YOLO radi brzinom 155 fps. Oboje ispunjava zahtjeve detekcije u realnom vremenu pritom pružajući više nego dvostruko veću $mAP$ nego ostali detektori te kategorije.

YOLO donosi zaključke s obzirom na cijelu sliku pri davanju predikcija. Implicitno analizira i kontekstualne informacije o pojedinim kategorijama objekata što omogućuje manje pogrešnih detekcija pozadine.

Osim toga, model uči poopćene reprezentacije objekata što mu omogućuje da bude uspješnije primijenjen i na nove domene i neočekivane ulaze za razliku od njegovih konkurenata.

Ograničenja algoritma uglavnom su prostorne prirode. Svaka ćelija predviđa ograničen broj okvira i može detektirati samo jednu klasu objekata što utječe na broj međusobno bliskih objekata koje model može detektirati. Posebice velik problem javlja se pri detekciji malih objekata koji se pojavljuju u skupinama. Budući da model uči relativno grube značajke slika, može se pojaviti i poblem pri detektiranju objekata u novim ili neuobičajenim konfiguracijama i omjerima, no ipak, najveća je mana ovog algoritma netočna lokalizacija.

### 5.2. YOLOv2 i YOLO9000

Joseph Redmon i Ali Farhadi 2016. godine predstavljaju drugo izdanje YOLO algoritma - YOLOv2 i YOLO9000 predstavljajući ih radom "YOLO9000: Better, Faster, Stronger" [43].

U nastavku slijedi opis drugog YOLO izdanja prema autorima [43].

Jedan od glavnih ciljeva autora YOLOv2 algoritma bio je poboljšati probleme lokalizacije prisutne u prethodnoj verziji. Želja je povećati točnost pojednostavljenjem mreže istovremeno čineći podatke lakšima za učenje.

YOLOv2 poboljšava osnovni YOLO algoritam uvođenjem normalizacije serija podataka, visokorezolucijskog klasifikatora, konvolucije pomoću baznih okvira, dimenzijskog grupiranja, izravnog predviđanja lokacije, detaljnih značajki i višeskalnog učenja.

Normalizacija serija podataka (*batch normalization*) dodana na konvolucijske slojeve vodi k značajnom poboljšanju u konvergenciji algoritma zato što pomaže u regularizaciji podataka i omogućuje uklanjanje sloja ispadanja bez pojave preučenja. Ovime se poboljšava $mAP$ algoritma za 2%.

Visokorezolucijska YOLOv2 mreža prethodno se ugađa za klasifikaciju, a zatim se prenamijeni u mrežu za detekciju. To doprinosi povećanju $mAP$-a za 4%.

Iz prethodnog YOLO algoritma uklonjeni su potpuno povezani slojevi i za predikciju okvira koriste se bazni okviri (*anchor bounding boxes*). Uvođenje baznih okvira počiva na činjenici da se pojedini objekti uobičajeno pojavljuju na slikama s gotovo konstantnim odnosima dimenzija. Ovi se okviri mogu dalje koristiti kao neka vrsta predložaka za generiranje okvira za detekciju. Za korištenje baznih okvira mreža se prilagodi kako bi mape značajki imale neparan broj stupaca i redaka, odnosno kako bi postojala jedna središnja ćelija na slici. Razlog tomu je što se objekti najčešće nalaze na sredini slike pa je bolje ukoliko je jedna centralna lokacija odgovorna za predikciju okvira tog objekta nego četiri susjedne.

Pokazalo se da je umjesto ručnog određivanja baznih okvira bolje koristiti algoritam grupiranja k-sredina (*K-Means Clustering*). To je jednostavan algoritam strojnog učenja, koji se primjenjuje na skupu okvira za učenje, koji uči automatski predvidjeti dobre bazne okvire. Oni se sada nazivaju prethodnicima (*priors*). Time se modelu odmah pri početku učenja pruža bolja reprezentacija, olakšava učenje i omogućava detekcija do pet objekata po ćeliji.

Koristi se izravno predviđanje lokacije pojedinih okvira na način da se najprije utvrdi kojoj ćeliji pripada pojedini bazni okvir i koji mu je pomak (*offset*) u odnosu na nju. Zatim se predviđa okvir čiji su parametri također određeni u odnosu na tu ćeliju.

YOLOv2 dolazi s uvođenjem finih mapa značajki. Obično se na izlazu iz konvolucijskih slojeva YOLO algoritma nalazi $13 \times 13$ mapa grubih značajki. Na raniji se konvolucijski sloj dodaje prolazni sloj (*passthrough layer*) koji omogućava vađenje finije $26 \times 26$ mape značajki. Prolazni sloj pripaja visokorezolucijske značajke niskorezolucijskima stavljajući ih u različite kanale. Na ovako proširenu izlaznu mapu značajki spojen je detektor.

Jedan česti problem tadašnjih detektora bio je zahtjev na fiksnu veličinu ulazne slike. YOLOv2 algoritam to je riješio promjenjivom veličinom ulaza pri učenju. Svakih 10 serija podataka mreža

sama odabire novu dimenziju ulaza koja je višekratnik broja 32 u rasponu 320 do 608. Ovo je omogućilo mreži dobru i brzu detekciju na različitim rezolucijama slika.

Kako bi se povećala brzina detekcije za bazu je odabrana nova mrežna arhitektura Darknet-19 koja se sastoji od 19 konvolucijskih slojeva i 5 slojeva sažimanja maksimumom. Arhitektura Darknet-19 mreže dana je Tablicom 5.1.

*Tablica 5.1. Arhitektura Darknet-19 mreže [43]*

| Type | Filters | Size/Stride | Output |
|---|---|---|---|
| Convolutional | 32 | 3 x 3 | 224 x 224 |
| Maxpool | | 2 x 2 / 2 | 112 x 112 |
| Convolutional | 64 | 3 x 3 | 112 x 112 |
| Maxpool | | 2 x 2 / 2 | 56 x 56 |
| Convolutional | 128 | 3 x 3 | 56 x 56 |
| Convolutional | 64 | 1 x 1 | 56 x 56 |
| Convolutional | 128 | 3 x 3 | 56 x 56 |
| Maxpool | | 2 x 2 / 2 | 28 x 28 |
| Convolutional | 256 | 3 x 3 | 28 x 28 |
| Convolutional | 128 | 1 x 1 | 28 x 28 |
| Convolutional | 256 | 3 x 3 | 28 x 28 |
| Maxpool | | 2 x 2 / 2 | 14 x 14 |
| Convolutional | 512 | 3 x 3 | 14 x 14 |
| Convolutional | 256 | 1 x 1 | 14 x 14 |
| Convolutional | 512 | 3 x 3 | 14 x 14 |
| Convolutional | 256 | 1 x 1 | 14 x 14 |
| Convolutional | 512 | 3 x 3 | 14 x 14 |
| Maxpool | | 2 x 2 / 2 | 7 x 7 |
| Convolutional | 1024 | 3 x 3 | 7 x 7 |
| Convolutional | 512 | 1 x 1 | 7 x 7 |
| Convolutional | 1024 | 3 x 3 | 7 x 7 |
| Convolutional | 512 | 1 x 1 | 7 x 7 |
| Convolutional | 1024 | 3 x 3 | 7 x 7 |
| Convolutional | 1000 | 1 x 1 | 7 x 7 |
| Avgpool | | Global | 1000 |
| Softmax | | | |

Općenito je velik problem algoritama za detekciju objekata mali skup podataka za učenje. Na Internetu je daleko više dostupnih skupova podataka za klasifikaciju, nego za detekciju. Autori su stoga YOLOv2 mrežu najprije učili na skupu podataka za klasifikaciju objekata, a zatim je prenamijenili i učili na skupu podataka za detekciju objekata. Klasifikacijski podaci su brojniji i omogućavaju mreži učenje značajki povećavajući joj robusnost, a detekcijski podaci služe za

poboljšanje sposobnosti lokalizacije objekata. Ovime je mreži omogućeno da s ograničenim uspjehom detektira objekte koji nisu bili prisutni u skupu za detekciju.

Uz YOLOv2, autori su predstavili i YOLO9000 koji ima mogućnost detekcije više od 9000 klasa objekata. YOLO9000 učen je istovremeno na ImageNet klasifikacijskom skupu podataka i COCO detekcijskom skupu podataka. Pritom se javio problem što su razredi u klasifikacijskim slučajevima podataka detaljniji, dok su u detekcijskima obično generalizirani. Primjerice, skup podataka za klasifikaciju sadržava različite pasmine pasa kao klase, dok su u skupu podataka za detekciju sve te pasmine označene klasom "pas". Ako se model želi istovremeno učiti na ovakvim skupovima podataka, potrebno je pronaći način koherentnog spajanja ovih razreda uvođenjem hijerarhijske klasifikcije. Ona je izvršena korištenjem jezične baze podataka WordNet. Na temelju odabranih podataka iz ImageNet-a i njihovih međusobnih odnosa izgrađeno je hijerarhijsko stablo riječi WordTree.

Pri dodjeljivanju neke klase detektiranom objektu algoritam određuje i njenu uvjetnu vjerojatnost koja je umnožak svih uvjetnih vjerojatnosti čvorova na najkraćem putu koji vodi od korijena stabla do pojedine klase.

Uvođenjem hijerarhijske klasifikacije učinjen je velik korak unaprijed u algoritmima detekcije objekata.

### 5.3. YOLOv3

Joseph Redmon i Ali Farhadi 2018. godine predstavljaju treće izdanje YOLO algoritma - YOLOv3 radom "YOLOv3: An Incremental Improvement" prema kojem je u nastavku i opisano [44].

YOLOv3 ostvaruje još veću brzinu od svojih prethodnika te uspijeva riješiti problem detekcije malih objekata pa mu $mAP$ pri njihovoj detekciji postaje konkurentan tadašnjim naprednijim algoritmima.

Predikcija okvira, kao i kod YOLO9000, bazirana je na baznim okvirima dobivenima korištenjem dimenzijskog grupiranja pomoću *K-Means Clustering* algoritma. Nove predikcije okvira temelje se na parametrima prethodnika. Za funkciju gubitka koristi se suma kvadratnih

pogrešaka. Ukoliko neki prethodnik ima bolji *IoU* sa stvarnim okvirom nego predikcija, predikcija će biti zanemarena.

YOLOv3 predviđa okvire na tri različite skale i izlučuje značajke iz njih koristeći sličan koncept kao FPN. Finije značajke konkatenacijom se spajaju s grubljima. Ovo je omogućilo da mreža više nema problema s predikcijama malih objekata, ali i smanjilo broj mogućih objekata po ćeliji na tri.

Za ekstrakciju značajki koristi se nova mreža, daleko moćnija od Darknet-19, koja predstavlja njegovu kombinaciju s rezidualnim mrežama. Nova mreža ima 53 konvolucijska sloja i zove se Darknet-53. Značajno je veća, ali nije sporija. Arhitektura joj je dana Tablicom 5.2.

*Tablica 5.2. Arhitektura Darknet-53 mreže [44]*

| | **Type** | **Filters** | **Size/Stride** | **Output** |
|---|---|---|---|---|
| | Convolutional | 32 | 3 x 3 | 256 x 256 |
| | Convolutional | 64 | 3 x 3 / 2 | 128 x 128 |
| 1 x | Convolutional | 32 | 1 x 1 | |
| | Convolutional | 64 | 3 x 3 | |
| | Residual | | | 128 x 128 |
| | Convolutional | 128 | 3 x 3 / 2 | 64 x 64 |
| 2 x | Convolutional | 64 | 1 x 1 | |
| | Convolutional | 128 | 3 x 3 | |
| | Residual | | | 64 x 64 |
| | Convolutional | 256 | 3 x 3 / 2 | 32 x 32 |
| 8 x | Convolutional | 128 | 1 x 1 | |
| | Convolutional | 256 | 3 x 3 | |
| | Residual | | | 32 x 32 |
| | Convolutional | 512 | 3 x 3 / 2 | 16 x 16 |
| 8 x | Convolutional | 256 | 1 x 1 | |
| | Convolutional | 512 | 3 x 3 | |
| | Residual | | | 16 x 16 |
| | Convolutional | 1024 | 3 x 3 / 2 | 8 x 8 |
| 4 x | Convolutional | 512 | 1 x 1 | |
| | Convolutional | 1024 | 3 x 3 | |
| | Residual | | 0 | 8 x 8 |
| | Avgpool | | Global | |
| | Connected | | 1000 | |
| | Softmax | | | |

Treći YOLO algoritam uvodi i detekciju više objekata po ćeliji. Do sada je uporabom *softmax* klasifikacije podrazumijevano da jedan okvir sadrži jednu klasu objekata, no to često u stvarnosti nije slučaj. U YOLOv3 implementirana je klasifikacija s više oznaka razreda koje se mogu

preklapati. Tijekom učenja koristi se gubitak binarne unakrsne entropije, a za klasifikaciju nezavisni logistički klasifikatori.

Učenje mreže odvija se i dalje na cijelim slikama korištenjem klasičnih metoda augmentacije podataka, normalizacije serija podataka i višeskalnog učenja.

### 5.4. YOLOv4 i skalirani YOLOv4

Četvrto izdanje YOLO algortima YOLOv4 objavljeno je 2020. godine. Kako je Joseph Redmon odstupio od daljnjeg istraživanje u području računalnog vida, autorstvo preuzimaju Alexey Bochkovskiy, Chien-Yao Wang i Hong-Yuan Mark Liao i predstavljaju novo izdanje algoritma radom "YOLOv4: Optimal Speed and Accuracy of Object Detection" [45].

Nastavak donosi opis četvrtog YOLO izdanja prema autorima [45].

YOLOv4 nadogradnja je prijašnjeg YOLOv3 algoritma koja se sastoji od tri glavna dijela: bazne mreže za izlučivanje značajki (kralježnica algoritma, *backbone*), agregatora značajki (vrat algoritma, *neck*) i prediktora (detektor, glava algoritma, *head*). Za baznu je mrežu odabrana mreža CSPDarknet53, za agregaciju značajki SPP i PANet, a za prediktor YOLOv3.

Autori predstavljaju i dva nova pojma: *Bag of Freebies* (BoF) i *Bag of Specials* (BoS).

*Bag of Freebies* (BoF) pojam je koji predstavlja niz strategija za postizanje boljih rezultata primijenjenih u učenju mreže koje povećavaju cijenu učenja, ali ne utječu na trošak izvršavanja algoritma.

*Bag of Specials* (BoS) pojam je koji predstavlja niz modula i metoda za naknadno procesuiranje čijim se implementiranjem neznatno povećava trošak izvršavanja algoritma, a značajno se povećava točnost detekcije.

Bazna mreža koristi sljedeće BoF metode: CutMix i mozaičku augmentaciju podataka, DropBlock regularizaciju i izglađivanje naziva klasa.

Od BoS metoda bazna mreža koristi: Mish aktivaciju, parcijalne veze unakrsnih stadija (*Cross-stage partial connections*, CSP) i višeulazne težinske rezidualne veze (*Multi-input weighted residual connections*, MiWRC).

Detektor koristi sljedeće BoF metode: CIoU (*Complete IoU*) gubitak, CmBN (*Cross mini-Batch Normalization*) normalizaciju, DropBlock regularizaciju, mozaičku augmentaciju, samokontradiktorno učenje (*Self-Adversarial Training*, SAT), eliminaciju osjetljivosti rešetke, više baznih okvira na jedan stvarni okvir, kosinusno kaljenje, optimalne hiperparametre i nasumične oblike učenja.

Od BoS metoda detektor koristi: Mish aktivaciju, SPP (*Spatial-Pyramid Pooling*) blok, SAM (Spatial Attention Module) blok, PAN (*Path Aggregation Network*) blok i DIoU-NMS (*Distance IoU Non-Maximum Suppresion*).

Slijede kratki opisi najvažnijih od spomenutih BoF i BoS metoda redom.

CutMix je strategija za augmentaciju podataka koja kombinira dvije slike na način da regiju jedne slike zamijeni izrazanom regijom druge slike te tome prilagodi oznake razreda.

Mozaička augmentacija kombinira (dijelove) četiri različite ulazne slike u jednu u različitim omjerima čime se omogućava mreži da promatra objekte izvan njihova konteksta.

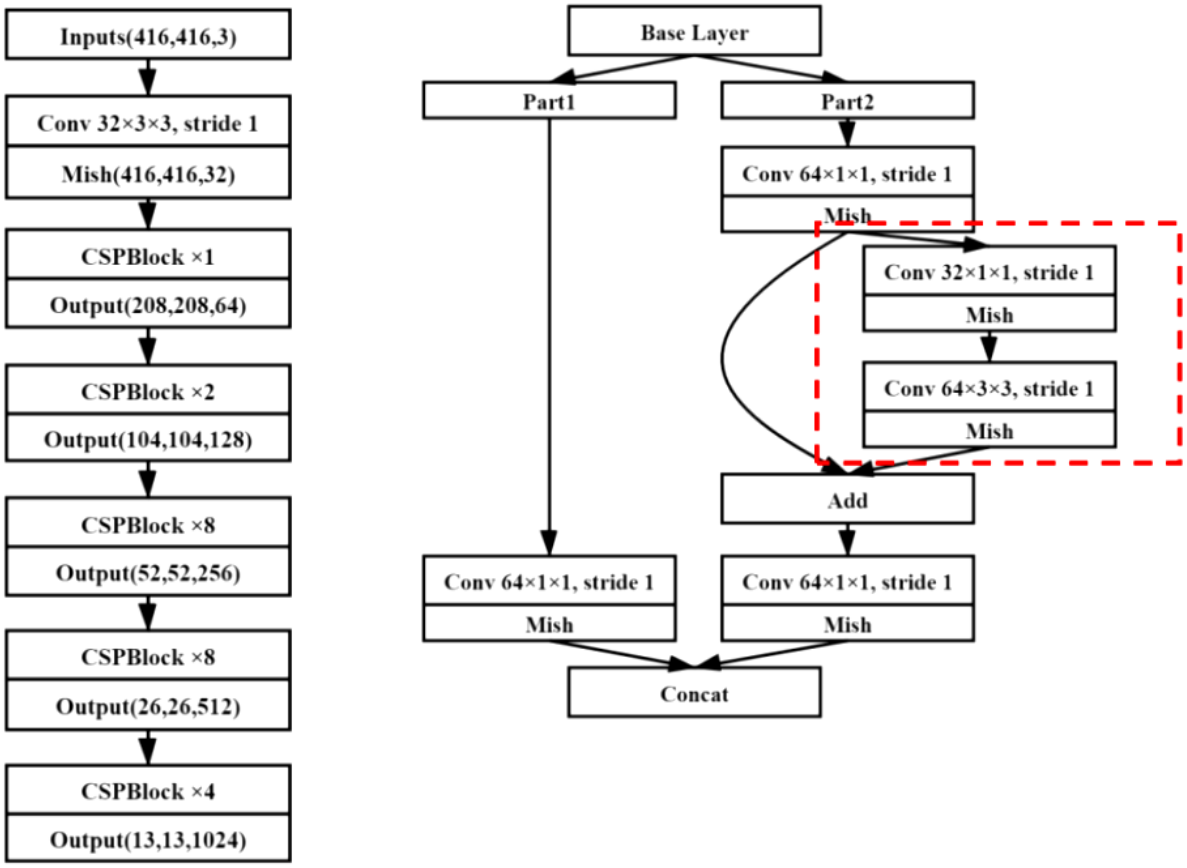
Na slikama se primjenjuje samokontradiktorno učenje (*Self-Adversarial Training*, SAT). Gubitak se propagira unatrag kroz mrežu kako bi se modificirao dio slike koji ima najveći utjecaj na samu mrežu. Na ovaj ju se način nastoji prisiliti da uči druge značajke i izgradi poopćeni model.

DropBlock regularizacija vrši se ispuštanjem blokova piksela iz slike čime se preventira preučenje.

Izglađivanje oznaka razreda (*class label smoothing*) još je jedan način preventiranja preučenja u kojem se mjera pouzdanosti predikcije ograničava na neku vrijednost. Ako mreža da predikciju s mjerom pouzdanosti jednakom 1, velike su mogućnosti da je riječ o pogrešci ili preučenju. Stoga se vrijednost mjere pouzdanosti ograničava, primjerice na vrijednost 0.9.

Mish aktivacijska funkcija nemonotona je samoregularizirajuća derivabilna funkcija čija uporaba u procesu učenja povećava točnost modela.

Kako je već spomenuto, *backbone* YOLOv4 algoritma utemeljen je na CSPDarknet53 mreži. CSPDarknet53 građen je od dva tipa blokova - konvolucijski blok i CSP (*Cross-Stage Partial*) blok, a struktura mu je prikazana Slikom 5.3.




*Slika 5.3. CSPDarknet53 struktura [46]*

Parcijalne veze unakrsnih stadija (*Cross-Stage Partial connections*, CSP) pojavljuju se u CSP bloku. Općenito, porastom broja slojeva u mreži zadnji slojevi imaju sve manje saznanja o naučenim značajkama iz prvih slojeva. U CSP se bloku mapa značajki iz baznog sloja razdvaja i jedan dio prolazi dalje kroz gusto povezani blok, a kopija se šalje izravno u sljedeću fazu obrade. Potom se ponovno spajaju hijerarhijom unakrsnih stadija. CSP omogućuje sačuvati fine značajke za efikasniju propagaciju, smanjenje broja parametara i bolji prolazak gradijenta kroz slojeve.

Višeulazne težinske rezidualne veze (*Multi-input Weighted Residual Connections*, MiWRC) složena su metoda u skaliranju koja služi povećanju točnosti i učinkovitosti. Ugađaju utege skalarno po razinama, a zatim spajaju mape značajki različitih skala.

CIoU (Complete *IoU*) je gubitak koji uzima u obzir preklapanje između predikcije okvira i stvarnog okvira, udaljenost između njihovih središnjih točaka i omjer visine i širine okvira. Utječe na bolju konvergenciju regresije okvira.

CmBN (*Cross mini-Batch Normalization*) metoda je normalizacije u kojoj se svaka mini serija podataka normalizira uzimajući u obzir i srednju vrijednost i devijaciju prethodnih mini serija podataka.

Kosinusno kaljenje vrsta je planirane stope učenja u kojoj se periodično počinje s visokom stopom učenja koja se naglo smanjuje, a zatim opet naglo povećava.

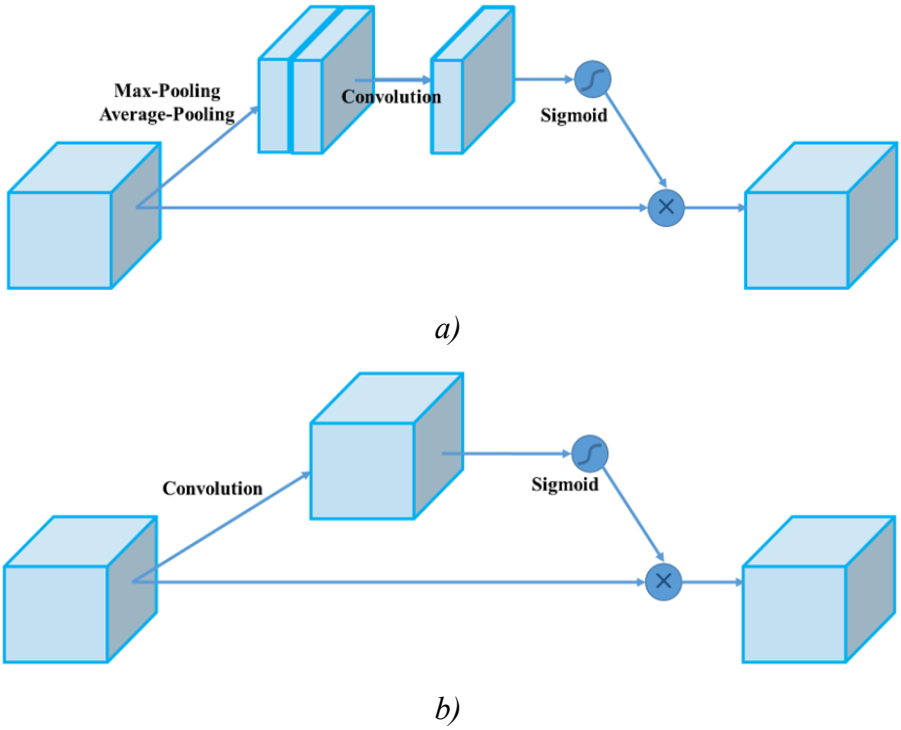
Optimizacija hiperparametara u YOLOv4 vrši se u prvih 10% vremena učenja mreže pomoću genetskih algoritama, metoda rješavanja optimizacijskih problema inspiriranih prirodnom selekcijom.

Već spomenuto prostorno piramidalno sažimanje (SPP, *Spatial Pyramid Pooling*) služi povećanju receptivnog polja i izdvajanju najvažnijih značajki iz mreže. SPP blok dolazi nakon CSPDarknet53 mreže, a prije PAN mreže. U YOLOv4 koristi se modificirani SPP. Mapa značajki najprije se podijeli s obzirom na dubinu. Zatim se na svaki dio primijeni SPP tako da se najprije primijeni sažimanje maksimumom različitim veličinama filtara dobivajući mape značajki različitih dimenzija, a potom se iste kombiniraju ponovno u izlaznu mapu značajki.

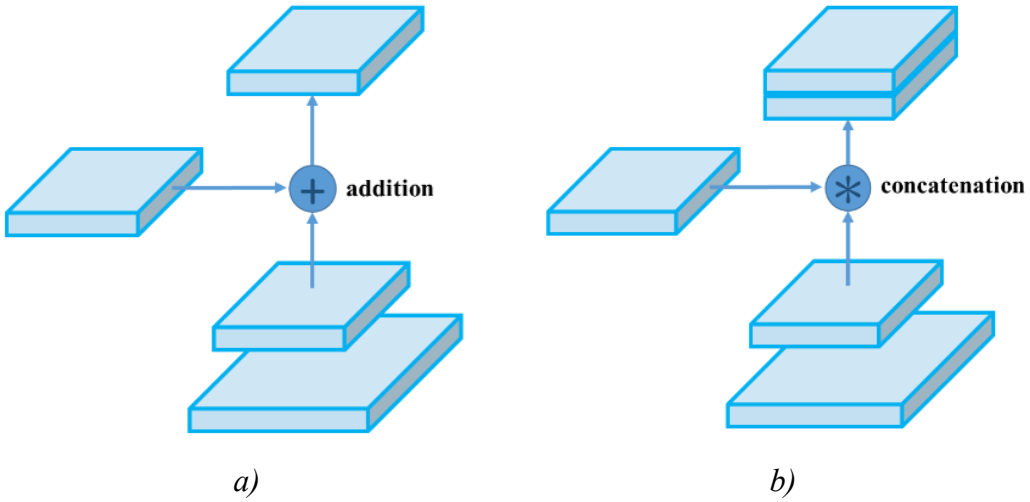
SAM (*Spatial Attention Module*) modul je kojim se ističu lokacije najvažnijih značajki. Općenito se na mapu značajki na izlazu iz nekog konvolucijskog sloja primijene sažimanje maksimumom i sažimanje uprosjećivanjem, a rezultirajuće se mape spoje i prolaze kroz još jedan konvolucijski sloj. Na izlazu se nalazi sigmoidalna aktivacijska funkcija čiji se izlaz unakrsno množi s originalnom mapom što daje istaknute lokacije najvažnijih značajki. U YOLOv4 SAM je modificiran na način da je uklonjeno sažimanje maksimumom i sažimanje uprosječivanjem. Mapa značajki prolazi kroz konvolucijski sloj, a zatim se množi sa svojim originalom dajući na izlazu istaknute lokacije najvažnijih značajki. Usporedba SAM i modificiranog SAM dana je Slikom 5.4.

PAN (PANet, *Path Aggregation Network*) mreža nastavlja se na SPP blok. Služi za prikupljanje značajki i omogućuje bolju propagaciju informacija kroz slojeve. Originalno PAN zbraja značajke iz susjednih slojeva. U YOLOv4 koristi se modificirani PAN koji značajke susjednih

slojeva spaja jedne na druge. Usporedba PAN mreže i modificirane PAN mreže dana je Slikom 5.5.




*Slika 5.4. Usporedba: a) SAM; b) modificirani SAM [45]*




*Slika 5.5. Usporedba: a) PAN; b) modificirani PAN [45]*

DIoU-NMS (*Distance IoU Non-Maximum Suppresion*) vrsta je nemaksimalne supresije kojom se uklanja višak predviđenih okvira. Potrebno je odrediti $IoU$ predviđenog i stvarnog okvira i

udaljenost između središnje točke predviđenog i stvarnog okvira. Ukoliko je razlika manja od neke vrijednosti praga, predikcija se zadržava, a ukoliko je vrijednost veća, taj se okvir briše.

U usporedbi s prethodnom, trećom, verzijom algoritma, YOLOv4 ima značajno povećanje $mAP$ za 10% i fps za 12%. Učen je na COCO skupu podataka da detektira 80 različitih klasa objekata.

Chien-Yao Wang, Alexey Bochkovskiy i Hong-Yuan Mark Liao 2021. godine radom "Scaled-YOLOv4: Scaling Cross Stage Partial Network" predstavili su i skalirani YOLOv4. Skalirani YOLOv4 predstavlja nadogradnju mreže YOLOv4 algoritma na četiri nove skalirane mreže YOLOv4-CSP, YOLOv4-P5, YOLOv4-P6 i YOLOv4-P7. Optimizirana je bazna mreža, a PAN koristi CSP veze i mish aktivaciju. Za treniranje utega koristi se eksponencijalni pomični prosjek (*Exponential Moving Average*, EMA). Za svaku se rezoluciju mreže trenira zasebna neuronska mreža, dok se u YOLOv4 trenirala jedna mreža za sve rezolucije. Promijenjene su aktivacije za visinu i širinu, što je omogućilo brže učenje i uveden je parametar za zadržavanje omjera visine i širine ulaza [47].

Skalirani YOLOv4 u doba svoje pojave predstavljao je najbolji detektor objekata s aspekta brzine i točnosti.

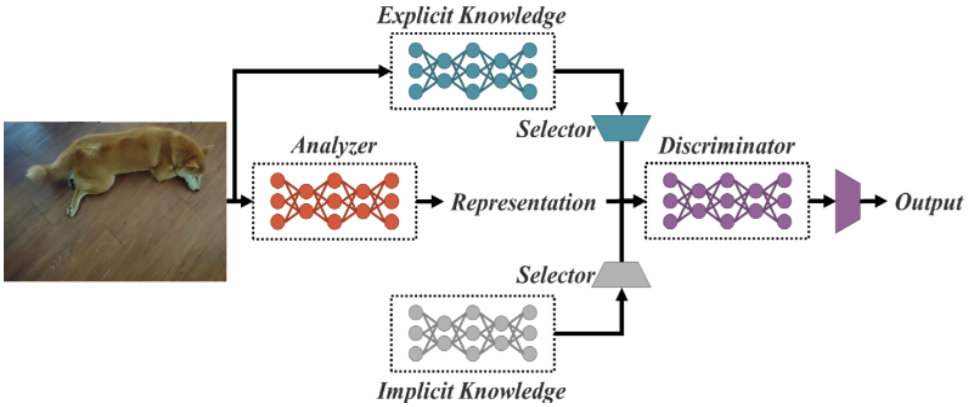
### 5.5. YOLO-R

Chien-Yao Wang, I-Hau Yeh i Hong-Yuan Mark Liao 2021. godine radom "You Only Learn One Representation: Unified Network for Multiple Tasks" predstavili su YOLO-R (*You Only Learn One Representation*) algoritam. U nastavku slijedi njegov opis prema autorima [48].

Motivaciju su pronašli u načinu na koji ljudski mozak stječe znanje. Razlikuje se izričito (eksplicitno) znanje od individualnoga (implicitnoga) znanja. "Eksplicitno je znanje formalno, sistematično, lako razumljivo, nalazi se u knjigama, video-zapisima, bazama podataka, izvješćima i sl. Implicitno je znanje individualizirano, teško je za komunikaciju, sadrži uvjerenja, spoznaje, mentalne modele, perspektive; shvaća se i primjenjuje na podsvjesnoj razini, a proizašlo je iz individualnih opažanja i iskustava" [49]. Zahvaljujući iskustvu, koje je poput velike baze podataka, ljudi mogu kombinirati eksplicitno i implicitno znanje u interpretaciji novih, prije neviđenih informacija.

Konvolucijske neuronske mreže najčešće su učene za izvršavanje jednog zadatka i u tome su uspješne. Međutim, teško su prilagodljive izvršavanju drugih zadataka.

Želja je autora stvoriti jedinstvenu konvolucijsku mrežu koja ujedinjuje i eksplicitno i implicitno znanje kako bi bila u mogućnosti izvršavati više zadataka istovremeno. Vizualno je to prikazano Slikom 5.6.




*Slika 5.6. Princip rada mreže koja ujedinjuje eksplicitno i implicitno znanje pri izvršavanju zadatka [48]*

Eksplicitno znanje koje mreža stječe odgovara plitkim slojevima mreže i temeljeno je na promatranjima dok implicitno znanje odgovara dubljim slojevima mreže.

Za model eksplicitnog znanja autori koriste skalirani YOLOv4-CSP.

Reprezentaciju implicitnog znanja autori definiraju kao konstantni tenzor $Z = \{\mathbf{z}_1, \mathbf{z}_3, \dots, \mathbf{z}_k\}$. Istražili su njegov zapis u obliku vektora, neuronske mreže i matrice od čega je matrični zapis dao najbolje rezultate u primjeni.

Implicitne se reprezentacije mogu primijeniti na različite zadatke kao što su redukcija dimenzija višestrukog prostora, poravnanje kernela, predikcija pomaka okvira, automatsko pretraživanje hiperparametara baznog okvira, odabir značajki, postavljanje preduvjeta za slijed izračuna.

U konvencionalnim je mrežama izlaz predstavljen predikcijom izlaza i pogreškom. Što je pogreška manja, model je prilagođeniji izvršavanju zadanog zadatka te manje sposoban izvršavati druge zadatke.

Kako bi se dobio model koji izvršava više zadataka potrebno je "popustiti" pogrešku. Izlaz se može zapisati izrazom 5.5

$$y = f_\theta(\mathbf{x}) + \epsilon + g_\phi(\epsilon_{ex}(\mathbf{x}) + \epsilon_{im}(\mathbf{z})), \quad (5.5)$$

gdje je $y$ ciljani izraz, $\epsilon$ odstupanje predikcije od ciljanog izraza, $f_\theta(\mathbf{x})$ funkcija neuronske mreže s parametrima $\theta$ primijenjena na ulaz $\mathbf{x}$, $\epsilon_{ex}(\mathbf{x})$ i $\epsilon_{im}(\mathbf{z})$ su operacije eksplicitne i implicitne pogreške, a $g_\phi$ je operacija koja kombinira eksplicitno i implicitno znanje.

Ako se eksplicitno i implicitno znanje integriraju u funkciju $f_\theta$, izraz 5.5 može se zapisati izrazom 5.6

$$y = f_\theta(\mathbf{x}) \star g_\phi(\mathbf{z}), \quad (5.6)$$

gdje je $\star$ neki operator koji se može koristiti za kombinaciju funkcija $f_\theta(\mathbf{x})$ i $g_\phi(\mathbf{z})$. Autori su istražili operacije zbrajanja, množenja i spajanja (konkatenacije). Ovisno o zadatku koji se izvršava, pojedini operator daje bolje rezultate.

Istraživanje primjene YOLO-R algoritma u području detekcije objekata pokazalo je da YOLO-R daje približno jednaku ili veću preciznost kao konkurentni modeli, ali ih pritom nadmašuje u brzini.

### 5.6. YOLOv7

YOLOv7 najnovije je izdanje YOLO algoritma koje nadmašuje sve dosadašnje algoritme detekcije objekata i po brzini i po točnosti. Službeno je predstavljen u srpnju 2022. godine radom "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors" čiji su autori Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao [50]. Kod je javno dostupan (*open source*) na platformi GitHub na korisničkome profilu WongKinYiu (Chien-Yao Wang) [51].

U nastavku potpoglavlja YOLOv7 je predstavljen kako ga opisuju autori [50, 51].

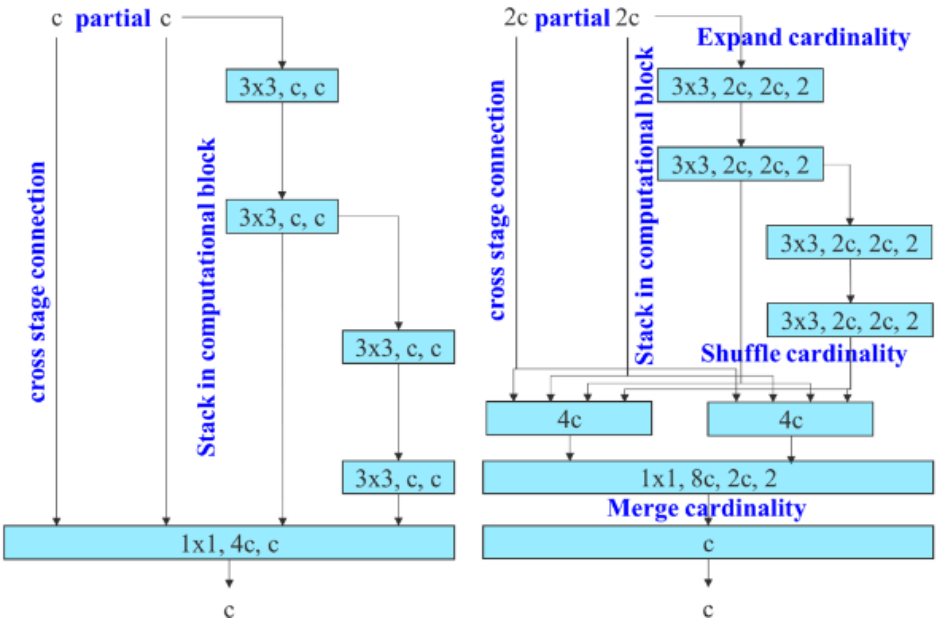
Kao bazni modeli za izgradnju YOLOv7 poslužili su YOLOv4, skalirani YOLOv4 i YOLO-R, modeli na kojima su autori prethodno radili. Za razliku od njih YOLOv7 ima 40% manje

parametara i 50% manje izračuna. Zahtijeva jeftiniji računalni hardver i trenira se brže ne zahtijevajući velike skupove podataka niti predtrenirane utege.

Autori donose optimizaciju arhitekture i procesa učenja. Uvode nekoliko značajnih promjena u arhitekturi te predstavljaju optimizirane module i metode koji mogu povećati cijenu učenja kako bi se poboljšala točnost, ali bez povećanja cijene izvršenja detekcije. Te module i metode nazivaju *trainable bag-of-freebies*, odnosno BoF modulima i metodama koji se mogu učiti. Njihovim uvođenjem nastoje postići robusniju funkciju gubitka, učinkovitije dodjeljivanje oznaka i učinkovitije učenje.

YOLOv7 arhitektura ima CSPDarknet *backbone*. Na njega autori nadodaju E-ELAN (*Extended Efficient Layer Aggregation Network*) i uvode novu metodu skaliranje modela baziranih na konkatenaciji.

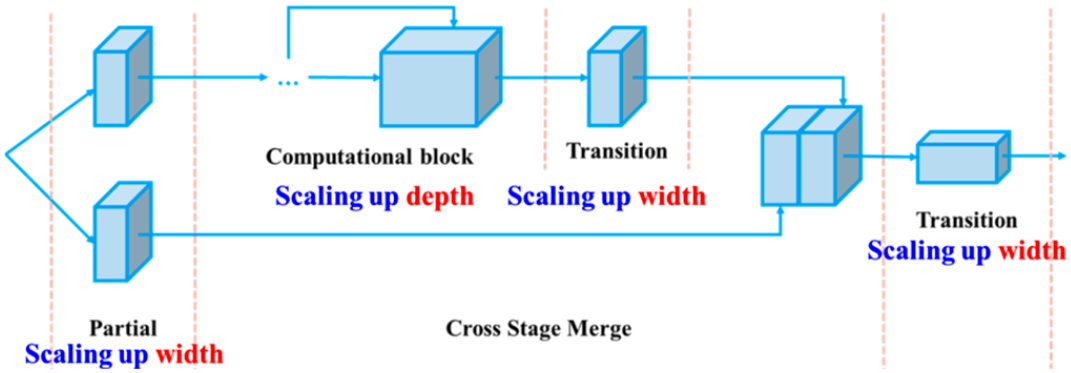
ELAN (*Efficient Layer Aggregation Network*) mreža nastala je s težnjom za dizajniranjem što učinkovitije mreže nadziranjem najduljeg najkraćeg puta gradijenta. Autori YOLOv7 predlažu njenu modificiranu inačicu E-ELAN. E-ELAN (*Extended Efficient Layer Aggregation Network*) je računski blok u YOLOv7 *backbone*-u. Omogućava modelu bolje učenje korištenjem proširenja, miješanja i spajanja kardinalnosti kako bi se postiglo neprekinuto poboljšanje učenja mreže bez da se naruši izvorni put gradijenta. Usporedba ELAN i E-ELAN mreže dana je Slikom 5.7.




*a)* *b)*

*Slika 5.7. Usporedba: a) ELAN; b) E-ELAN [50]*

Skaliranje modela način je kojim se model može prilagoditi različitim brzinama izvršavanja i uređajima u koje je implementiran. Skaliranje modela može se vršiti s obzirom na rezoluciju (veličinu ulazne slike), dubinu (broj slojeva), širinu (broj kanala) i stadije (broj piramida značajki) te utječe na količinu parametara mreže i izračuna, brzinu izvršavanja i točnost. Kod modela čija je arhitektura bazirana na konkatenaciji, nije moguće mijenjati samo jedan parametar skaliranja bez da se naruši cijeli model. Autori predlažu složeno skaliranje modela (*compound model scaling*) koje omogućava modelu zadržati svojstva i optimalnu strukturu. Kada se, primjerice, skalira dubina računskog bloka, potrebno je izračunati i promjenu njegovog izlaznog kanala, a zatim se napravi skaliranje dubine koje će jednako promijeniti i tranzicijske slojeve kako je prikazano Slikom 5.8.




*Slika 5.8. Složeno skaliranje dubine i širine za model baziran na konkatenaciji [50]*

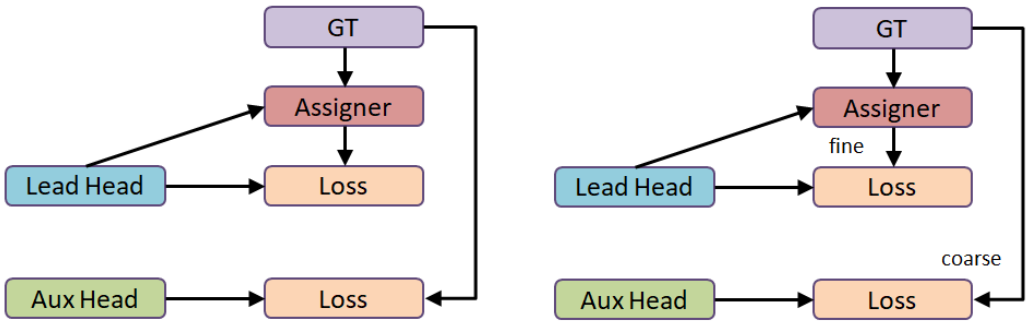
Autori predstavljaju već spomenute nove BoF metode koje se mogu trenirati: plansku reparametriziranu konvolucija i grubo i fino dodjeljivanje oznaka.

Promatranjem strategija za reparametrizaciju modela primjenjivih na slojeve različitih mreža s konceptom puta propagacije gradijenta predlažu plansku reparametrizaciju modela. Tehnike reparametrizacije modela spajaju više računskih modula u jedan u trenutku primjene algoritma. Dijele se na dvije vrste - onu na razini modula i onu na razini modela. Autori YOLOv7 algoritma razvili su novu reparametrizaciju na razini modula. Preko tijeka gradijenta promatrali su kako reparametrizirana konvolucija djeluje u kombinaciji s različitim mrežama. Koristili su pritom RepConv model za strojno učenje. RepConv kombinira $3 \times 3$ konvoluciju, $1 \times 1$ konvoluciju i jediničnu vezu identiteta u jednom konvolucijskom sloju. Ta veza identiteta može narušiti reziduale i konkatenaciju u nekim arhitekturama, koji osiguravaju različitost gradijenata za

različite mape značajki. Iz tog razloga autori predlažu korištenje RepConv bez jedinične veze - RepConvN za dizajniranje arhitekture planske reparametrizirane konvolucije.

YOLO arhitektura u pravilu završava glavom (detektorom). Po uzoru na tehniku dubokog nadzora uvodi se sporedna (pomoćna) glava. Duboki se nadzor kao tehnika često koristi pri učenju dubokih neuronskih mreža, a glavna mu je zamisao dodati pomoćnu glavu u srednje slojeve mreže koja je vođena utezima s pomoćnim gubitkom. Sporedna glava služi kao pomoć pri učenju dok vodeća (glavna) glava daje konačan izlaz - dodjeljuje oznake. Dodatno je uveden i mehanizam dodjeljivanja oznaka koji uzima u obzir predikcije i stvarne vrijednosti i dodjeljuje meke oznake (*soft labels*). Meke oznake koriste izračune i optimizacijske metode koje uzimaju u obzir kvalitetu i raspodjelu predikcija na izlazu u usporedbi sa stvarnim željenim izlazom.

Autori predlažu dvije nove metode dodjeljivanja oznaka u kojima se i pomoćna i vodeća glava vode predikcijom vodeće glave i tako uče: dodjeljivanje oznaka vođeno vodećom glavom (*lead head guided label assigner*) i od grubog do finog dodjeljivanja oznaka vođenog vodećom glavom (*coarse-to-fine lead head guided label assigner*) (Slika 5.9).




*Slika 5.9. Dodjeljivanje oznaka: a) vođeno vodećom glavom; b) od grubog do finog vođeno vodećom glavom [50]*

Dodjeljivanje oznaka vođeno vodećom glavom vrši se s obzirom na predikciju glavne glave i stvarnu vrijednost, na temelju čega se kroz optimizacijski proces dodjeljuju meke oznake. Ove oznake koriste se u učenju i za pomoćnu i za glavnu glavu. Glavna glava ima bolje sposobnosti učenja nego sporedna i njome dobivene oznake daju dobru reprezentaciju raspodjele i korelacije između ulaza i željenog izlaza. Sporedna glava izravno uči informacije, koje je naučila glavna glava, što glavnoj glavi omogućuje da se dalje usmjeri k učenju još nenaučenih rezidualnih informacija.

Metoda od grubog do finog dodjeljivanja oznaka vođenog vodećom glavom dodjeljuje meke oznake na isti način kako je prethodno opisano. Međutim, generiraju se dvije vrste mekih oznaka - fine i grube. Fine oznake jednake su mekim oznakama kakve se dobiju prethodno opisanom metodom, a grube oznake dobiju se tako da se više ćelija slike tretira kao da su pozitivne (sadrže objekt) i one služe za učenje sporedne glave. Ova se metoda pri ispitivanju pokazala boljom.

Od ostalih BoF metoda korištene su normalizacija serije podataka, implicitno znanje po uzoru na YOLO-R i EMA (*Exponential Mean Average*) model.

Kreirano je sedam različitih YOLOv7 modela: YOLOv7, YOLOv7-tiny, YOLOv7-X, YOLOv7-W6, YOLOv7-E6, YOLOv7-D6 i YOLOv7-E6E. Trenirani su na COCO skupu podataka. Usporedba pojedinih modela dana je Tablicom 5.3, a arhitektura osnovnog modela dana je Tablicom 5.4.

*Tablica 5.3. Usporedba različitih YOLOv7 modela [50]*

| **Model** | **Broj parametara [milijuni]** | **Brzina [fps]** | **AP test [%]** | **Aktivacijska funkcija** |
|---|---|---|---|---|
| **YOLOv7-tiny** | 6.2 | 286 | 38.7 | Leaky ReLU |
| **YOLOv7** | 36.9 | 161 | 51.4 | SiLU |
| **YOLOv7-X** | 71.3 | 114 | 53.1 | SiLU |
| **YOLOv7-W6** | 70.04 | 84 | 54.9 | SiLU |
| **YOLOv7-E6** | 97.2 | 56 | 56.0 | SiLU |
| **YOLOv7-D6** | 154.7 | 44 | 56.6 | SiLU |
| **YOLOv7-E6E** | 151.7 | 36 | 56.8 | SiLU |

*Tablica 5.4. Arhitektura YOLOv7 modela [50]*

| | **YOLOv7** |
|---|---|
| **Stage 0** | 3x3/1 Conv, 32 |
| **Stage 1** | 3x3/2 Conv, 64<br>3x3/1 Conv, 64 |
| **Stage 2** | 3x3/2 Conv, 128<br>2-4 ELAN, 256 |
| **Stage 3** | /2 Down, 256<br>2-4 ELAN, 512 |
| **Stage 4** | /2 Down, 512<br>2-4 ELAN, 1024 |
| **Stage 5** | /2 Down, 1024<br>2-4 ELAN, 1024 |
| **Stage 5** | CSPSPP, 512 |
| **Stage 4** | *2 Up, 512<br>1-4 ELAN, 256 |
| **Stage 3** | *2 Up, 256<br>1-4 ELAN, 128 |
| **Stage 4** | /2 Down, 512<br>1-4 ELAN, 256 |
| **Stage 5** | /2 Down, 1024<br>1-4 ELAN, 512 |

# 6. DETEKCIJA KARCINOMA MOKRAĆNOG MJEHURA KORIŠTENJEM YOLOV7 ALGORITMA

Ovo poglavlje donosi primjenu osnovnog YOLOv7 algoritma u učenju modela da prepoznaje tumorske nakupine na CT i MRI snimkama frontalnog, sagitalnog i horizontalnog presjeka trbušne šupljine.

Poglavlje je podijeljeno u nekoliko cjelina koje pobliže opisuju korišteni skup slikovnih podataka, njegovo anotiranje i primjenu YOLOv7 algoritma u procesu učenja, dok posljednja cjelina donosi i raspravlja dobivene rezultate.

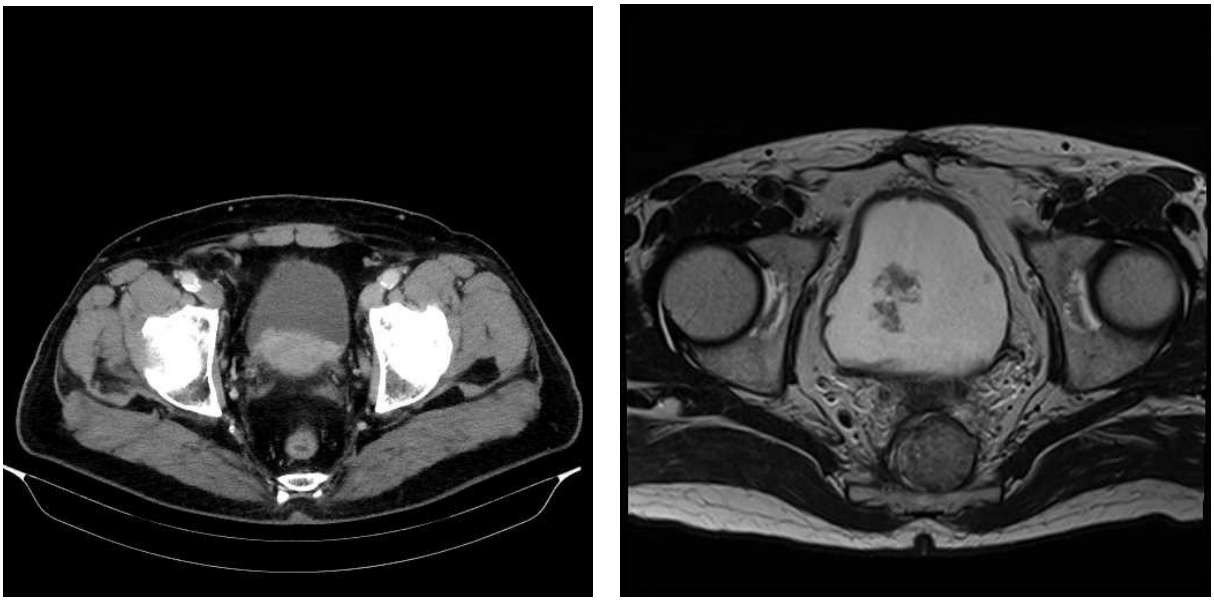
## 6.1. Skup podataka

Podaci korišteni u procesu učenja modela sastoje se od CT i MRI slika frontalnog, horizontalnog i sagitalnog presjeka trbušne šupljine u JPG formatu.

Primjeri CT i MRI zapisa frontalnog, horizontalnog i sagitalnog presjeka trbušne šupljine dani su redom Slikom 6.1, Slikom 6.2 i Slikom 6.3.

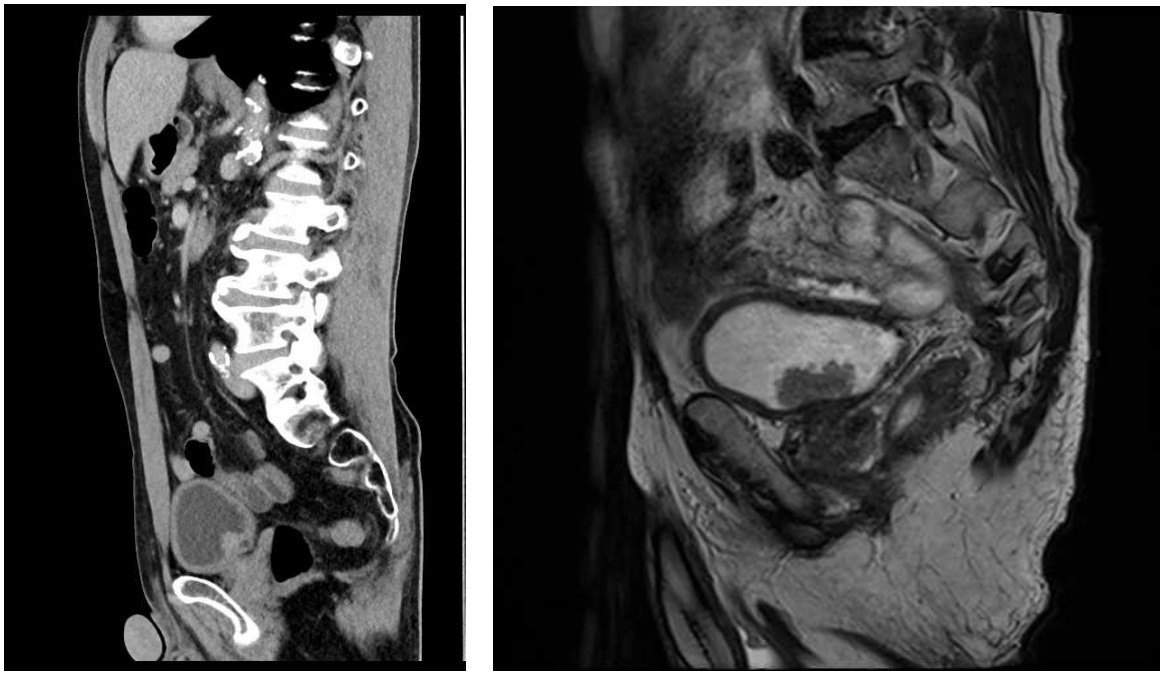
*a)* *b)*

*Slika 6.1. Frontalni presjek trbušne šupljine: a) CT; b) MRI [52]*



*a)* *b)*

*Slika 6.2. Horizontalni presjek trbušne šupljine: a) CT; b) MRI*



*a)* *b)*

*Slika 6.3. Sagitalni presjek trbušne šupljine: a) CT; b) MRI*

Cijeli se skup sastojao od ukupno 392 slike frontalnog presjeka, 276 slika horizontalnog presjeka i 144 slike sagitalnog presjeka trbušne šupljine od kojih je svaka prikazivala tumorske nakupine na mokraćnom mjehuru.

Slike su podijeljene u tri skupa - skup za učenje (*training set*), skup za validaciju (*validation set*) i skup za ispitivanje (*testing set*) u omjeru 8: 1: 1 kako je prikazano Tablicom 6.1.

*Tablica 6.1. Podjela podataka na skup za učenje, validaciju i ispitivanje*

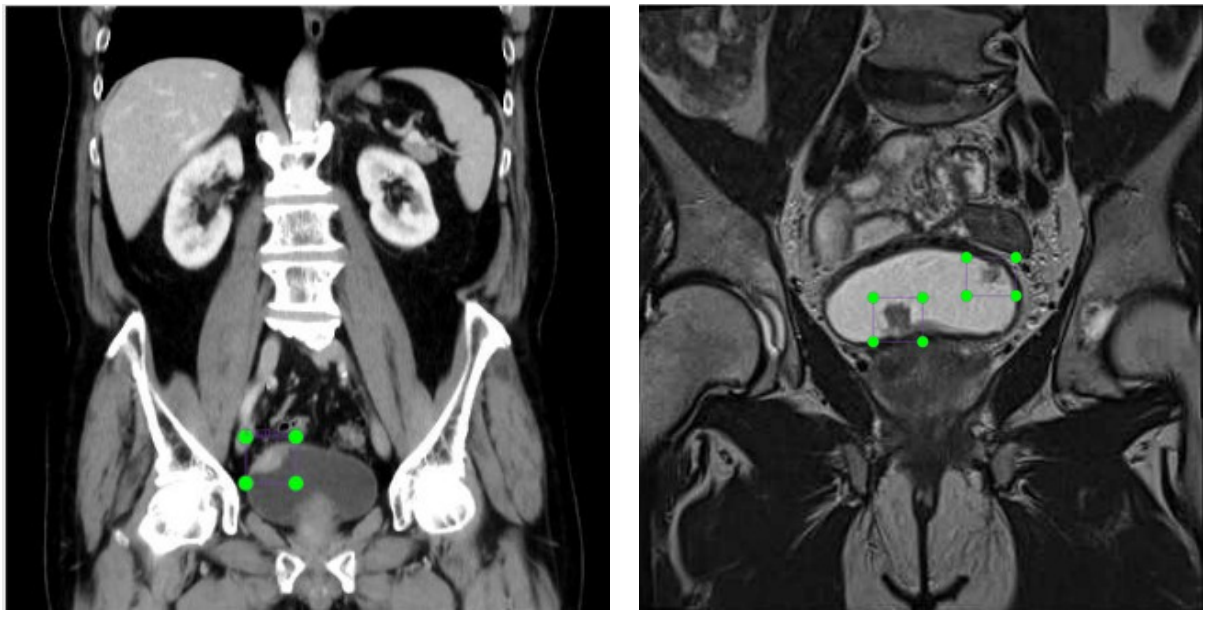
| | ***Training set* (80%)** | ***Validation set* (10%)** | ***Testing set* (10%)** |
|---|---|---|---|
| **Frontalni presjek** | 314 | 39 | 39 |
| **Horizontalni presjek** | 221 | 28 | 27 |
| **Sagitalni presjek** | 115 | 15 | 14 |

Jednom razvrstane slikovne datoteke prenesene su na Google disk osobnog korisničkog računa.

### 6.2. Anotiranje podataka

Kako bi se slikovni zapisi mogli koristiti u učenju modela potrebno je na njima označiti (anotirati) tumore. U tu svrhu korišten je alat za grafičku anotaciju slika LabelImg.

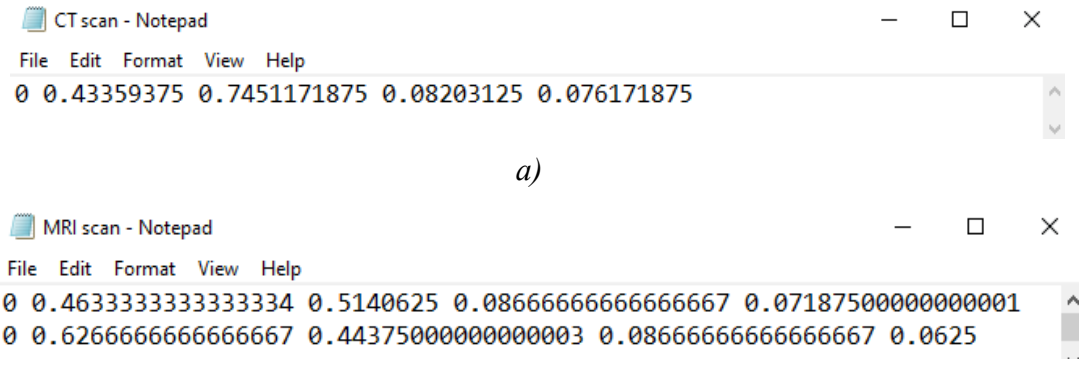
Proces anotiranja provodi se postavljanjem okvira (*bounding box*) oko područja koje sadrži tumor i označavanja pripadnosti pojedinog okvira odgovarajućem razredu kako je prikazano Slikom 6.4. U ovom slučaju postojao je jedan razred pod nazivom "tumor".



*a)* *b)*

*Slika 6.4. Označavanje područja prisutnosti tumora ostavljanjem okvira na frontalnom presjeku trbušne šupljine: a) CT; b) MRI [52]*

LabelImg anotacije pohranjuje u obliku tekstualnih datoteka koje sadrže pet oznaka - razred kojem objekt pripada, koordinate središnje točke okvira te širinu i visinu okvira skaliranih s obzirom na sliku kako je prikazano Slikom 6.5.




*Slika 6.5. Tekstualne datoteke koje odgovaraju okvirima oko područja prisutnosti tumora na Slici 6.4. za: a) CT; b) MRI [52]*

Po završetku procesa anotiranja, tekstualne su datoteke podijeljene u odgovarajuće skupove za učenje, validaciju i ispitivanje modela i prenesene na Google disk.

### 6.3. Učenje

Proces učenja modela računalno je zahtjevan ukoliko se provodi na osobnim računalima. Kako bi se izbjegao takav problem, za učenje modela korišteno je razvojno okruženje Google Colaboratory koje korisniku omogućuje izvršavanje koda zapisanog u Python programskom jeziku na besplatnom GPU-u.

S platforme GitHub od autora je preuzet *open source* kod YOLOv7 algoritma koji je prenesen u Jupyter bilježnicu [51].

Pojedine datoteke potrebne za izvršavanje koda izmijenjene su kako bi se optimiziralo učenje koda za vlastiti slučaj. Učenje modela prilagođeno je učenju prepoznavanja jedne klase objekata na slikama u nijansama sivih tonova. Potonja izmjena omogućava ubrzanje procesa učenja, jer je, umjesto učenja trokanalne reprezentacije slika, model odmah prilagođen na jednokanalnu reprezentaciju.

Pokretanjem koda dodijeljen je Tesla T4 Tensor Core GPU.

Proces učenja podijeljen je u tri dijela izvršena sljedećim redoslijedom: učenje prepoznavanja tumora na frontalnom presjeku trbušne šupljine, učenje prepoznavanja tumora na horizontalnom presjeku trbušne šupljine i učenje prepoznavanja tumora na sagitalnom presjeku trbušne šupljine.

Kod za izvršavanje procesa učenja i ispitivanje dan je u Prilogu A.

Učenje je započeto na frontalnom presjeku trbušne šupljine, jer je za isti bilo najviše dostupnih podataka. Korišten je model unaprijed učen na COCO skupu podataka. Parametri učenja dani su Tablicom 6.2.

*Tablica 6.2. Odabrani parametri učenja modela*

| **Parametar** | **Opis** | **Učenje na frontalnom presjeku** | **Učenje na horizontalnom presjeku** | **Učenje na sagitalnom presjeku** |
|---|---|---|---|---|
| data | Sadrži vezu na skup podataka za učenje i validaciju | custom_data_frontal.yaml | custom_data_horizontal.yaml | custom_data_sagittal.yaml |
| img-size | Veličina slike | 640 | 640 | 640 |
| batch-size | Broj slika po seriji podataka | 16 | 16 | 16 |
| cfg | Korišteni model | yolov7.yaml (modificiran za jednu klasu) | yolov7.yaml (modificiran za jednu klasu) | yolov7.yaml (modificiran za jednu klasu) |
| weights | Inicijalni težinski faktori modela | yolo7.pt (predtrenirani na COCO skupu podataka) | best.pt (optimalni utezi dobiveni učenjem na frontalnom presjeku) | best.pt (optimalni utezi dobiveni učenjem na horizontalnom presjeku) |
| epoch | Broj prolazaka algoritma kroz skup podataka za učenje | 600 | 600 | 600 |

Po izvršenju 600 epoha učenja najbolji dobiveni model za detekciju karcinoma na frontalnom presjeku trbušne šupljine ispitan je na skupu slika za ispitivanje, a zatim dalje korišten u procesu učenja prepoznavanja tumora na horizontalnom presjeku trbušne šupljine. Korišteni parametri za učenje na horizontalnom presjeku trbušne šupljine dani su Tablicom 6.2.

Po izvršenju 600 epoha učenja najbolji dobiveni model za detekciju karcinoma na horizontalnom presjeku trbušne šupljine ispitan je na skupu slika za ispitivanje, a zatim dalje korišten u procesu

učenja prepoznavanja tumora na sagitalnom presjeku trbušne šupljine. Korišteni parametri za učenje na sagitalnom presjeku trbušne šupljine dani su Tablicom 6.2.

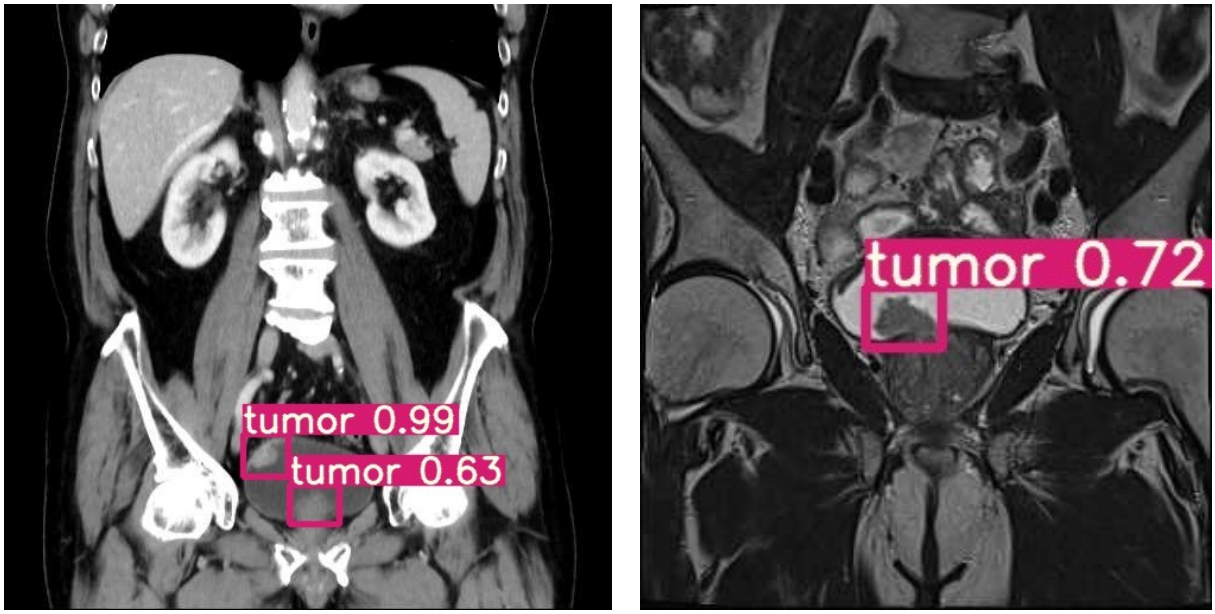
Po izvršenju 600 epoha učenja, najbolji dobiveni model za detekciju karcinoma na sagitalnom presjeku trbušne šupljine ispitan je na skupu podataka za ispitivanje.

U nastavku slijedi prikaz rezultata.

### 6.4. Rezultati

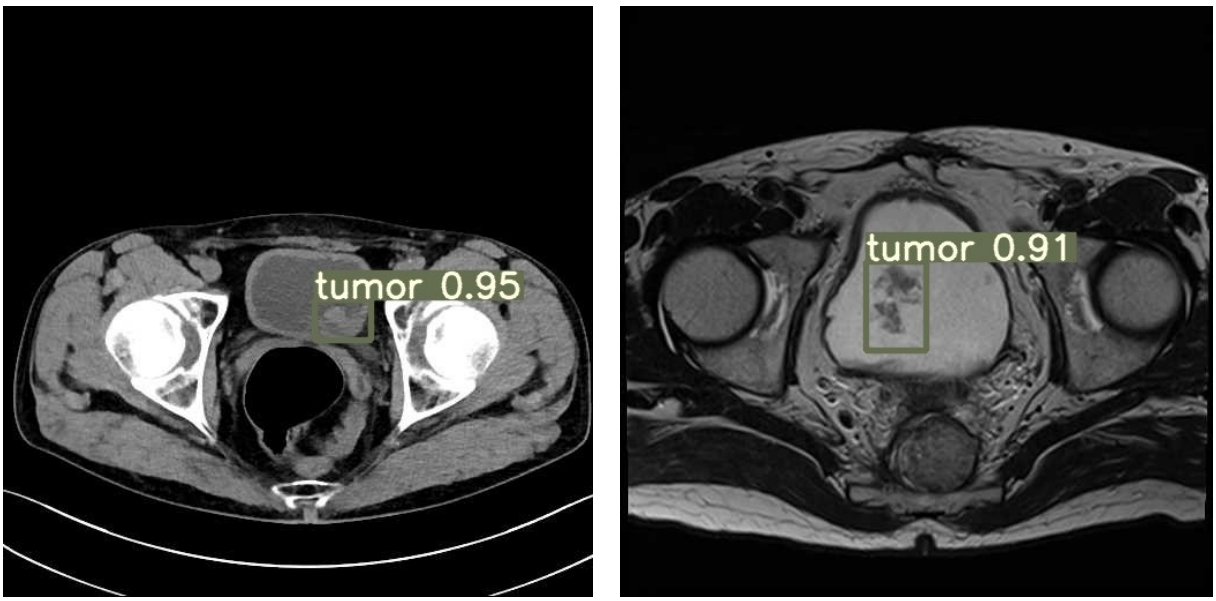
Po završetku procesa učenja ostvaren je $mAP@0.5$ u iznosu 94.4% za frontalni presjek, 85.6% za horizontalni presjek i 96.1% za sagitalni presjek što su izrazito dobri rezultati. Prikaz grafova koji pobliže prikazuju proces učenja u vidu gubitaka i *PR* krivulja dan je u Prilogu B.

Težinski parametri za pojedini presjek trbušne šupljine ispitani su na dotad neviđenim pripadajućim skupovima podataka za ispitivanje. Primjeri rezultata detekcije karcinoma mokraćnog mjehura na frontalnom, horizontalnom i sagitalnom presjeku trbušne šupljine dani su redom Slikom 6.6, Slikom 6.7 i Slikom 6.8.



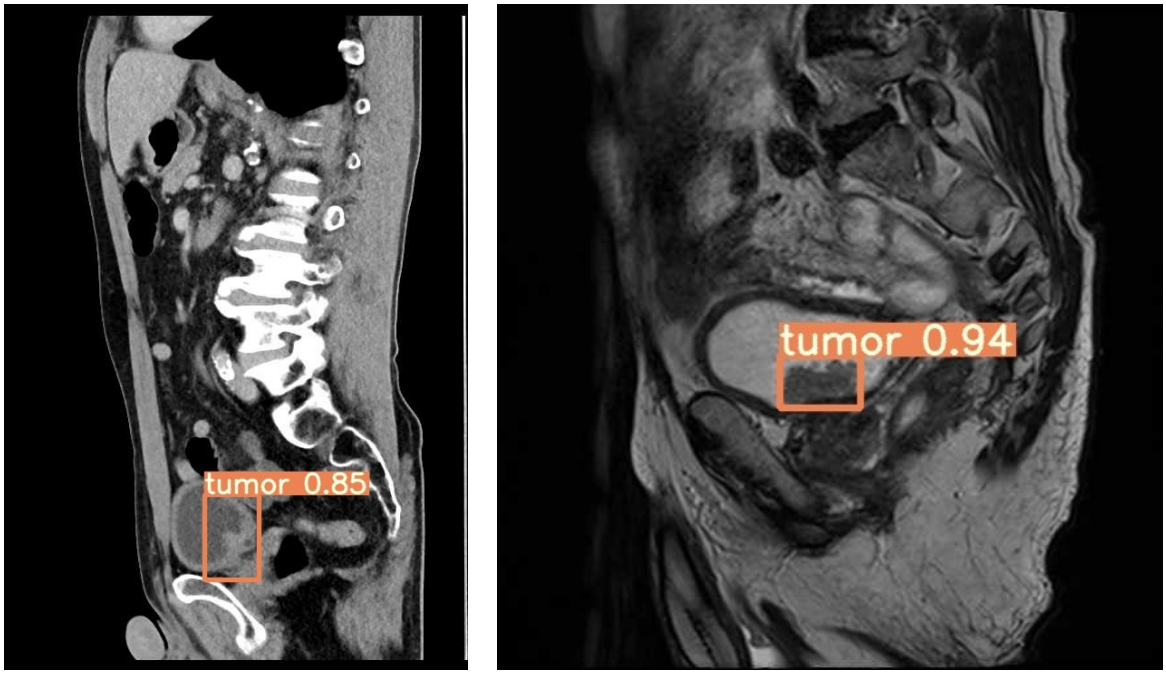

*a)* *b)*

*Slika 6.6. Detektirana prisutnost tumora mokraćnog mjehura na frontalnom presjeku trbušne šupljine: a) CT; b) MRI*




*a)* *b)*

*Slika 6.7. Detektirana prisutnost tumora mokraćnog mjehura na horizontalnom presjeku trbušne šupljine: a) CT; b) MRI*




*a)* *b)*

*Slika 6.8. Detektirana prisutnost tumora mokraćnog mjehura na sagitalnim presjeku trbušne šupljine: a) CT; b) MRI*

Primjena naučenih modela za detekciju tumora mokraćnog mjehura na sva tri presjeka trbušne šupljine urodila je i više nego dobrim rezultatima čime se izvršeno učenje modela može zaključiti uspješnim.

# 7. ZAKLJUČAK

Iako nerijetko pogrešno shvaćena i promatrana s dozom skepticizma, umjetna inteligencija igra središnju ulogu u razvoju modernog društva. Gotovo da ne postoji područje ljudskih života u koje već nije ušla, no daleko je najljepša kada se koristi za spašavanje života.

Medicina je jedno od područja koja su objeručke prihvatila umjetnu inteligenciju i sve njene prednosti kao što je pomoć liječnicima pri dijagnosticiranju i tretiranju raznih bolesti. Karcinom je jedna od njih.

Opisan je karcinom mokraćnog mjehura i dvije metode njegovoga dijagnosticiranja - kompjutorizirana tomografija i magnetska rezonanca na čijim se slikama mogu primijeniti algoritmi detekcije objekata za utvrđivanje prisutnosti tumora.

Donesena je problematika detekcije objekata kao posebnog područja računalnog vida koje se u današnje vrijeme temelji na uporabi konvolucijskih neuronskih mreža koje mogu učiti. Predstavljeni su neki od parametara pomoću kojih se njihov učinak u detekciji objekata vrednuje te skupovi podataka na kojima one uče. Zatim je dan pregled algoritama koji se koriste u detekciji objekata, a dijele se na one s pristupom u dva koraka i one s pristupom u jednom koraku.

Od brojnih algoritama jedan je od najistaknutijih YOLO (*You Only Look Once*) algoritam koji pripada algoritmima detekcije u jednom koraku. Predstavlja detekciju objekata kao regresijski problem, a u srpnju 2022. godine predstavljena je njegova najnovija inačica YOLOv7 koja slovi za trenutno najbrži i najtočniji algoritam za detekciju objekata.

YOLOv7 algoritam naučen je i primijenjen je za detektiranje tumora mokraćnog mjehura na CT i MRI snimkama frontalne, horizontalne i sagitalne osi trbušne šupljine pri čemu rezultira srednjom prosječnom preciznošću ($mAP$) od 94.4%, 85.6% i 96.1%, redom. Ovim rezultatima zaključeno je uspješno učenje i primjena YOLOv7 modela na detekciju karcinoma mokraćnog mjehura.

# 8. LITERATURA


[1] Gilbert, F.J.; Lemke, H.: "Computer-aided diagnosis", The British Journal of Radiology Vol. 78, pp. S1-S2, 2014.

[2] The University of Chicago: "Computer-aided Diagnosis/Machine Learning/AI", s Interneta, https://radiology.uchicago.edu/research/cad, 23. kolovoza 2022.

[3] MEFST: "Bladder Tumors and BPH", s Interneta https://neuron.mefst.hr/docs/katedre/urologija/2018-2019/ENG%20-%20Urology/BLADDER%20TUMORS%20and%20BPH.pdf , 4. kolovoza 2022.

[4] National Cancer Institute: "Dictionary of Cancer Terms", s Interneta, https://www.cancer.gov/publications/dictionaries/cancer-terms/def/bladder, 3. rujna 2022.

[5] Šitum, M.; Gotovac, J.: "Urologija", Medicinska naklada, Zagreb, 2011.

[6] Friedlander, T.: "Bladder Cancer", s Interneta, https://www.youtube.com/watch?v=tFEVZHu8Cv0, 15. lipnja 2022.

[7] Siegel, A.: "Bladder Cancer: What You Should Know", s Interneta, https://njurology.com/bladder-cancer-what-you-should-know/, 3. rujna 2022.

[8] MacVicar, D.; Reznek, R. H.: "Carcinoma of the Bladder", Cambridge University Press, New York, 2008.

[9] Mayo Clinic: "Bladder cancer", s Interneta, https://www.mayoclinic.org/diseases-conditions/bladder-cancer/symptoms-causes/syc-20356104, 18. veljače 2022.

[10] Affidea Hrvatska: "Višeslojna kompjutoriziranom tomografija MSCT", s Interneta, https://affidea.hr/usluga/viseslojna-kompjutorizirana-tomografija-msct/, 4. kolovoza 2022.

[11] Affidea Hrvatska: "Magnetska rezonanca MR", s Interneta, https://affidea.hr/usluga/magnetska-rezonanca-mr/, 4. kolovoza 2022.

[12] MC Frockman, J.: "Artificial Intelligence and Machine Learning", 2019.

[13] Chang, A.: "Common Misconceptions and Future Directions for AI in Medicine: A Physician-Data Scientist Perspective", Artificial Intelligence in Medicine, pp. 3-6, Springer International Publishing, 2019.

[14] Li, F.; Johnson, J.; Yeung, S.: "Computer Vision", s Interneta, http://cs231n.stanford.edu/slides/2017/, 6. kolovoza 2022.

[15] Zhiqiang, W.; Jun, L.: "A review of object detection based on convolutional neural network," 36th Chinese Control Conference (CCC), pp. 11104-11109, 2017.

[16] Bulić, F.J.: "Konvolucijske reprezentacije za dohvat slike na temelju sadržaja", Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2018.

[17] Jähne, B., Haußbecker, H.: "Computer Vision and Applications: A Guide for Students and Practitioners", Academic Press, San Diego, 2000.

[18] Wang, M.; Leelapatra, W.: "A Review of Object Detection Based on Convolutional Neural Networks and Deep Learning" International Scientific Journal of Engineering and Technology (ISJET), Vol. 6, No. 1, pp. 1–7, 2022.

[19] Szeliski, R: "Texts in Computer Science, Computer Vision, Algorithms and Applications", Springer, Cham, 2022.

[20] Dalbelo Bašić, B.; Čupić, M.; Šnajder, J.: "Umjetne neuronske mreže", repozitorij Fakulteta elektrotehnike i računarstva, Zagreb, 2008.

[21] MathWorks: "Object Detection", s Interneta, https://www.mathworks.com/discovery/object-detection.html, 18. veljače 2022.

[22] Dhillon, A.; Verma, G. K.: "Convolutional neural network: a review of models, methodologies and applications to object detection", Progress in Artificial Intelligence, Vol. 9, pp. 85–112, 2020.

[23] Zhao, L.; Li, S.: "Object Detection Algorithm Based on Improved YOLOv3", Electronics, Vol. 9, No. 3: 537, 2020.

[24] Car, Z.: "Primjena umjetne inteligencije", predavanja, 2021./2022.

[25] Srivastava, S. i dr.: "Comparative analysis of deep learning image detection algorithms", Journal of Big Data, Vol. 8, No. 1, 2021.

[26] Sultana, F.; Sufian, A.; Dutta, P.: "A Review of Object Detection Models based on Convolutional Neural Network", Intelligent Computing: Image Processing Based Applications. Advances in Intelligent Systems and Computing, Vol. 1157, 2020.

[27] Galvez, R. L. i dr.: "Object Detection Using Convolutional Neural Networks", Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference, pp. 2012-2016, Jeju, 2018.

[28] ImageNet, s Interneta, https://www.image-net.org/, 16. kolovoza 2022.

[29] Kang, D.; Duong, P.; Park, J.; "Application of Deep Learning in Dentistry and Implantology", The Korean Academy of Oral and Maxillofacial Implantology, Vol. 24, pp. 148-181, 2020.

[30] Swapna, K. E.: "Convolutional Neural Network", s Interneta, https://developersbreach.com/convolution-neural-network-deep-learning/, 15. kolovoza 2022.

[31] IBM, s Interneta, https://www.ibm.com/cloud/learn, 17. kolovoza 2022.

[32] Practice Probs: "Evaluation Metrics and Loss Function, Precision and Recall", s Interneta, https://www.practiceprobs.com/problemsets/evaluation-metrics-and-loss-functions/precision-and-recall/#, 28. kolovoza 2022.

[33] PASCAL VOC, s Interneta, http://host.robots.ox.ac.uk/pascal/VOC/, 16. kolovoza 2022.

[34] MS COCO, s Interneta, https://cocodataset.org, 16. kolovoza 2022.

[35] Girshick, R. i dr.: "Rich feature hierarchies for accurate object detection and semantic segmentation", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013.

[36] He, K. i dr.: "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 37, No. 9, 2014.

[37] Girshick, R.: "Fast R-CNN", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013.

[38] Ren, S. i dr.: " Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 6, 2015.

[39] He, K. i dr.: "Mask R-CNN", 2017.

[40] Redmon, J. i dr.: "You Only Look Once: Unified, Real-Time Object Detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[41] Liu, W. i dr: "SSD- Single Shot Multibox Detector", European Conference on Computer Vision, pp. 21-37, 2016.

[42] Lin, T.: "Focal Loss for Dense Object Detection", IEEE International Conference on Computer Vision (ICCV), pp. 2999-3007, 2017.

[43] Redmon, J.; Farhadi, A.: "YOLO9000: Better, Faster, Stronger", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525, 2017.

[44] Redmon, J.; Farhadi, A.: "YOLOv3: An Incremental Improvement", 2018.

[45] Bochkovskiy, A.; Wang, C.; Liao, H. M.: "YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020.

[46] Xu, P. i dr.: "On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning", Remote Sens, Vol. 13, No. 10, 2021.

[47] Wang, C.; Bochkovskiy, A.; Liao, H. M.: "Scaled-YOLOv4: Scaling Cross Stage Partial Network", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13024-13033, 2021.

[48] Wang, C.; Yeh, I.; Liao, H. M.: "You Only Learn One Representation: Unified Network for Multiple Tasks", 2021.

[49] Leksikografski zavod Miroslav Krleža, s Interneta, https://www.enciklopedija.hr/znanje, 18. kolovoza 2022.

[50] Wang, C.; Bochkovskiy, A.; Liao, H. M.: "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", 2022.

[51] Wang, C.: "yolov7", s Interneta, https://github.com/WongKinYiu/yolov7, 5. kolovoza 2022.

[52] Cvija, T., Severinski, K.: "Urinary bladder cancer detection using YOLOv5 algorithm", RI-STEM 2022 Proceedings, 26-28, 2022.

# 9. SAŽETAK I KLJUČNE RIJEČI

Ovaj diplomski rad opisuje detekciju karcinoma mokraćnog mjehura primjenom YOLO (*You Only Look Once*) algoritma. Opisana je problematika računalnog vida i detekcije objekata te uloga konvolucijskih neuronskih mreža pri rješavanju iste. Konvolucijske neuronske mreže opisane su u pogledu strukture, procesa učenja i vrednovanja detekcije objekata. Predstavljena su rješenja detekcije objekata temeljena na konvolucijskim neuronskim mrežama u obliku algoritama s pristupom u dva i u jednom koraku. Iz algoritama s pristupom u jednom koraku izdvojen je i pobliže opisan YOLO algoritam s naglaskom na svoju najnoviju inačicu YOLOv7. YOLOv7 model učen je detekciji tumora na mokraćnom mjehuru na skupovima podataka koji su se sastojali od CT i MRI slikovnih zapisa frontalnog, horizontalnog i sagitalnog presjeka trbušne šupljine. Učenje za pojedini presjek vršeno je odvojeno i rezultiralo je srednjom prosječnom preciznošću od 94.4%, 85.6% i 96.1% za frontalni, horizontalni i sagitalni presjek, redom. Ispitivanjem modela na novim, modelu nepoznatim, slikama utvrđena je uspješna detekcija karcinoma mokraćnog mjehura.

**Ključne riječi:** umjetna inteligencija, duboko učenje, računalni vid, detekcija objekata, konvolucijske neuronske mreže, YOLO algoritam, YOLOv7

# 10. SUMMARY AND KEY WORDS

This master's thesis describes urinary bladder cancer detection using YOLO (*You Only Look Once*) algorithm. It describes the problem of computer vision, object detection, and the role of using convolutional neural networks as its solution. Convolutional neural networks are described in the means of their structure, training process, and object detection evaluation. Solutions of object detection based on convolutional neural networks are introduced as one- and two-stage approach algorithms. From one-stage approach algorithms YOLO algorithm has been chosen and described in more detail with emphasis on its newest version YOLOv7. A YOLOv7 model has been trained to detect tumor on the urinary bladder by using datasets that consisted of CT and MRI images of frontal, horizontal and sagittal plane of the abdomen. Learning process for each plane has been performed separately and it has resulted in 94.4%, 85.6%, and 96.1% mean average precision for frontal, horizontal and sagittal plane, respectively. Model testing on new, to the model unknown, images concluded in successful detection of the urinary bladder cancer.

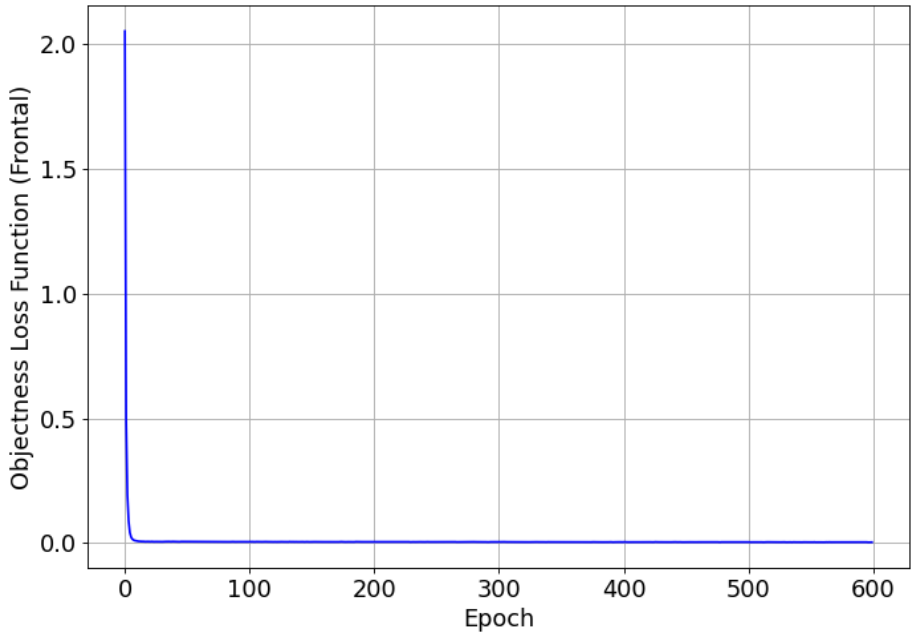
**Keywords:** artificial intelligence, deep learning, computer vision, object detection, convolutional neural networks, YOLO algorithm, YOLOv7

# 11. PRILOZI

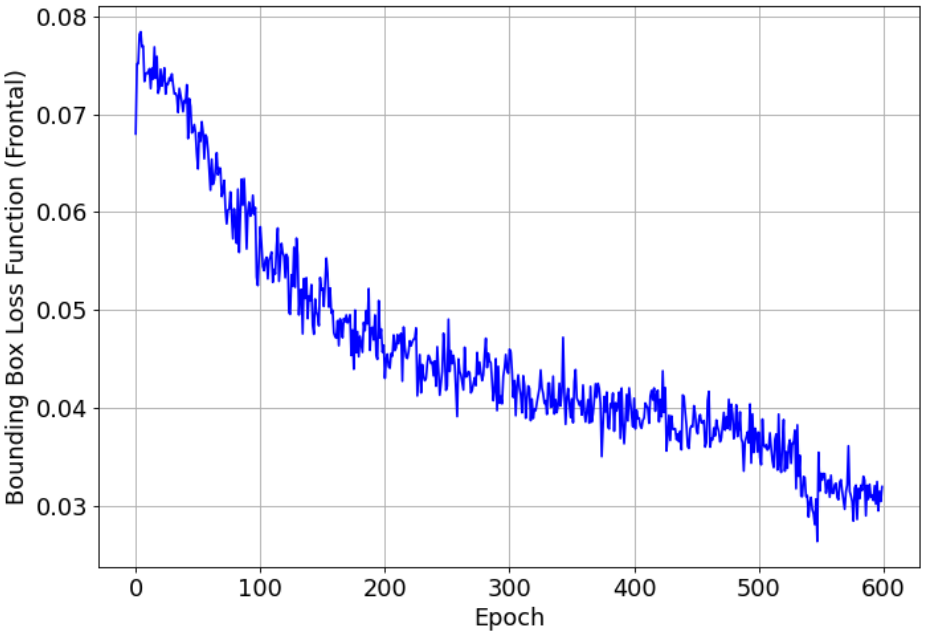
Prilog A. Kod za učenje i ispitivanje modela

```
#uvoz potrebnih modula i frameworka
import sys
import torch
print(f"Python version: {sys.version}, {sys.version_info} ")
print(f"Pytorch version: {torch.__version__} ")

#upravljanje i nadzor GPU-a
!nvidia-smi

#uvoz osobnog Google Drivea
from google.colab import drive
drive.mount('/content/drive')

# Preuzimanje YOLOv7 koda
!git clone https://github.com/WongKinYiu/yolov7
%cd yolov7
!ls

# Preuzimanje predtreniranih utega
!wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt335

#instaliranje potrebnih knjižnica i modula za YOLOv7
!pip install -r requirements.txt

#uputa algoritmu da je input grayscale
USE_GRAY_INPUT=1

# učenje detekcije na FRONTALNOM presjeku abdomena
!python train.py --workers 8 --device 0 --batch-size 16 --
data /content/drive/MyDrive/yolov7Files/custom_data_frontal.yaml --
cfg /content/drive/MyDrive/yolov7Files/yolov7.yaml --
weights /content/drive/MyDrive/yolov7Files/runs/yolov7-frontal2/weights/best.pt --
name /content/drive/MyDrive/yolov7Files/runs/yolov7-frontal --
hyp data/hyp.scratch.p5.yaml  --epoch 600

# učenje detekcije na HORIZONTALNOM presjeku abdomena
!python train.py --workers 8 --device 0 --batch-size 16 --
data /content/drive/MyDrive/yolov7Files/custom_data_horizontal.yaml --
cfg /content/drive/MyDrive/yolov7Files/yolov7.yaml --
weights /content/drive/MyDrive/yolov7Files/runs/yolov7-frontal3/weights/best.pt --
name /content/drive/MyDrive/yolov7Files/runs/yolov7-horizontal --
hyp data/hyp.scratch.custom.yaml --epoch 600
```

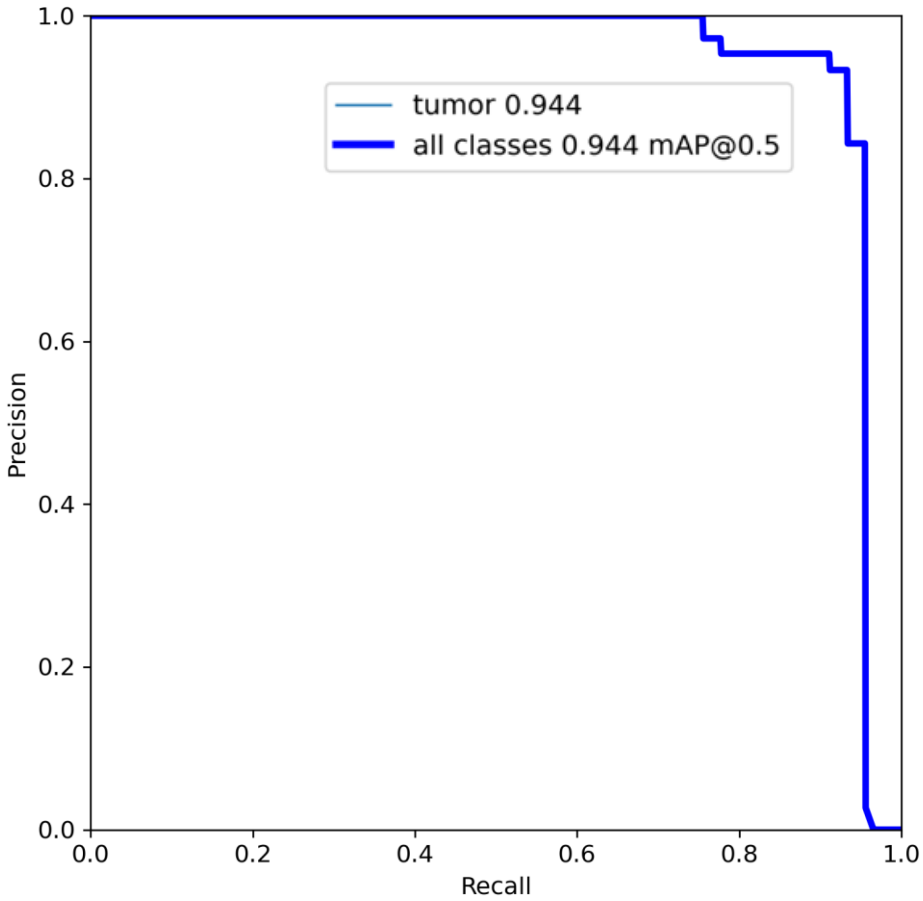
```
# učenje detekcije na SAGITALNOM presjeku abdomena
!python train.py --workers 8 --device 0 --batch-size 16 --
data /content/drive/MyDrive/yolov7Files/custom_data_sagittal.yaml --
cfg /content/drive/MyDrive/yolov7Files/yolov7.yaml --
weights /content/drive/MyDrive/yolov7Files/runs/yolov7-frontal3/weights/best.pt --
name /content/drive/MyDrive/yolov7Files/runs/yolov7-sagittal --
hyp data/hyp.scratch.custom.yaml --epoch 600

#Ispitivanje detekcije na FRONTALNOM presjeku abdomena na skupu podataka za TESTIRANJE
!python detect.py --weights /content/drive/MyDrive/yolov7Files/runs/yolov7-
frontal3/weights/best.pt --img 640 --conf 0.25 --
source /content/drive/MyDrive/images/frontal_scan_test --
name /content/drive/MyDrive/yolov7Files/detect/yolov7-frontal-test

#Ispitivanje detekcije na FRONTALNOM presjeku abdomena na skupu podataka za VALIDACIJU
!python detect.py --weights /content/drive/MyDrive/yolov7Files/runs/yolov7-
frontal3/weights/best.pt --img 640 --conf 0.25 --
source /content/drive/MyDrive/images/frontal_scan_val --
name /content/drive/MyDrive/yolov7Files/detect/yolov7-frontal-val

#Ispitivanje detekcije na HORIZONTALNOM presjeku abdomena na skupu podataka za TESTIRA
NJE
!python detect.py --weights /content/drive/MyDrive/yolov7Files/runs/yolov7-
horizontal5/weights/best.pt --img 640 --conf 0.25 --
source /content/drive/MyDrive/images/horizontal_scan_test  --
name /content/drive/MyDrive/yolov7Files/detect/yolov7-horizontal-test

#Ispitivanje detekcije na HORIZONTALNOM presjeku abdomena na skupu podataka za VALIDAC
IJU
!python detect.py --weights /content/drive/MyDrive/yolov7Files/runs/yolov7-
horizontal5/weights/best.pt --img 640 --conf 0.25 --
source /content/drive/MyDrive/images/horizontal_scan_val --
name /content/drive/MyDrive/yolov7Files/detect/yolov7-horizontal-val

#Ispitivanje detekcije na SAGITALNOM presjeku abdomena na skupu podataka za TESTIRANJE
!python detect.py --weights /content/drive/MyDrive/yolov7Files/runs/yolov7-
sagittal/weights/best.pt --img 640 --conf 0.25 --
source /content/drive/MyDrive/images/sagittal_scan_test --
name /content/drive/MyDrive/yolov7Files/detect/yolov7-sagittal-test

#Ispitivanje detekcije na SAGITALNOM presjeku abdomena na skupu podataka za VALIDACIJU
!python detect.py --weights /content/drive/MyDrive/yolov7Files/runs/yolov7-
sagittal/weights/best.pt --img 640 --conf 0.25 --
source /content/drive/MyDrive/images/sagittal_scan_val --
name /content/drive/MyDrive/yolov7Files/detect/yolov7-sagittal-val
```



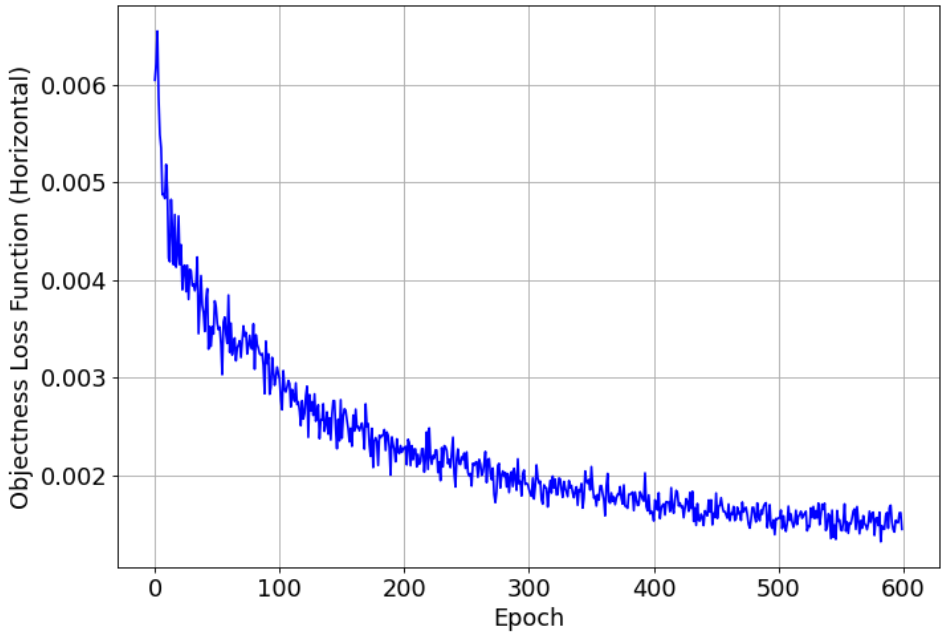

*Slika B.1. Objectness Loss (frontalni presjek)*



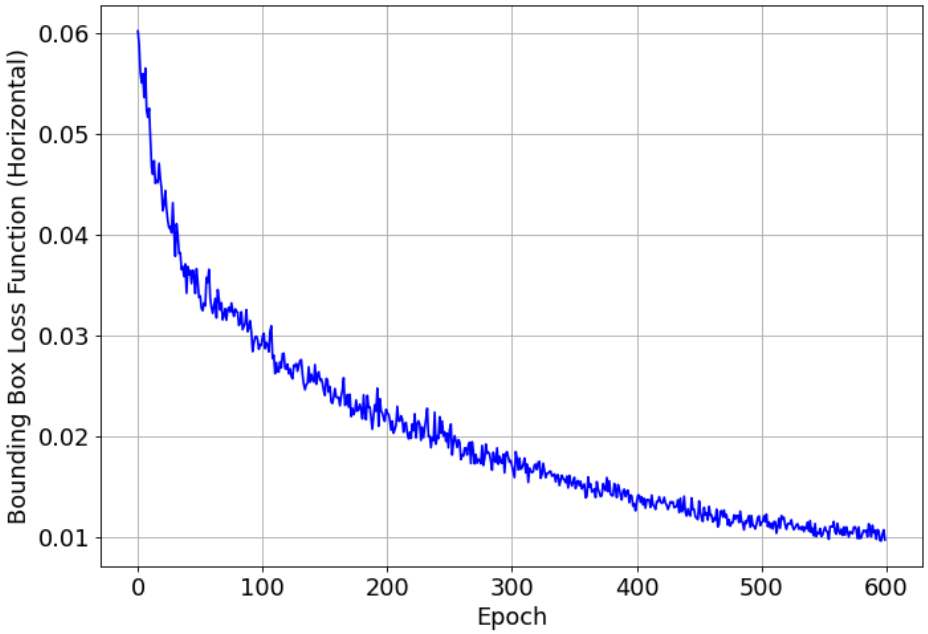

*Slika B.2. Bounding Box Loss (frontalni presjek)*



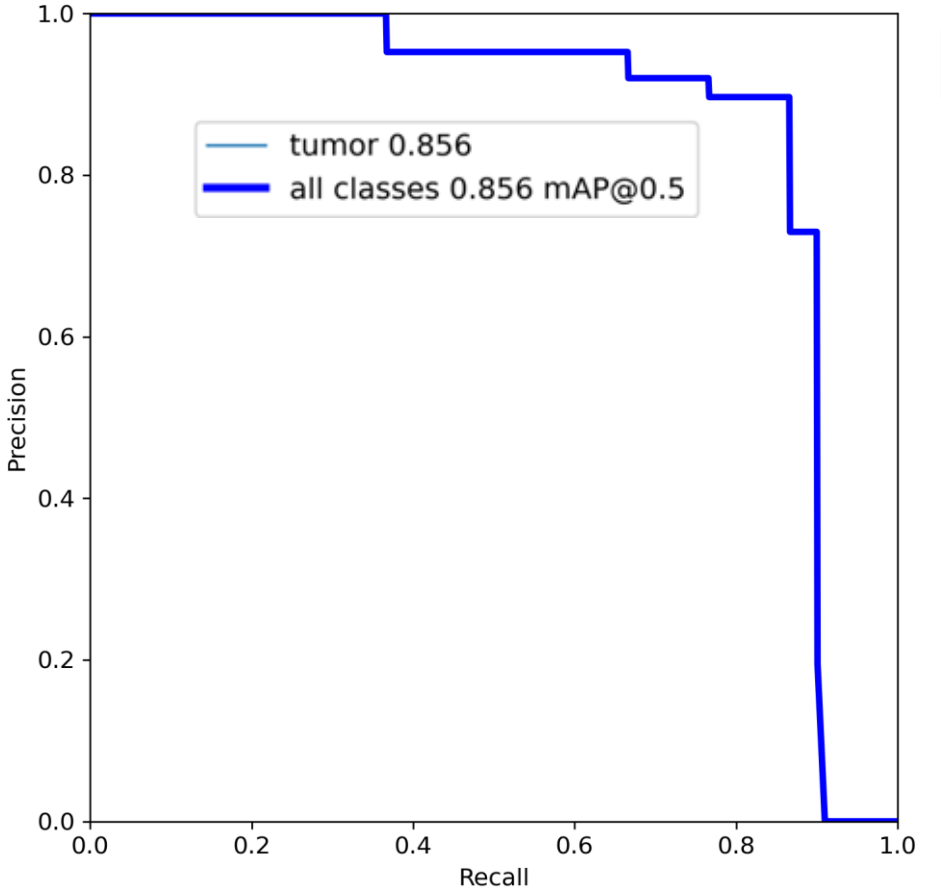

*Slika B.3. PR curve (frontalni presjek)*



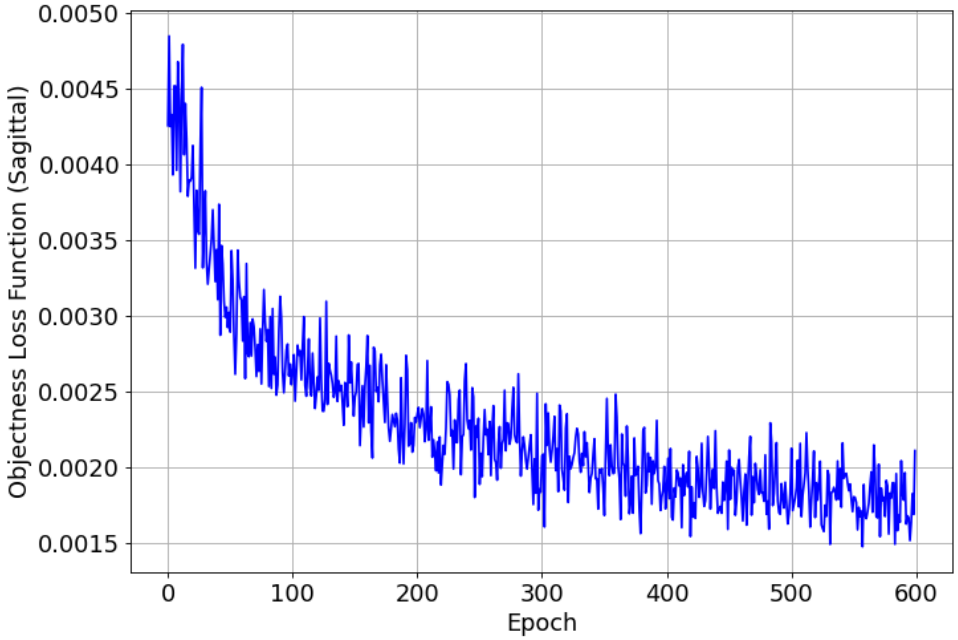

*Slika B.4. Objectness Loss (horizontalni presjek)*



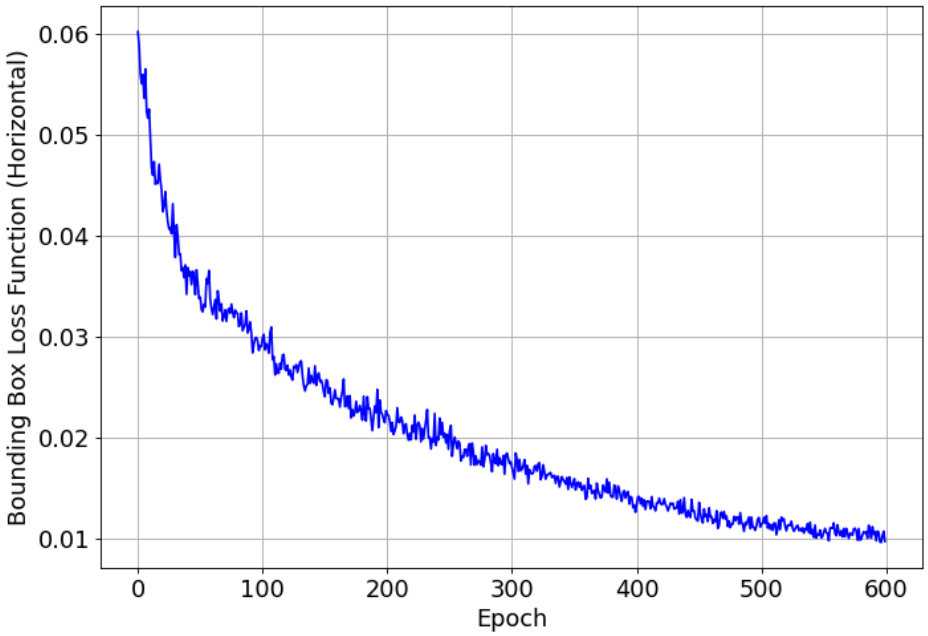

*Slika B.5. Bounding Box Loss (horizontalni presjek)*



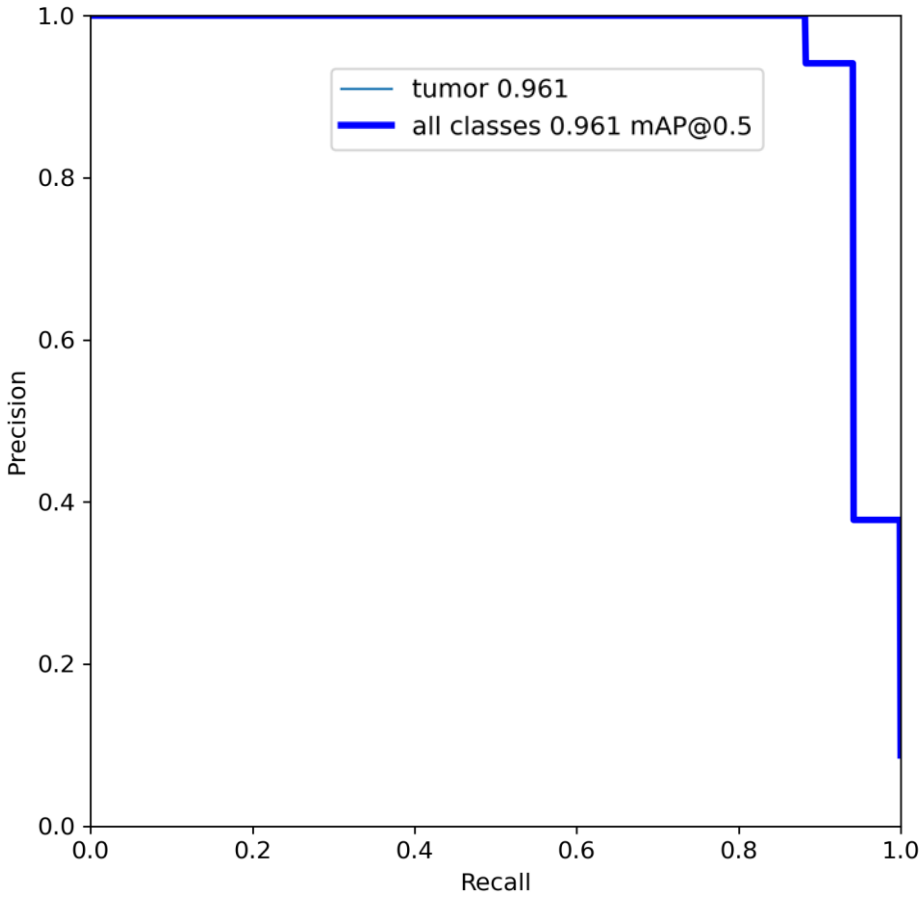

*Slika B.6. PR curve (horizontalni presjek)*




*Slika B.7. Objectness Loss (sagitalni presjek)*




*Slika B.8. Bounding Box Loss (sagitalni presjek)*




*Slika B.9. PR curve (sagitalni presjek)*