Zahvaljujem mentoru prof. dr. sc. Siniši Šegviću na izdvojenom vremenu i stručnim savjetima tijekom cijelog studija i pisanja diplomskog rada.

Hvala kolegama Mateju Šikiću Vagiću i Karlu Lochertu na suradnji na projektu u sklopu programa SPOCK čija je tema usko vezana uz temu i sadržaj ovoga rada.

Sadržaj

Uvod 1

1. Konvolucijske neuronske mreže 2

1.1. Konvolucijski slojevi 2

1.2. Sloj sažimanja 4

2. Detekcija objekata 6

2.1. Opis problema 6

2.2. Usporedba jednoprolaznih i dvoprolaznih detektora 6

3. Jednoprolazni detektori 8

3.1. You Only Look Once (YOLO) 8

3.1.1. Osnovni koncept modela YOLO 8

3.1.2. CSPDarknet-53 9

3.1.3. Prostorno piramidalno sažimanje 11

3.1.4. Arhitektura FPN 12

3.1.5. PANet 13

3.1.6. YOLO detekcijska glava 14

3.1.7. Konačna arhitektura modela YOLOv5 15

3.1.8. Funkcija gubitka u modelu YOLOv5 15

4. Dvoprolazni detektori 17

4.1. Arhitektura R-CNN 17

4.2. Arhitektura Fast R-CNN 19

4.3. Arhitektura Faster R-CNN 21

4.3.1. Arhitektura mreže ResNet-50 21

4.3.2. Arhitektura RPN 23

4.3.3. Mjera preklapanja okvira - $\boldsymbol{IoU}$ kod RPN modula 25

4.3.4. Učenje modela Faster R-CNN 26

4.4. Algoritam NMS 28
5. Programska implementacija 30
5.1. Priprema skupa podataka 30
5.2. Sastav vlastitog skupa podataka 31
5.3. Biblioteka Detectron2 32
6. Eksperimenti i rezultati na vlastitom skupu podataka 34
6.1. Evaluacijske mjere 34
6.2. Eksperimenti nad arhitekturom YOLOv5 37
6.2.1. Učenje svih slojeva mreže 37
6.2.2. Učenje sa zamrzavanjem kralježnice modela 38
6.2.3. Učenje sa zamrzavanjem svih osim posljednjeg sloja 40
6.2.4. Usporedba dvaju eksperimenata na mreži YOLOv5x 41
6.3. Eksperimenti nad arhitekturom Faster R-CNN 43
6.3.1. Učenje svih slojeva mreže 43
6.3.2. Učenje sa zamrznutom kralježnicom modela 43
6.3.3. Usporedba dvaju eksperimenata na mreži Faster R-CNN 44
6.4. Usporedba dobivenih rezultata dvaju modela 45
Zaključak 47
Literatura 48
Sažetak 50

Uvod

Računalni vid grana je dubokog učenja čiji je zadatak prepoznati, analizirati i razumjeti sadržaj pojedine slike izvlačenjem značajki pomoću dubokih modela koji se mogu predstaviti kompozicijom nelinearnih transformacija. Neki od problema kojima se područje računalnog vida bavi su klasifikacija koja podrazumijeva svrstavanje slika u razrede, semantička segmentacija koja pridjeljuje klasu svakom pikselu na slici te detekcija objekata čiji je cilj odrediti koji objekti od definiranih klasa su prisutni na slici i gdje se na njoj nalaze. Ovaj rad bavi se upravo detekcijom objekata.

Temelj dubokih modela za detekciju objekata su konvolucijske neuronske mreže. One omogućuju dobivanje pogodnijih reprezentacija ulazne slike stvaranjem mapa značajki, ali na račun smanjenja rezolucije slike, čime se otežava mogućnost detektiranja objekta. Rješenje problema kompromisa između rezolucije i semantike mapa značajki donose arhitekture za piramidalno izlučivanje značajki koje učenjem na više mjerila iskorištavaju pogodnosti oba svijeta.

Modeli za detekciju objekata koji čine današnje stanje tehnike mogu se podijeliti na jednoprolazne i dvoprolazne detektore. Dvoprolazni detektori su općenito vrlo precizni, no njihov je problem nedovoljna brzina zaključivanja koja im smanjuje mogućnost korištenja u realnom vremenu. Jednoprolazni modeli drastično ubrzavaju vrijeme koje je potrebno za detekciju objekata.

U radu su opisani jednoprolazni detektor YOLOv5 te dvoprolazni model Faster R-CNN. Objašnjen je njihov način funkcioniranja te je detaljno obrazložena njihova arhitektura. U eksperimentalnom dijelu rada proveden je postupak učenja i evaluacije navedena dva modela na vlastitom skupu podataka te je napravljena usporedba među rezultatima dobivenima mjerenjem performansi u pogledu uspješnosti detekcije osoba i brzine izvođenja arhitektura YOLOv5 i Faster R-CNN.

1. Konvolucijske neuronske mreže

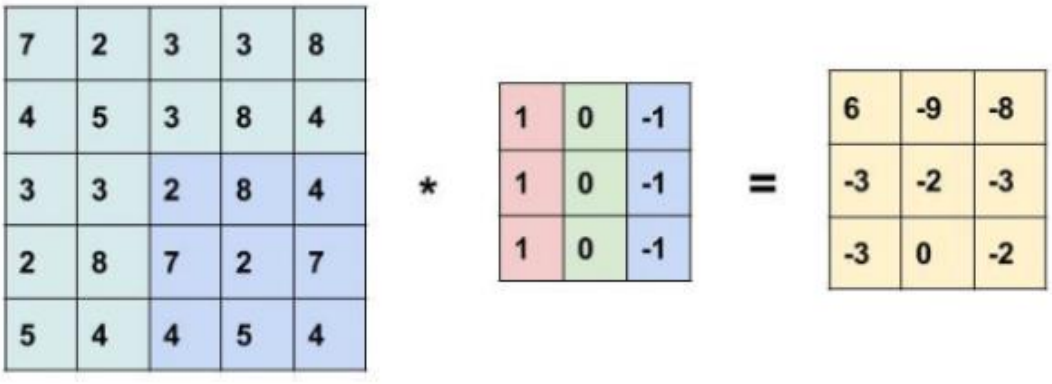
Neuronske mreže čije skrivene slojeve čine potpuno povezani slojevi nisu prikladne za probleme kojima se bavi računalni vid. Razlog tomu je taj što ovakve modele karakterizira veliki broj parametara koji proizlazi iz velikog broja poveznica među neuronima. Potpuno povezane arhitekture zbog toga mogu dovesti do sporog učenja modela i prenaučenosti (engl. *overfitting*) koji uzrokuje smanjenje mogućnosti generaliziranja modela. Ovaj problem pogotovo je uočljiv kada se radi o učenju nad podacima velikih dimenzija kao što su slike koje mogu biti visoke rezolucije. S obzirom na to da su ulazne vrijednosti modela svi pikseli slike, podrazumijevano je da je na ulazu u mrežu $C \cdot H \cdot W$ vrijednosti, pri čemu su H i W redom visina (engl. *height*) i širina (engl. *width*) slike, a C broj kanala (engl. *channel*) koji je kod slika u RGB formatu jednak tri, a kod crno-bijelih slika jednak jedan. Iz ovoga se može vidjeti da bi, ukoliko se koriste potpuno povezani slojevi, već na samom početku mreže bilo izrazito puno parametara. Stoga ovakvo rješenje nije prikladno.

Tip umjetnih neuronskih mreža koji rješava navedene probleme potpuno povezanih modela jesu konvolucijske neuronske mreže. U njima je naglasak na konvolucijskim slojevima koji izvlače pojedine značajke iz slika i latentnih reprezentacija uz manji broj parametara od potpuno povezanih slojeva. Uz njih, konvolucijske neuronske mreže sadrže i slojeve sažimanja i potpuno povezane slojeve, pri čemu se najčešće naizmjenično pojavljuju grupe konvolucijskih slojeva nakon kojih slijedi sloj sažimanja, dok je potpuno povezani slojevi uobičajeno posljednji sloj u mreži. U nastavku su opisani i objašnjeni navedeni slojevi koji grade konvolucijske neuronske mreže.

1.1. Konvolucijski slojevi

Konvolucijski slojevi (engl. *convolution layers*) najvažniji su slojevi konvolucijske neuronske mreže. Parametri konvolucijskog sloja sastoje se od skupa filtera, odnosno jezgri. Operacija konvolucije zapravo je prolazak jezgre preko cijele ulazne slike, pri čemu jezgra podrazumijevano ima male dimenzije visine i širine. Prolaskom jezgre po slici, vrijednosti piksela slike koji se u danom trenutku poklapaju s jezgrom, množe se skalarno s odgovarajućim vrijednostima same jezgre te se dobiveni umnošci zbroje. Time se dobiva vrijednost koja se upisuje u pripadajuću ćeliju koja će biti dio izlaza iz sloja. Nakon toga, jezgra se pomiče udesno po ulaznoj slici za određen broj piksela koji čini korak (engl. *stride*)

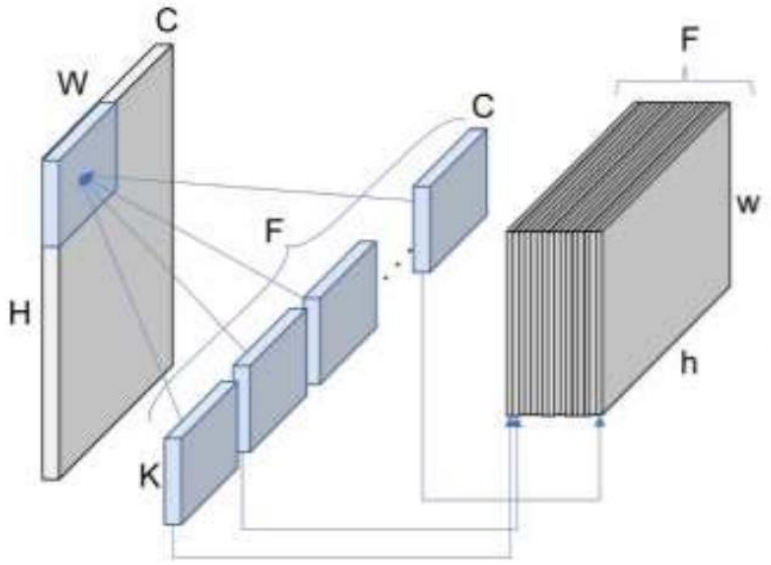
te se opisani postupak ponavlja. Potrebno je proći po cijeloj ulaznoj slici da bi se dobio konačan izlaz, odnosno mapa značajki određenih dimenzija, ovisno o veličini ulazne slike i jezgre. Bitno je uočiti kako se operacijom konvolucije smanjuje prostorna dimenzija mape značajki. Slika 1.1 prikazuje vizualni prikaz operacije konvolucije.



Slika 1.1: Prikaz operacije konvolucije na jednom primjeru. (preuzeto dana 22.05.2022. s: http://www.zemris.fer.hr/~ssegvic/project/pubs/bratulic20bs.pdf)

Jezgra konvolucijskog sloja male je dimenzije i najčešće je neparna (3, 5 ili 7), a određuju je četiri parametra – dubinama (engl .*depth*) ulazne i izlazne mape značajki, već spomenutim korakom, te nadopunjavanjem nulama (engl. *zero-padding*). Dubina jezgre definira semantičku dimenziju izlazne mape značajki, a veličina koraka, kao i sama konvolucija, utječe na izlaznu prostornu dimenziju. Spomenuto nadopunjavanje nulama služi za održavanje izvorne prostorne dimenzije mape značajki te ono određuje za koliko će se proširiti ulazna mapa značajki, pri čemu se novoproširenim značajkama koje se dodaju na rubove, pridjeljuje vrijednost nula kako i sam naziv sugerira.

Broj kanala, odnosno dubina jezgre, odgovara dubini ulaza u konvolucijski sloj, pa je time određeno da je dubina izlaza kada se koristi jedna jezgra jednaka jedan. Ipak, u konvolucijskim slojevima ne koristi se samo jedna jezgra, već više njih. Iz ovoga slijedi da dubina izlazne mape značajki odgovara broju primijenjenih jezgri. Vizualni prikaz ovoga ilustriran je na slici 1.2 dolje.



Slika 1.2: Vizualni prikaz konvolucije uz primjenu više jezgara. (preuzeto dana 22.05.2022. s: http://pabloruizruiz10.com/resources/CNNs/Convolution_Pooling.pdf)

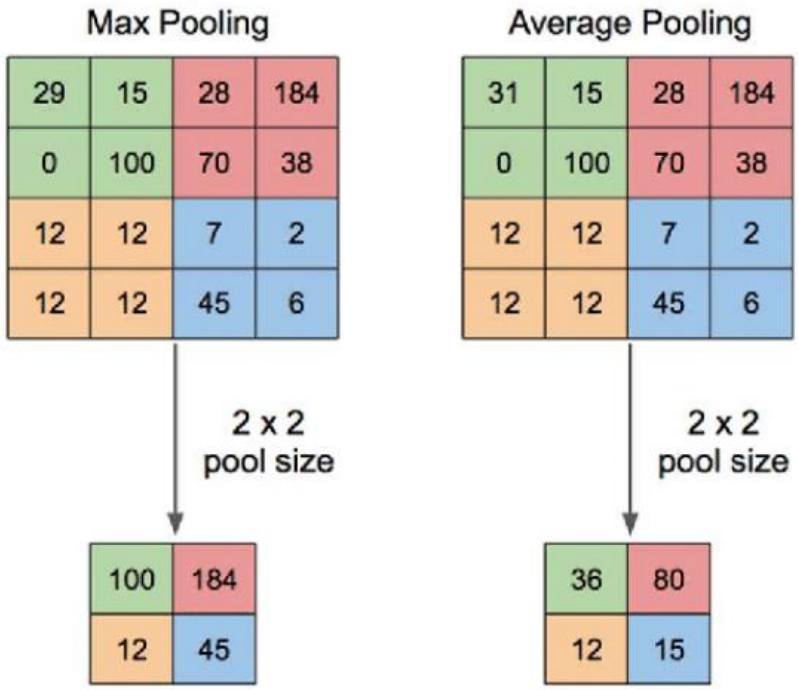
Svrha operacije konvolucije je ekstrahiranje i uočavanje značajki i pojedinih uzoraka na slikama promatranjem odnosa bliskih neurona.

Konvolucijski slojevi imaju svojstvo ekvivarijantnosti na translaciju, što omogućuje opisivanje složenih lokalnih međuovisnosti malim brojem parametara. Ovo svojstvo prikladno je za složene podatke s rešetkastom topološkom strukturom poput slika, teksta, govora, itd.

1.2. Sloj sažimanja

Sloj sažimanja (engl. *pooling layer*) u konvolucijskim neuronskim mrežama ima svrhu reduciranja broja parametara mreže na način da smanjuje prostorne dimenzije ulaza u sloj. Ovime se umanjuju i vremenska i prostorna složenost postupka učenja dubokog modela. Najčešće korišteni načini sažimanja su sažimanje srednjom vrijednošću (engl. *average pooling*) te sažimanje maksimalnom vrijednošću (engl. *max pooling*). Kao i kod operacije konvolucije, jezgra se pomiče po ulazu, no pri tome se u svakom koraku primjenjuje operacija maksimuma ili uprosječavanja ovisno o načinu sažimanja. Dakle, svakim pomakom se u pripadajuću ćeliju izlaza zapisuje najveća ili prosječna vrijednost uzevši u obzir vrijednosti ulaza koje se u danom trenutku prostorno podudaraju s jezgrom. Sažimanje se često provodi s jezgrom širine i visine jednake dva te s korakom jednakim dva kako ne bi

dolazilo do preklapanja prilikom pomicanja jezgre po ulazu. Sažimanje se događa na svim dubinama ulaza neovisno o ostalim dubinama i stoga dubina mape značajki ostaje jednaka kao na ulazu u sloj. Ilustracija operacije sažimanja prikazana je na slici 1.3 dolje.

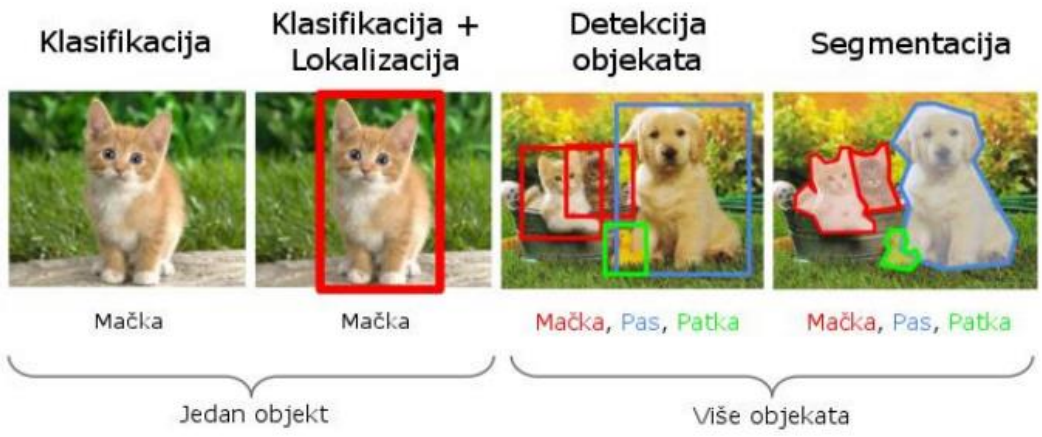


Slika 1.3: Sažimanje maksimalnom (lijevo) i srednjom vrijednošću (desno). (preuzeto dana 22.05.2022. s: https://qph.fs.quoracdn.net/main-qimg-939c3123c48e27301f1a89c0a299dca8)

2. Detekcija objekata

2.1. Opis problema

Rješavanje problema detekcije objekata na slikama može se razdijeliti u četiri kategorije problema. Prvi od njih je sama klasifikacija objekata, koja podrazumijeva pridjeljivanje razreda slici ovisno o tome koji se objekt nalazi na njoj. Problem klasifikacije je najjednostavniji problem od prethodno spomenute četiri kategorije, no nije trivijalan ukoliko je broj klasa koje međusobno treba razlikovati velik. Druga od navedene četiri kategorije je klasifikacija u kombinaciji s lokalizacijom. Ova vrsta problema računalnog vida obuhvaća klasifikaciju i definiranje lokacije objekta toga razreda na slici, odnosno određivanje gdje se objekt točno nalazi. Treća kategorija je detekcija objekata, koja je vrlo slična klasifikaciji uz lokalizaciju, no uz poopćenje da se na slici mora pronaći svaki objekt koji pripada bilo kojem od razreda iz podatkovnog skupa. Ovaj rad će se detaljnije baviti upravo ovom vrstom problema. Konačno, četvrta kategorija problema je segmentacija primjeraka – pridjeljivanje razreda svakom pojedinom pikselu na slici. [1] Slikovno objašnjenje navedenih kategorija vidljivo je na slici dolje.



Slika 2.1: Kategorije detekcije objekata na slikama od najlakše do najteže (prema desno). [1]

2.2. Usporedba jednoprolaznih i dvoprolaznih detektora

U današnje vrijeme koriste se dvije vrste detektora. To su jednoprolazni i dvoprolazni detektori. Jednoprolazni detektori funkcioniraju na način da problemu detekcije objekata pristupaju kao regresijskom problemu. Naime, takvi detektori iz ulazne slike uče

vjerojatnosti pojedinih razreda i koordinate okvira objekata. Za razliku od njih, dvoprolazni detektori imaju dvije faze, od kojih prva koristi modul za predlaganje regija (RPN) kako bi se izlučile regije od interesa (engl. *regions of interest*). Zatim, druga faza iskorištava navedene regije radi klasifikacije te regresije okvira. Dvoprolaznih detektori u obje faze provode regresiju okvira, pri čemu prva faza daje djelomične poravnate kandidate, dok druga faza služi da se okviri bolje prilagode oznakama.

Logično je zaključiti da dodatan prolaz dvoprolazne detektore čini sporijima, no oni u pravilu imaju veću točnost od jednoprolaznih detektora. Ovisno o tome je li za primjenu modela bitnija točnost ili brzina, potrebno je odlučiti se za prikladan tip detektora. Još jedna prednost dvoprolaznih detektora koja proizlazi iz generiranja regija od interesa je to što dvoprolazni detektori u drugoj fazi moraju raditi s puno manje okvira nego jednoprolazni modeli, upravo zbog toga što imaju manje negativnih kandidata. Ova činjenica povlači i pogodnost da glavni dio mreže, onaj za klasifikaciju i regresiju, može izlučiti više bogatijih značajki ako se poveća. [2]

U sljedećem poglavlju bit će detaljnije objašnjene najpoznatije arhitekture jednoprolaznih i dvoprolaznih modela koje čine današnje stanje tehnike.

3. Jednoprolazni detektori

3.1. You Only Look Once (YOLO)

YOLO je jednoprolazni detektor čija je prva verzija predložena u znanstvenom radu *„You Only Look Once: Unified, Real-Time Object Detection“* koji je inicijalno objavljen 2015. godine na CVPR konferenciji. Rad je pristupio problemu na regresijski način umjesto prenamjenjivanja klasifikatora da radi detekciju objekata, kako je to bilo u prijašnjim radovima. Model YOLO objedinjuje odvojene komponente prijašnjih detektora u jednu neuronsku mrežu te ona predviđa okvire i razredne vjerojatnosti direktno iz slika u jednoj evaluaciji. S obzirom na to da je cijeli cjevovod detekcije samo jedna mreža, on može biti direktno s kraja na kraj optimiziran za performanse detekcije te autori rada ističu njegovu brzinu i mogućnost predviđanja lokalizacije objekata u realnom vremenu kao veliku prednost [3]. U usporedbi s dotadašnjim detektorima, YOLO je imao daleko veću točnost od ostalih modela koji su radili u realnom vremenu, no i dalje je imao manju točnost od sporijih detektora [4]. YOLO detektor je nadograđivan kroz godine te je trenutno aktualna verzija YOLOv5 čija je arhitektura izložena u sljedećim odjeljcima.

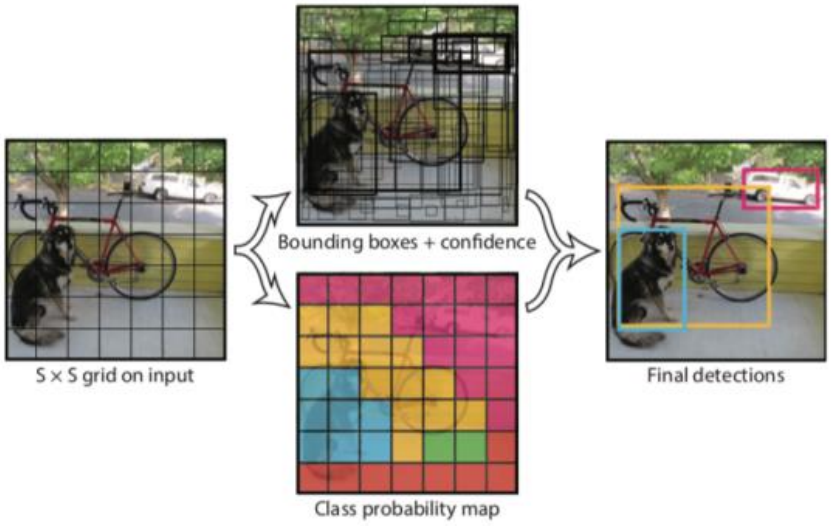
3.1.1. Osnovni koncept modela YOLO

Koncept na kojem se temelji osnovni model YOLO vidi se na slici 3.1 dolje. Na početku se slika podijeli na $S \cdot S$ ćelija jednakih veličina. Ukoliko centar objekta pripada unutar neke ćelije, upravo je ona odgovorna za njegovo detektiranje. Za svaku od ćelija se predviđa B okvira te se računaju pouzdanosti koje govore nalazi li se unutar okvira objekt i koliko je mreža sigurna u svoju predikciju. Pouzdanost odgovara umnošku vjerojatnosti da se u okviru nalazi objekt te omjera presjeka i unije između predviđenog okvira i stvarne lokacije objekta (kako je označeno u skupu podataka). Omjer presjeka i unije pri tome ukazuje na to koliko se stvarni i predviđeni okvir preklapaju.

Svaki okvir sastoji se od pet predikcija: x, y, w, h te pouzdanost. Uređeni par (x, y) predstavlja centar okvira relativiziran u odnosu na granice ćelije. Širina i visina su relativizirani u odnosu na cijelu sliku.

Svaka ćelija također predviđa C vjerojatnosti razreda, uvjetovanih time da se unutar ćelije nalazi objekt. Bez obzira na broj okvira B, predviđa se samo jedan skup razrednih vjerojatnosti po ćeliji.

Naposljetku se odabiru okviri koji imaju najveće vjerojatnosti te se dobiju detektirani objekti.



Slika 3.1: Princip rada modela YOLO. [4]

Arhitektura modela YOLOv5 redom sadrži arhitekturu modela CSPDarknet-53, što je zapravo kralježnica mreže, zatim slijede SPP sloj i PANet kao vrat, te naposljetku dolazi YOLO detekcijska glava. [5] U nastavku je detaljnije opisan svaki od dijelova.

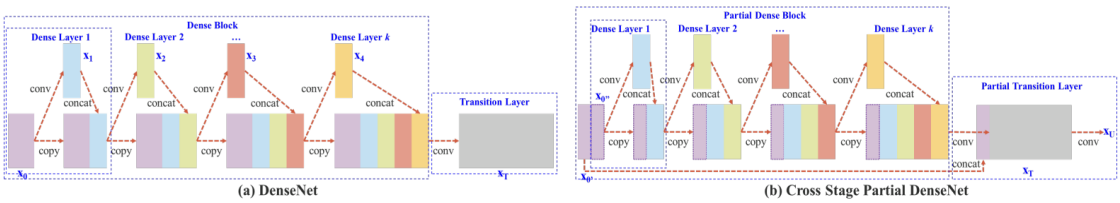
3.1.2. CSPDarknet-53

Mreža CSPDarknet-53 kombinacija je strategije korištene u znanstvenom radu koji predlaže CSPNet [6] i mreže Darknet-53. Mreža Darknet-53 sadrži rezidualne veze i uzastopne $3x3$ i $1x1$ konvolucijske slojeve kojih ukupno ima 53, otkuda dolazi broj koji se krije u nazivu mreže [7]. Njezina struktura prikazana je na slici (Slika 3.2).

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Slika 3.2: Arhitektura modela Darknet-53. [7]

Ideja iza dizajniranja modela CSPNet (punim nazivom engl. *Cross Stage Partial Network*) je omogućiti arhitekturi bogatiju kombinaciju gradijenata, a istovremeno smanjiti broj izračuna. To se postiže na način da se mape značajki baznog sloja podijele na dva dijela te se zatim spajaju kroz predloženu hijerarhiju. Glavni koncept ovakve arhitekture je propagiranje gradijenata kroz različite puteve u mreži njihovim cijepanjem. Na slici 3.3 dolje može se vidjeti na koji način CSPNet funkcionira na primjeru usporedbe mreže DenseNet s mrežom CSPDenseNet u koju je ugrađena CSPNet strategija. CSPDenseNet dijeli mape značajki na dva dijela, od kojih jedan dio prolazi kroz gusto povezani blok i prijelazni sloj, dok se drugi dio kombinira s prenesenim mapama značajki do sljedeće faze.

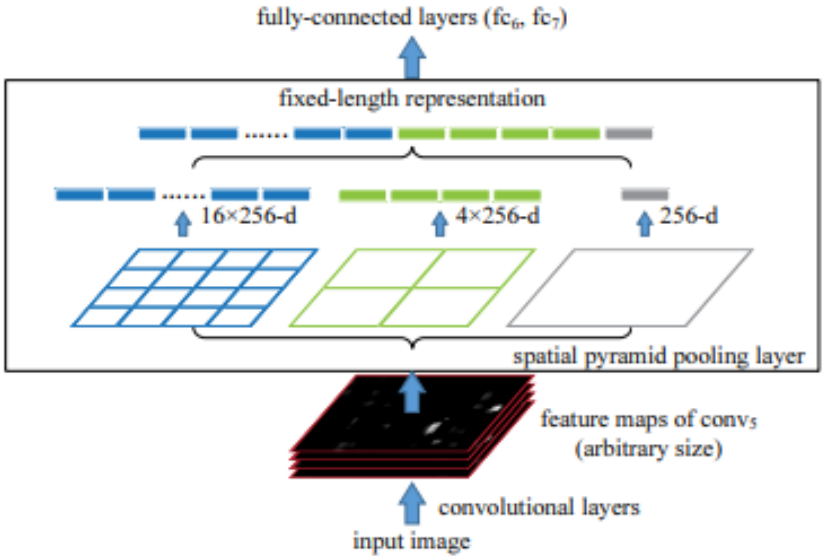


Slika 3.3: Ilustracija (a) mreže DenseNet i (b) predložene mreže CSPDenseNet. [6]

Jedan od doprinosa ovakvog modela je to što uzima u obzir problem višak redundantnih informacija o gradijentima koji rezultira neučinkovitom optimizacijom i skupim izračunima pri zaključivanju. Detektori koji se temelje na CSPNet-u rješavaju problem jačanja moći učenja konvolucijske mreže, odstranjivanja uskih grla pri izračunima te smanjivanja memorijskog troška.

3.1.3. Prostorno piramidalno sažimanje

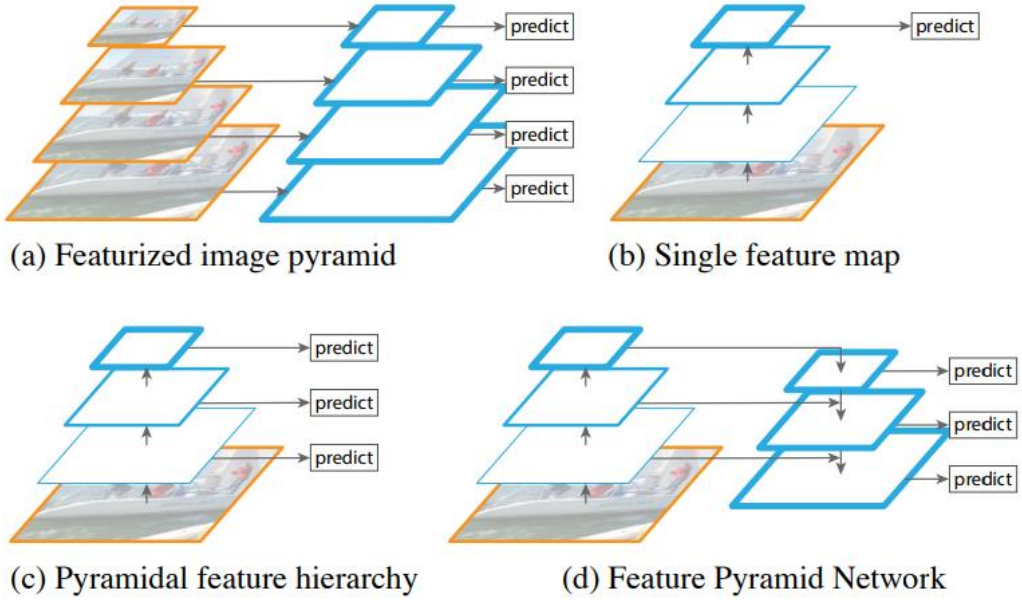
Sloj prostornog piramidalnog sažimanja rješava problem da konvolucijske neuronske mreže generalno zahtijevaju fiksnu veličinu ulazne slike. Izrezivanje i iskrivljenje (engl. *warp*) slike nije prikladno rješenje ovog problema s obzirom na to da izrezivanje ne garantira da će izrezana regija obuhvatiti objekt u cijelosti, a iskrivljenje može rezultirati neželjenim geometrijskim izobličenjima. Konvolucijski slojevi ne zahtijevaju da slika mora biti fiksnih dimenzija te mogu generirati mape značajki bilo kojih veličina, no potpuno povezani slojevi imaju ovo ograničenje po definiciji. Sloj prostornog piramidalnog sažimanja uklanja ovo ograničenje modela sažimanjem mapa značajki i generiranjem izlaza fiksne veličine na način da zadrži prostorne informacije sažimanjem po lokalnim prostornim koševima (engl. *spatial bins*). Spomenuti prostorni koševi veličina su proporcionalnih ulaznoj slici, tako da je njihov broj fiksan. Ovakvim sažimanjem na izlazu se dobivaju kM-dimenzionalni vektori, pri čemu je M broj koševa, a k broj mapa značajki konvolucijskog sloja koji ulazi u sloj prostornog piramidalnog sažimanja. [8] Slika 3.4 ilustrira ovu metodu.



Slika 3.4: Struktura mreže sa slojem prostornog piramidalnog sažimanja. Ovdje je 256 broj jezgri posljednjeg konvolucijskog sloja. [8]

3.1.4. Arhitektura FPN

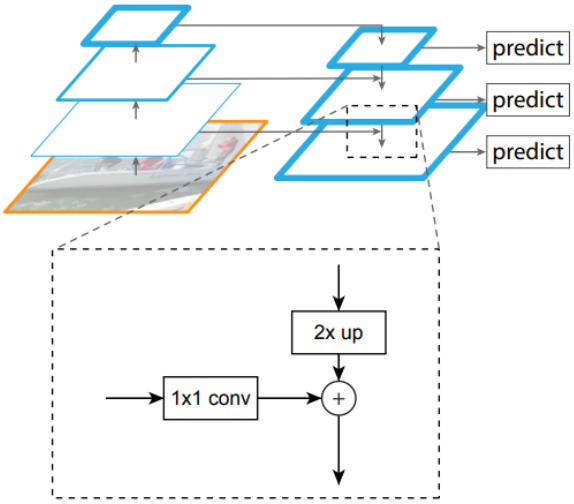
Arhitektura FPN (engl. *Feature Pyramid Network*) [9] mreža je piramidalne arhitekture za ekstrakciju značajki koje se koriste za detekciju objekata. Ovakve arhitekture izračunavaju značajke na svakom mjerilu slike, što je računski i vremenski zahtjevniji pristup od pristupa u kojem se radi detekcija na samo jednoj razini, ali prednost je veći potencijal detekcije objekata različitih veličina. Alternativno tome postoji pristup u kojem se radi nad različitim razinama, ali se pritom koriste značajke dobivene konvolucijskom neuronskom mrežom. Arhitektura FPN kombinacija je ovih strategija – radi na jednom mjerilu slike, ali na više razina, te iskorištava mape značajki iz unaprijednog prolaza konvolucijske mreže. Opisani pristupi ilustrirani su na slici (Slika 3.5).



Slika 3.5: Piramidalne arhitekture za izlučivanje značajki: slika na više mjerila (a), značajke i predikcija na jednoj razini (b), piramida značajki kao ekvivalent piramidi slika (c), FPN koji se temelji na ideji iz b) i c) (d). [9]

Mreža FPN lateralnim vezama aditivno kombinira mape značajki iz *bottom-up* prolaza, koje su veće rezolucije te imaju očuvane prostorne informacije, te mape značajki iz *top-down* prolaza, pri čemu se dobivaju semantički jače reprezentacije. Svaka razina kombinira reprezentaciju na *top-down* putanji s lateralnom vezom iz odgovarajuće razine u *bottom-up* putanji. Prvenstveno se reprezentacija na *top-down* putanji naduzorkuje kako bi bila iste rezolucije kao i reprezentacija iz lateralne veze. Sljedeći korak je da se nad reprezentacijom

iz lateralne veze provede $1x1$ konvolucija kako bi obje reprezentacije imali jednak broj mapa značajki. Na ovaj se način uravnotežuje relativni utjecaj obje podatkovne putanje i omogućuje njihovo jednostavno kombiniranje sumiranjem. Naposljetku se provodi još jedna operacija konvolucije s jezgrom veličine $3x3$ te se time dobiva konačni izlaz. Ovako dobivene mape značajki mogu se dalje koristiti kao ulaz u mrežu za predlaganje regija pri lokalizaciji. Opisani postupak ilustriran je na slici (Slika 3.6).



Slika 3.6: Građevni blok ilustrira lateralnu vezu i *top-down* putanju koje se spajaju zbrajanjem. [9]

3.1.5. PANet

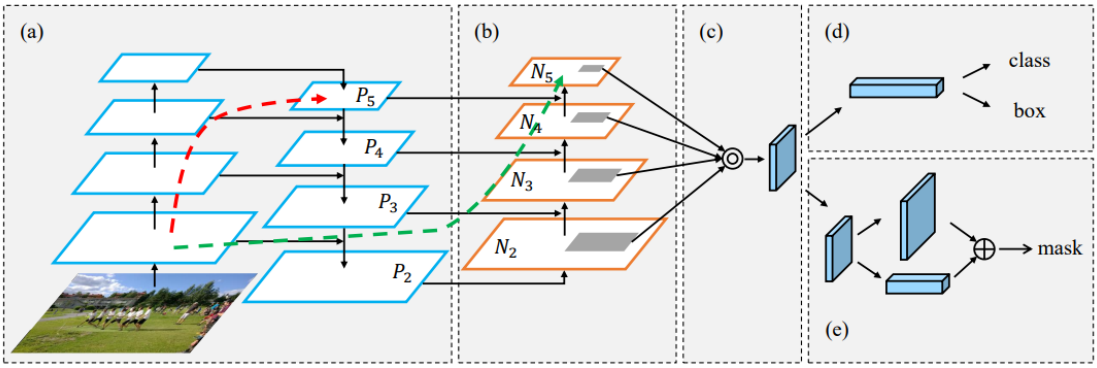
PANet (engl. punog naziva *Path Aggregation Network*) [10] je ukomponiran u YOLOv5 model kako bi poboljšao segmentaciju instanci uz očuvanje prostornih informacija koje pomažu u ispravnoj lokalizaciji piksela za formiranje maske. Arhitektura mreže PANet prikazana je na slici (Slika 3.7).

Pod oznakom (a) vidimo da PANet koristi FPN arhitekturu, te se nadograđuje na nju. Budući da se neuroni dubljih slojeva specijaliziraju za objekte u cijelosti, dok se ostali neuroni najčešće aktiviraju na lokalne uzorke i teksture [11], postoji potreba za augmentacijom *top-down* putanje da bi se bolje propagirale semantički snažne značajke te poboljšale sve značajke s razumno dobrim klasifikacijskim sposobnostima u mreži FPN.

Gradi se još jedna putanja, augmentacija *bottom-up* puta prikazana pod (b), s lateralnim vezama od dna prema vrhu. Stoga, postoji „prečica“ označena zelenom linijom na slici, za razliku od dužeg puta označenog crvenom linijom.

Zatim slijedi adaptivno sažimanje značajki, prikazano na (c), koje omogućuje svakom kandidatu pristup informacijama iz svih razina predikcije. U arhitekturi FPN, kandidati su dodijeljeni različitim razinama značajki sukladno veličinama kandidata. Kandidati manjih veličina dodijeljeni su značajkama nižih razina, dok su kandidati većih veličina dodijeljeni značajkama viših razina. Značajke viših razina su generirane uz velika receptivna polja te obuhvaćaju bogatiji kontekst. Omogućavanje kandidatima manjih veličina da imaju pristup ovim značajkama bolje iskorištava korisne kontekstualne informacije za predikciju. Slično tome, značajke nižih razina sadrže finije detalje i visoku lokalizacijsku točnost te činjenica da im kandidati većih veličina mogu pristupiti donosi veliku prednost. Adaptivno sažimanje značajki radi na način da se svaki kandidat mapira na različite razine značajki, kao što je ilustrirano sivim pravokutnicima pod (b). Zatim se koristi poravnavanje regija interesa kako bi se sažele rešetke značajki iz različitih razina. Nakon toga se primjenjuje operacija sume ili maksimiziranja po elementima.

Naposljetku, rešetka značajki se koristi za svakog od kandidata za daljnju predikciju, primjerice klasifikaciju, regresiju okvira i predikciju maske.



Slika 3.7: Ilustracija arhitekture. (a) FPN kralježnica. (b) *Bottom-up* augmentacija putanje. (c) Adaptivno sažimanje značajki. (d) Klasifikacija i regresija okvira. [10]

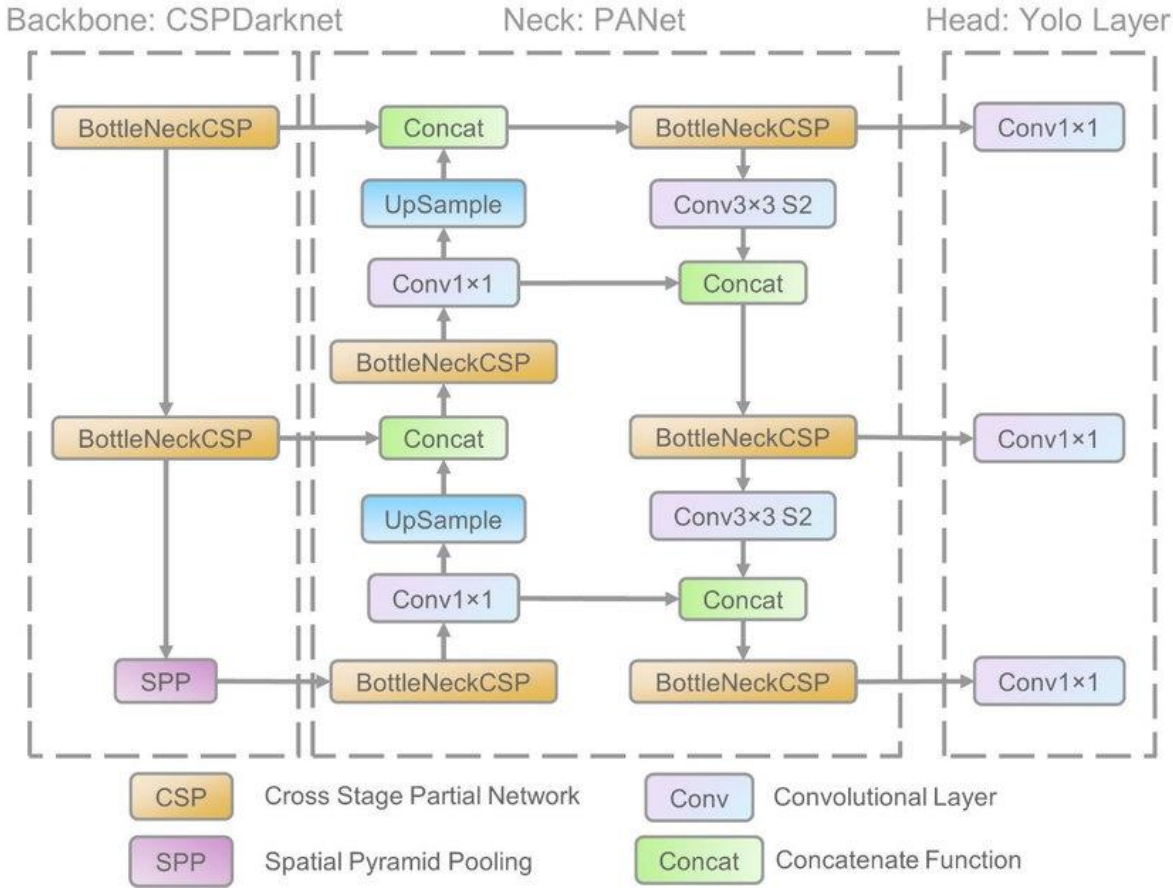
3.1.6. YOLO detekcijska glava

YOLO predviđa okvire na tri različita mjerila. Na osnovni ekstraktor značajki dodaju se konvolucijski slojevi. Posljednji od njih predviđa trodimenzionalni tenzor koji enkodira okvir, vjerojatnost da se unutar njega nalazi objekt te predikciju razreda. U radu [7] je navedeno da se predviđa po tri okvira na svakom mjerilu te je stoga tenzor dimenzija $N \cdot N \cdot$

$[3 \cdot (4 + 1 + K)]$, pri čemu broj četiri označava parametre okvira, broj jedan vjerojatnost objekta unutar okvira, dok K predstavlja broj razreda.

3.1.7. Konačna arhitektura modela YOLOv5

U prethodnim odjeljcima pojedinačno su objašnjene osnovne gradivne strukture mreže YOLOv5 te motivacija iza njihovog nastanka i korištenja, kao i njihove funkcionalnosti i svrha. Povezivanjem svih prethodno opisanih jedinica u jednu cjelinu nastaje neuronska mreža koju nazivamo YOLOv5, te je njena konačna arhitektura prikazana na slici (Slika 3.8).



Slika 3.8: Potpuna arhitektura modela YOLOv5. (preuzeto dana 10.06.2022. s: https://github.com/ultralytics/yolov5/issues/6998)

3.1.8. Funkcija gubitka u modelu YOLOv5

Funkcija gubitka u modelu YOLOv5 sastoji se od tri dijela: gubitka koji se odnosi na klase, objekte te lokaciju. Gubitci za klase i objekte računaju se funkcijom gubitka binarne

unakrsne entropije. Gubitak vezan za objekt sastoji se od gubitaka za tri predikcijska sloja (P3, P4 i P5 – čiji se položaji unutar mreže mogu vidjeti na slici 3.8 gore). Svaki od njih se množi određenim težinskim faktorom prije sumiranja, kao što se vidi iz sljedeće jednadžbe.

$$L_{obj} = 4.0 \cdot L_{obj}^{small} + 1.0 \cdot L_{obj}^{medium} + 0.4 \cdot L_{obj}^{large} \tag{3.1}$$

Gubitak vezan za lokaciju, odnosno regresijski gubitak okvira izračunava se funkcijom gubitka $cIOU$ (engl. *Complete Intersection over Union*), prikazanom formulom (3.2), koji je agregacija površine preklapanja, udaljenosti i omjera stranica. S u formuli označava površinu preklapanja kao $1 - IoU$, D je normalizirana IoU udaljenost između centra predikcije i centra oznake, a V je dosljednost omjera stranica. $cIoU$ gubitak koristi geometrijske mjere za regresiju okvira, što pridonosi bržoj konvergenciji i boljim performansama u odnosu na gubitak omjera presjeka i unije.

$$\mathcal{L} = S(\mathcal{B}, \mathcal{B}^{gt}) + D(\mathcal{B}, \mathcal{B}^{gt}) + V(\mathcal{B}, \mathcal{B}^{gt}) \tag{3.2}$$

U konačnici, ova tri gubitka se zbrajaju te se dobiva ukupni gubitak.

4. Dvoprolazni detektori

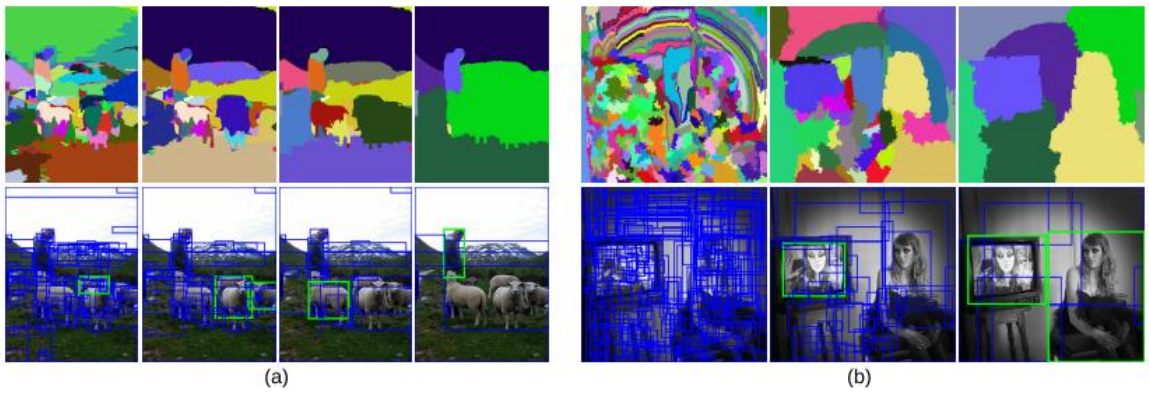
U ovom poglavlju bit će opisane arhitekture na temelju kojih je u konačnici sastavljen model Faster R-CNN kao nadogradnja na te iste arhitekture. Faster R-CNN ima bolje performanse od osnovnih modela na kojima se temelji te je i dalje konkurentan u odnosu na njih.

4.1. Arhitektura R-CNN

Arhitektura R-CNN (punog naziva engl. *Region Based Convolutional Network*) [12] kombinira konvolucijsku neuronsku mrežu s idejom predlaganja regija.

Njena struktura složena je od tri različita modula. Prvi od njih je modul zadužen za predlaganje regija te on radi tako da uzima sliku s ulaza i stvara 2000 kandidata regija. U ovu svrhu koristi se algoritam selektivnog pretraživanja (engl. *selective search algorithm*) koji je puno efikasniji od primjerice vrlo računski zahtjevnog iscrpnog pretraživanja.

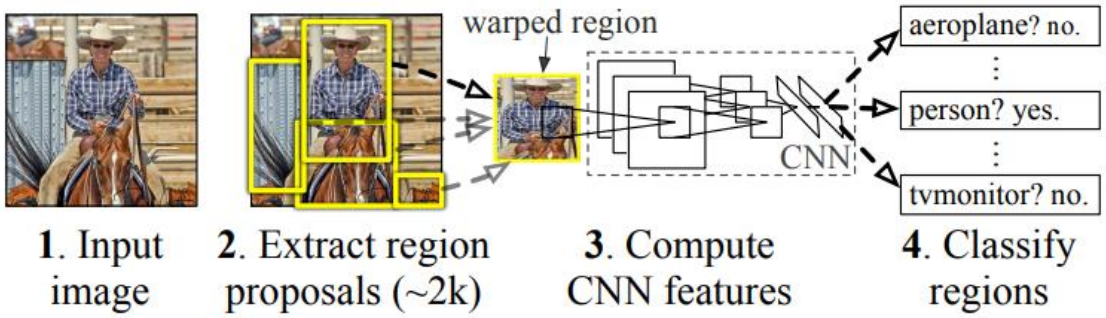
Algoritam radi na sljedeći način. Prvo se generiraju inicijalne podsegmentacije ulazne slike. One podsegmentacije koje su slične se zatim rekurzivno kombiniraju u sve veće podsegmentacije u velikom broju iteracija. Sličnost se definira po boji, teksturi, veličini (pri čemu manje regije imaju veću vjerojatnost za spajanjem) te ispunjenju. Nakon što su dobivene finalne segmentacije, odnosno predložene regije, stvaraju se okviri koji najbolje odgovaraju rubovima segmentacija. [13] Opisani algoritam ilustriran je na slici 4.1 dolje.



Slika 4.1: Dva primjera algoritma selektivnog pretraživanja koji pokazuju nužnost različitih mjerila. [13]

Nakon modula za predlaganje regija slijedi modul zadužen za izlučivanje značajki, čija je funkcionalnost izlučivanje značajki iz navedenih regija uz pomoć konvolucijske neuronske mreže. Iz svake se regije kandidata izlučuje vektor značajki dimenzionalnosti 4096, a značajke se računaju prolazom RGB slike kroz mrežu koja se sastoji od pet konvolucijskih i dva potpuno povezana sloja.

Naposljetku dolazi modul zadužen za klasifikaciju koji pridjeljuje određeni razred svakoj od predloženih regija. Modul za svaki razred koristi SVM (engl. *support vector machine*) naučen za tu klasu te algoritmom potiskivanja ne-maksimalnih odziva, uzevši u obzir klasifikacijsku mjeru okvira te mjeru preklapanja s ostalim okvirima (omjer presjeka i unije), bira konačni skup okvira s najvećom vjerojatnosti da se preklapaju s objektima. Skica ovog postupka prikazana je na slici (Slika 4.2).



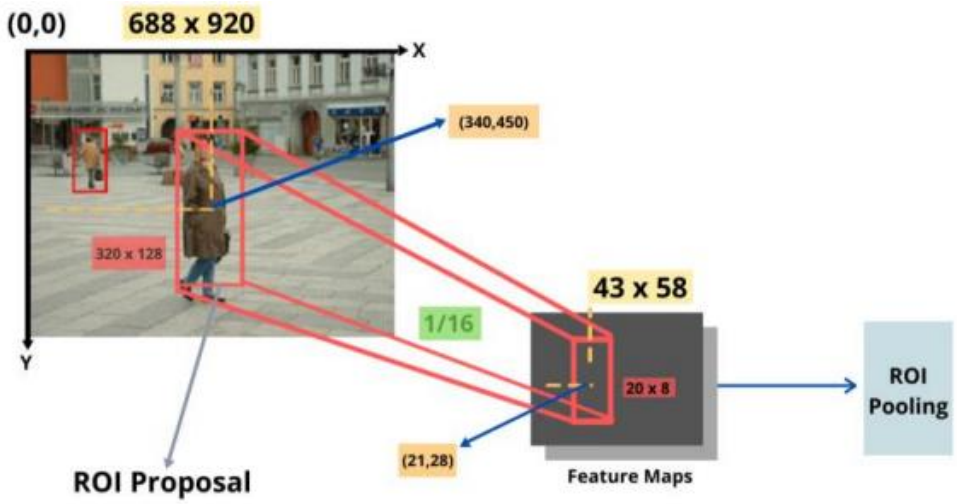
Slika 4.2: Pregled R-CNN sustava za detekciju objekata s označenim slijednim koracima rada. [12]

Problem s arhitekturom R-CNN je da je učenje modela vrlo sporo s obzirom na to da je potrebno klasificirati 2000 predloženih regija po jednoj slici. Uz to, vrijeme zaključivanja modela nije dovoljno brzo da bi model mogao raditi u realnom vremenu. Nadalje, algoritam selektivnog pretraživanja je fiksan algoritam te se model ne uči za vrijeme rada algoritma, što može dovesti do generiranja lošijih kandidata. [14] Također, jedan od problema je i taj da se iste aktivacije računaju više puta u slučajevima kad se kandidati preklapaju. Rješenje nekih od ovih problema nudi arhitektura Fast R-CNN.

4.2. Arhitektura Fast R-CNN

Arhitektura Fast R-CNN [15] nadogradnja je na arhitekturu R-CNN opisanu u prethodnom podpoglavlju. Poboljšanja podrazumijevaju brže učenje modela i kraće vrijeme zaključivanja modela te točnost detekcije. Također, učenje Fast R-CNN modela je jednorazinsko za razliku od višerazinskog cjevovoda učenja kod R-CNN modela, jer se paralelno uče klasifikator i regresor.

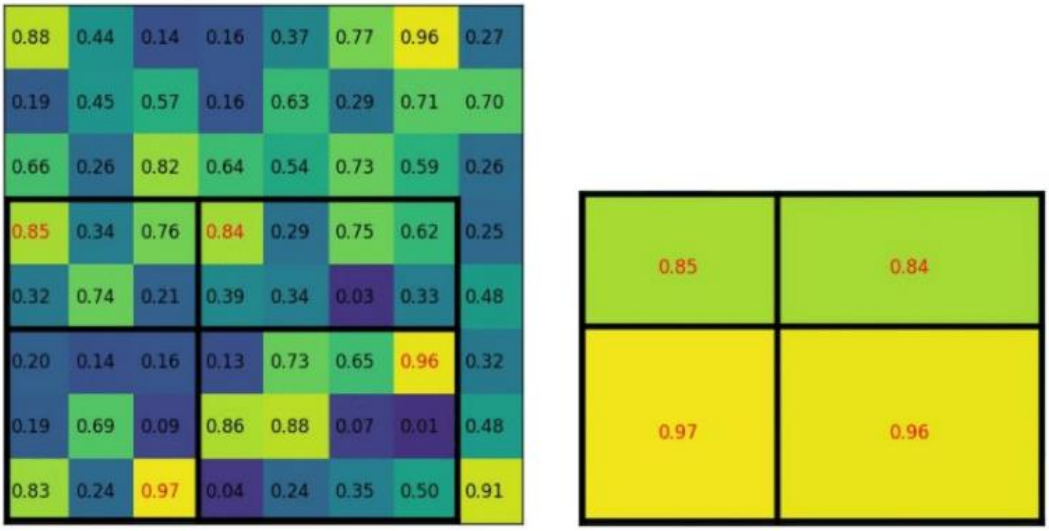
Fast R-CNN na ulazu dobiva sliku i skup predloženih regija. Ulazna slika prolazi kroz kovolucijske slojeve i slojeve sažimanja kako bi se iz nje dobile mape značajki. Algoritmom selektivnog pretraživanja izlučuju se regije kandidati. Za svaku sliku je potrebno napraviti samo jedan unaprijedni prolaz, a ne onoliko unaprijednih prolaza koliko ima predloženih regija, što je slučaj u modelu R-CNN. Ova pogodnost je moguća jer se koristi postupak projekcije regija interesa (engl. *RoI projection*). Ono što postupak radi jest preslikavanje dimenzija regija interesa ulazne slike u dimenzije mape značajki uz odgovarajući faktor poduzorkovanja, što je prikazano na slici (Slika 4.3).



Slika 4.3: Ilustracija projekcije regije kandidata s ulazne slike na mapu značajki uz faktor poduzorkovanja 1/16. [16]

Predložene regije interesa, mapirane na dimenzije mape značajki, dalje ulaze u sloj sažimanja po regijama (engl. *RoI pooling*). Sloj sažimanja po regijama koristi sažimanje maksimalnom vrijednošću u svrhu pretvaranja značajki unutar bilo koje regije interesa u malu mapu značajki fiksnih prostornih dimenzija $H \cdot W$. Pri tome su H i W hiperparametri

sloja te su neovisni o pojedinim regijama interesa. Regija interesa je definirana uređenom četvorkom (r, c, h, w) koja specificira gornji lijevi kut te visinu i širinu. Sažimanje regija interesa maksimalnom vrijednošću funkcionira na način da se prozor regije interesa dimenzija $h \cdot w$ podijeli na rešetku koja se sastoji od $H \cdot W$ ćelija, od kojih je svaka ćelija otprilike veličine $\frac{h}{H} \cdot \frac{w}{W}$. Zatim se provodi standardna operacija sažimanja maksimalnom vrijednošću unutar svake od ćelija pri čemu maksimumi iz svake ćelije idu u izlazno polje. Postupak sažimanja regija interesa prikazan je na slici 4.4 dolje. [2]



Slika 4.4: Vizualni prikaz primjer RoI sažimanja. (preuzeto dana 20.06.2022. s: https://deepsense.ai/wp-content/uploads/2017/02/roi_pooling-1.gif)

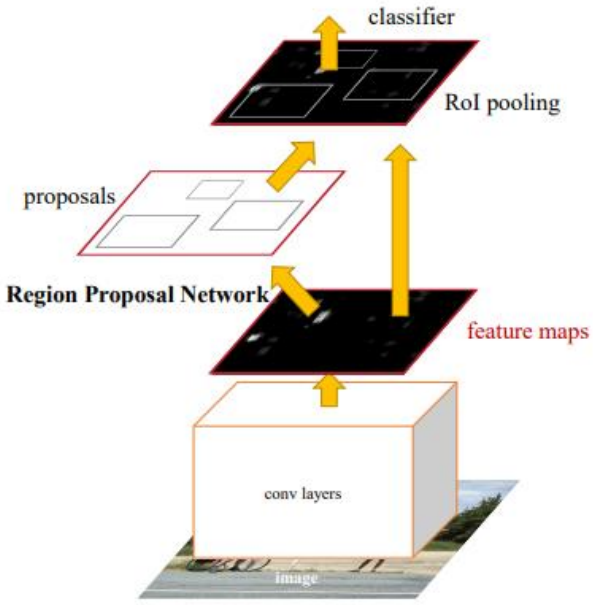
Kao što je već spomenuto, Fast R-CNN učenje je jednorazinsko s obzirom na paralelno učenje klasifikatora i regresora. Klasifikacijom u jedan od razreda predviđa se vjerojatnosna distribucija po regijama interesa pri čemu ulaz u funkciju softmax čine izlazi iz potpuno povezanog sloja. Gubitak klasifikacije je logistički gubitak. Regresijska grana podrazumijeva regresiju okvira za svaki od razreda pri čemu se definira pomak koji je neovisan o mjerilu te logaritam omjera visine i širine predloženog okvira u odnosu na oznaku. Gubitak regresije je glatki-L1 gubitak.

Fast R-CNN ima zajednički gubitak sastavljen od dvije prethodno navedene komponente. Dva gubitka se sumiraju u zajednički gubitak, pri čemu je uz regresijski gubitak dodan multiplikator koji regulira ravnotežu između dviju komponenti gubitka. Iz ove činjenice

proizlazi zaključak da se istovremeno uče klase i dimenzije okvira, što omogućuje jednorazinsko učenje.

4.3. Arhitektura Faster R-CNN

Faster R-CNN arhitektura [17] nadogradnja je na arhitekturu Fast R-CNN. Problem kod Fast R-CNN mreže je činjenica da je izračun regija kandidata usko grlo. Faster R-CNN ukida ovu nepogodnost uvođenjem modula za predlaganje regija čiji predloženi kandidati regija zatim ulaze u Fast R-CNN detektor. RPN modul, koristeći mehanizme pozornosti, govori Fast R-CNN arhitekturi kamo treba gledati pri traženju objekata. Faster R-CNN arhitektura skicirana je na slici (Slika 4.5). Faster R-CNN arhitektura kao kralježnicu ima konvolucijsku neuronsku mrežu. U eksperimentalnom dijelu ovog rada koristi se Faster R-CNN s kralježnicom ResNet-50 i FPN modulom koji je već opisan u poglavlju 2.1.4.

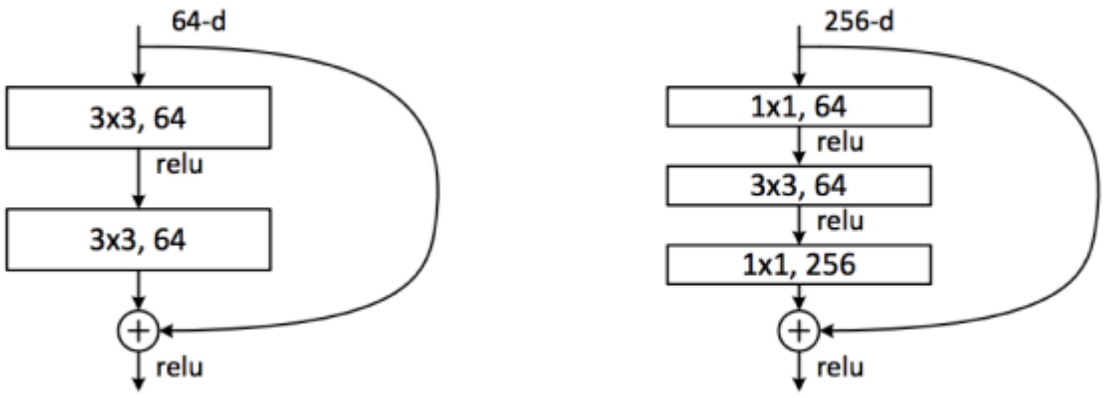


Slika 4.5: Faster R-CNN je jedna, ujedinjena mreža za detekciju objekata. RPN modul služi kao mehanizam pozornosti ove ujedinjene mreže. [17]

4.3.1. Arhitektura mreže ResNet-50

Mreža ResNet-50 je rezidualna neuronska mreža. Rezidualne neuronske mreže općenito su vrlo pogodne za probleme iz računalnog vida zato što rješavaju problem zasićenja točnosti

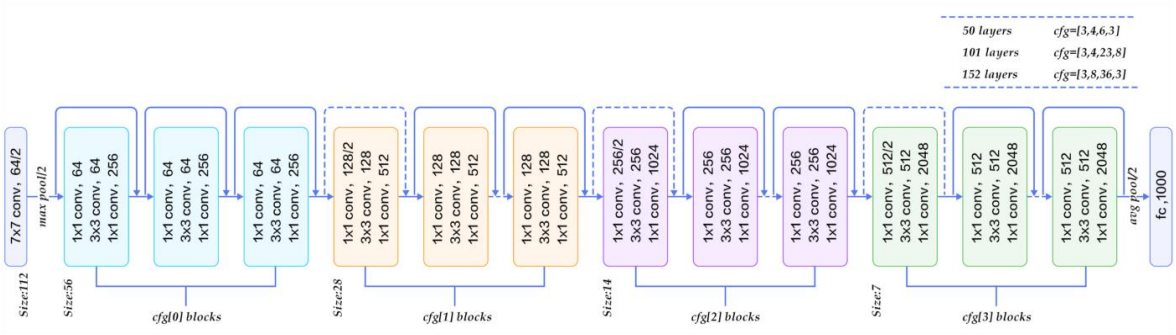
u dubokim modelima. Njih karakterizira uvođenje rezidualnih jedinica [18] koje koriste prečice (engl. *skips*) u svrhu pospješivanja stabilnosti gradijenata ranijih slojeva dubokog modela. Ulaz rezidualne jedinice račva se na dvije grane, od kojih prva prolazi kroz dva konvolucijska sloja, a druga grana ostaje jednaka kao na samom ulazu u jedinicu, odnosno nad njom se primjenjuje samo funkcija identiteta. Nakon toga se grane ponovno spoje na način da se izlazi iz obje grane sumiraju. U slučaju da zbrajanje nije moguće zbog nepodudaranja dimenzija mapa značajki, provodi se dodatna konvolucija s veličinom jezgre $1x1$ i po potrebi korakom dva kako bi se povećao broj mapa značajki. Druga grana, koja je u opisanom primjeru ostala identična kao na ulazu, naziva se prečicom. Ovakav opis odgovara osnovnoj rezidualnoj jedinici (engl. *baseline unit*). Ona se koristi kod umjereno dubokih modela koji imaju do 30 slojeva, kao što je ResNet-18 sa 18 slojeva. Međutim, kod izrazito dubokih modela javlja se mogućnost prenaučenosti modela zbog prevelikog broja parametara. Iz tog razloga uvedena je i rezidualna jedinica s uskim grlom koja smanjuje broj parametara. Ova jedinica sastoji se od tri konvolucije, od kojih je prva konvolucija zadužena za smanjenje broja mapa značajki, a posljednja za povećanje na isti broj kao na ulazu u blok. [19] Obje vrste rezidualnih jedinica prikazane su na slici 4.6 dolje.



Slika 4.6: Osnovna rezidualna jedinica (lijevo) i jedinica s uskim grlom (desno). [19]

Mreža ResNet-50 ima sljedeću arhitekturu po slojevima. Prvi sloj mreže nad ulaznom trokanalnom slikom provodi operaciju konvolucije s jezgrom veličine $7x7$ i korakom dva, pri čemu nastane mapa značajki dubine 64. Zatim se redom primjenjuju normalizacija, prijenosna funkcija zglobnice te sažimanje maksimalnom vrijednošću s jezgrom veličine $3x3$ i korakom dva. Idući koraci odnose se na prolazak kroz rezidualne blokove. Mreža

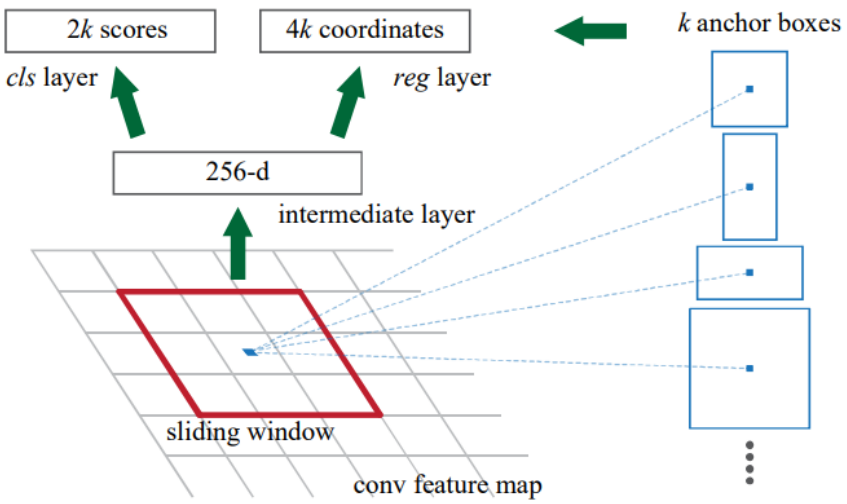
ResNet-50 ima četiri faze (engl. *stages*) koje se redom sastoje od tri, četiri, šest i ponovno tri rezidualne jedinice s uskim grlom. Naposljetku dolazi potpuno povezani sloj dimenzionalnosti 1000. Potpuna arhitektura mreže ResNet-50 prikazana je na slici (Slika 4.7).



Slika 4.7: Potpuna arhitektura mreže ResNet-50. (preuzeto dana 20.06.2022. s: https://www.researchgate.net/publication/336805103/figure/fig4/AS:817882309079050@1572009746601/ResNet-50-neural-network-architecture-56.ppm)

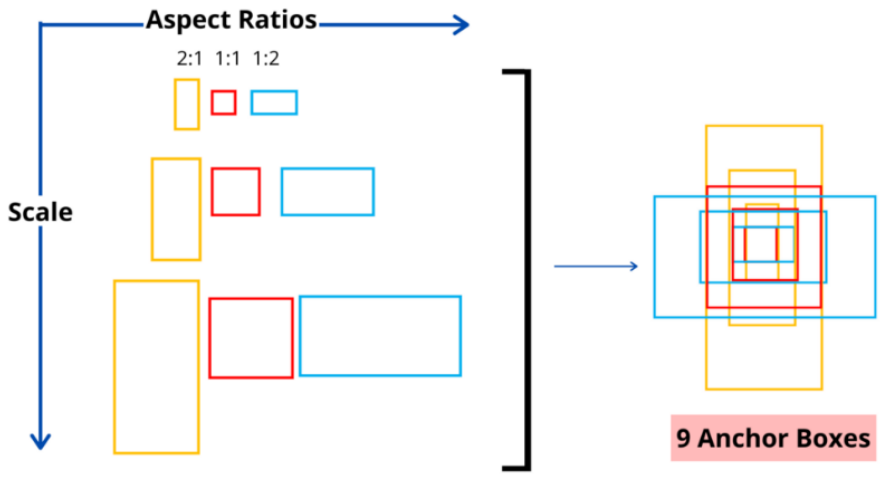
4.3.2. Arhitektura RPN

Arhitektura za predlaganje regija (punim nazivom engl. *Region Proposal Network*) [17] na ulazu prima sliku bilo koje veličine te na izlazu daje skup pravokutnih prijedloga regija pri čemu je svakome od njih pridružena mjera da se unutar danog okvira nalazi objekt (engl. *objectness score*). Ovaj proces funkcionira na način da ulazna slika prolazi kroz rezidualnu konvolucijsku neuronsku mrežu ResNet pri čemu se stvaraju mape značajki koje zatim FPN kombinira kako bi se dobile semantički bogate mape visoke rezolucije. Nakon toga, koristi se klizeći prozor koji klizi po mapi značajki koja je izlaz iz posljednjeg konvolucijskog sloja i radi preslikavanje u značajku manjih dimenzija. Ova značajka zatim ulazi u dva potpuno povezana sloja, odnosno u sloj za regresiju te sloj za klasifikaciju okvira. S obzirom na to da se koristi princip klizećeg prozora, potpuno povezani slojevi su dijeljeni među svim prostornim lokacijama. Ovaj koncept ilustriran je na slici (Slika 4.8).



Slika 4.8: Mreža za predlaganje regija RPN. Na svakoj poziciji postavi se k okvira određene površine i omjera stranica. Izlaz se svede na dimenzionalnost 256 te se paralelno provode regresija i klasifikacija. [17]

Kada govorimo o arhitekturi RPN, važno je objasniti koncept okvira (engl. *anchors*) koji predstavljaju potencijalne položaje objekata koji su poravnati s odgovarajućim slikovnim elementom mape značajki te imaju svojstvo invarijantnosti na translaciju i razinu piramide. Na svakoj lokaciji klizećeg prozora istovremeno se predviđa više regija kandidata, pri čemu je maksimalni broj mogućih prijedloga za svaku lokaciju označen sa k. Stoga regresijski sloj ima $4k$ izlaza koji enkodiraju koordinate k okvira, a klasifikacijski sloj na izlazu vraća $2k$ procjena vjerojatnosti nalazi li se objekt u okviru ili ne za svakog kandidata. Svaki od k predloženih kandidata parametriziran je u odnosu na k referentnih okvira. Referentni okvir je centriran s obzirom na trenutni klizeći prozor te je povezan uz tri različita mjerila i tri različita omjera stranica. Dakle, za svaku lokaciju klizećeg prozora, postoji $k = 9$ referentnih okvira kao što je prikazano na slici 4.8 gore i slici 4.9 dolje.



Slika 4.9: Vizualni prikaz koncepta referentnih okvira. (preuzeto dana 10.06.2022. s: https://miro.medium.com/max/1050/1*m2AWrKCsiOLgi6dF82dZug.png)

Za konvolucijsku mapu značajki veličine $W \cdot H$, postoji ukupno $W \cdot H \cdot k$ referentnih okvira.

Važno svojstvo okvira je njihova invarijantnost na translaciju, što znači da ukoliko se objekt pomakne na slici, treba se moći jednako dobro predvidjeti prisutnost objekta bez obzira na njegovu lokaciju. Također, okviri sami po sebi podržavaju višerazinske predikcije, odnosno RPN koristi slike i mape značajki samo na jednoj razini, ali referencira okvire različitih veličina i omjera stranica.

4.3.3. Mjera preklapanja okvira - IoU kod RPN modula

Mjera omjera presjeka i unije (engl. *Intersection-over-Union*) je omjer površine presjeka i unije površina predloženog okvira i oznake. To je mjera preklapanja dvaju okvira za koju ciljamo da bude što veća, maksimalno jednaka jedan. Razlikujemo tri tipa okvira sukladno vrijednosti koje omjer presjeka i unije može poprimiti: pozitivni, negativni i neutralni. Pozitivni okviri su oni koji imaju IoU mjeru veću od određenog predefiniranog praga za pozitivne okvire. Negativni okviri imaju IoU mjeru manju od praga za negativne okvire. Neutralni okviri imaju mjeru IoU veću od gornjeg praga za negativne okvire te donjeg praga za pozitivne okvire. Ovakva mjera u svrhu definiranja ispravnosti predloženih okvira u odnosu na njihove oznake ključna je za regresiju i klasifikaciju.

S obzirom na to da je mogući slučaj da se s nekom oznakom nijedan okvir ne preklapa u mjeri koja je veća od praga, svakoj oznaci svejedno treba pridijeliti onaj okvir koji se s njome preklapa u najvećoj mjeri te njega proglasiti pozitivnim okvirom.

Nakon što je definirano na koji način se okviri određuju u koju od kategorija, može se složiti zajednički gubitak kojeg čine klasifikacijski i regresijski gubitak u zbroju. Ove komponente analogne su onima kod Fast R-CNN arhitekture. Klasifikacijski gubitak računa se nad pozitivnim i negativnim okvirima i pri njegovom izračunu ne uzimaju se u obzir neutralni okviri. Regresijski gubitak se računa isključivo za pozitivne okvire. Također, klasifikacijski gubitak se normalizira veličinom mini-grupe, dok se regresijski gubitak normalizira brojem lokacija referentnih okvira.

Fast R-CNN i Faster R-CNN provode regresiju okvira na različit način. Fast R-CNN radi regresiju nad značajkama koje dolaze iz regija interesa koje su različitih veličina te se pri tome regresijski parametri dijele među svim veličinama. Faster R-CNN pri regresiji okvira uzima značajke istih prostornih dimenzija te se parametri regresije ne dijele među k regija. Uči se k regresora okvira pri čemu je svaki od njih odgovoran za jedno od mjerila i jedan od omjera stranica okvira te Faster R-CNN upravo zbog ovoga može uspješno predviđati okvire različitih veličina.

Treba napomenuti da arhitektura RPN uči na mini-grupama koje čine nasumično odabrani okviri sa slike. Ne uči se nad svim okvirima, iako je to moguće, iz razloga što bi u tom slučaju bilo znatno više negativnih od pozitivnih okvira te bi došlo do pristranosti prema negativnim okvirima, a idealan slučaj bi bio kad bi bio jednak broj pozitivnih i negativnih okvira.

4.3.4. Učenje modela Faster R-CNN

S obzirom na to da, kako je već spomenuto ranije, želimo postići da arhitekture RPN i Fast R-CNN dijele konvolucijske slojeve umjesto da se dvije mreže odvojeno uče. Predložena su tri pristupa učenja u kojima se dijele značajke: naizmjenično učenje, približno zajedničko učenje i nepribližno zajedničko učenje.

Naizmjenično učenje je pristup u kojem se prvo uči arhitektura RPN, te se dobivene predložene regije koriste za učenje modela Fast R-CNN. Zatim se tako naučeni model koristi za inicijalizaciju RPN modula te se ovaj proces iterativno ponavlja.

Približno zajedničko učenje spaja mreže RPN i Fast R-CNN u jednu mrežu za vrijeme učenja. U svakoj iteraciji stohastičkog gradijentnog spusta, unaprijedni prolaz generira prijedloge regija koje se tretiraju kao fiksirani predizračunati kandidati pri treniranju Fast R-CNN detektora. Pri unatražnom prolazu za dijeljene slojeve vrijedi da se kombiniraju propagirani signali i od RPN i Fast R-CNN gubitaka. Ovakav pristup naziva se približnim zajedničkim učenjem s obzirom na činjenicu da se ne računaju gradijenti gubitka s obzirom na koordinate kandidata koji su inače također izlaz iz modela Fast R-CNN, no u ovom slučaju nisu jer postoji samo izlaz detektora kao konačni izlaz.

Nepribližno zajedničko učenje, za razliku od približnog, u izračun uključuje gradijente po koordinatama okvira. U Fast R-CNN arhitekturi, sloj sažimanja regija interesa na ulazu prima konvolucijske značajke i predviđene okvire. Sukladno tome, u ovoj strategiji trebao bi se uvesti sloj sažimanja regija interesa koji bi bio diferencijabilan s obzirom na koordinate okvira.

Za učenje Faster R-CNN modela odabran je pristup naizmjeničnog učenja u četiri koraka. U prvom koraku se uči RPN modul stohastičkim gradijentnim spustom sa zaletom te se određuje kategorija svakog okvira s obzirom na mjeru preklapanja okvira kako je opisano u prethodnom odjeljku. Mreža je inicijalizirana težinama naučenima na skupu podataka ImageNet. U drugom koraku, uči se mreža Fast R-CNN uz korištenje kandidata regija koje je generirao RPN modul u prvom koraku učenja. U ovom koraku dvije mreže još uvijek ne dijele konvolucijske slojeve. U trećem koraku, koristi se mreža detektor, odnosno Fast R-CNN u svrhu inicijalizacije učenja RPN modula, s time da se fiksiraju dijeljeni konvolucijski slojevi te se uče isključivo slojevi jedinstveni za RPN modul. Sada dvije mreže dijele konvolucijske slojeve. Konačno, u četvrtom koraku se dijeljeni konvolucijski slojevi zadržavaju fiksiranima te se uče samo slojevi Fast R-CNN mreže. Ovakav pristup naizmjeničnog učenja može se primijeniti na više iteracija, no time nije uočen značajan napredak.

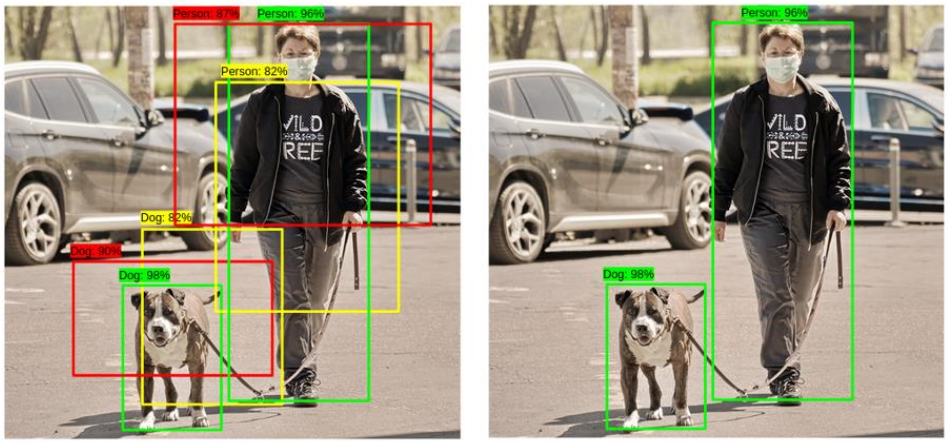
4.4. Algoritam NMS

Algoritam NMS (engl. *Non-maximum suppression*) koriste arhitekture R-CNN, Fast R-CNN i Faster R-CNN. On rješava problem višestrukih detekcija istog objekta, odnosno nepogodnost da se za neki objekt u slici dobije mnogo kandidata. U idealnom slučaju, za svaki objekt i njegovu oznaku, postoji samo jedan okvir koji se s njome dovoljno dobro preklapa prema mjeri preklapanja okvira. Način rada algoritma opisan je u nastavku.

Algoritam na ulazu prima listu svih predviđenih okvira u obliku uređenih petorki $(x1, y1, x2, y2, c)$ pri čemu su uređeni parovi $(x1, y1)$ i $(x2, y2)$ koordinate gornjeg lijevog i donjeg desnog ruba okvira, a c predstavlja klasifikacijsku mjeru – vjerojatnost da se unutar okvira nalazi objekt. Uz to, algoritmu je potrebno dati brojčani prag mjere preklapanja okvira. Predana lista okvira sortira se silazno prema klasifikacijskoj mjeri. Na početku algoritma, uklanja se svaki okvir čija je klasifikacijska mjera manja od zadanog praga.

Zatim, s obzirom na to da je lista sortirana, sigurno je da prvi element u njoj čini onaj okvir s najvećom klasifikacijskom mjerom. Sljedeće, uklanjamo taj okvir iz liste i dodajemo ga u novu, na početku praznu listu. U ovom trenutku se definira novi, dodatni prag za mjeru preklapanja okvira. Ovaj prag služi za uklanjanje okvira koji pripadaju istoj klasi, a međusobno imaju visoku mjeru preklapanja (ne u odnosu na oznaku, nego jedni u odnosu na druge). Objašnjenje iza ovoga krije se iza činjenice da ukoliko se dva okvira iste klase preklapaju u velikoj mjeri, vrlo je vjerojatno da oba okvira pokrivaju isti objekt. S obzirom na to da je cilj imati samo jedan okvir po objektu, pokušava se ukloniti onaj od okvira koji ima manju klasifikacijsku mjeru. Dakle, ukoliko u ulaznoj listi okvira imamo okvir klasifikacijske mjere 0.9 koji pripada jednoj od klasa te još jedan okvir klasifikacijske mjere 0.85 koji pripada istoj klasi, zadržat ćemo prvi okvir, a drugi ukloniti.

Ovaj proces se ponavlja za svaki okvir, te se u konačnici na izlazu iz algoritma dobiva lista jedinstvenih okvira sa zadovoljavajućim klasifikacijskim mjerama. Rezultat NMS algoritma prikazan je u nastavku na slici (Slika 4.10).

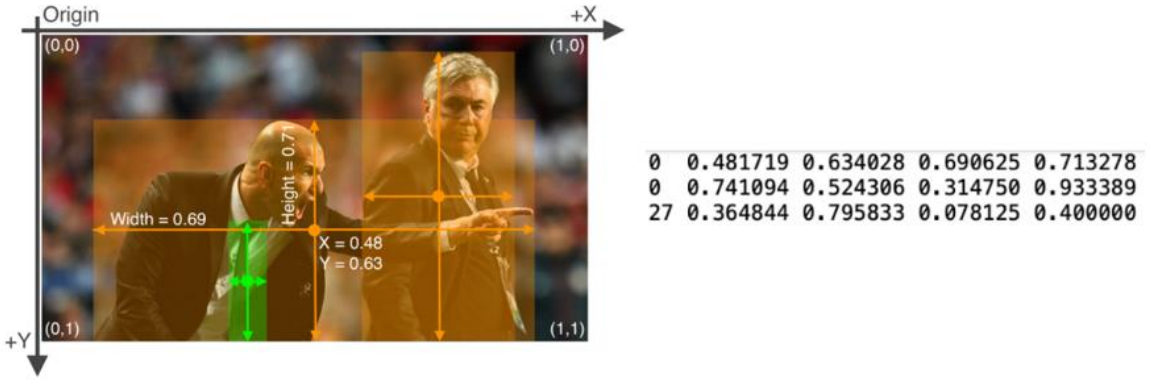


Slika 4.10: Prikaz ulaza u NMS algoritam (lijevo) i izlaza iz NMS algoritma (desno). (preuzeto dana 18.06.2022. s: https://www.analyticsvidhya.com/blog/2020/08/selecting-the-right-bounding-box-using-non-max-suppression-with-implementation/)

5. Programska implementacija

5.1. Priprema skupa podataka

Cilj ovog rada je razviti model koji će detektirati i razlikovati djecu od odraslih osoba. U svrhu toga, izrađen je vlastiti skup slika za učenje i validaciju. Vlastiti skup podataka sastoji se od slika koje su preuzete iz slobodno dostupnih i besplatnih izvora. Nakon preuzimanja slika, bilo je potrebno ručno izraditi anotacije za njih. Za izradu anotacija korišten je alat „makesense.ai“ (https://www.makesense.ai/) koji ne zahtijeva nikakvu dodatnu instalaciju, već ima mogućnost korištenja putem internetskog preglednika. Alat podržava različite tipove oznaka među kojima su pravokutnici, linije, točke i poligoni. S obzirom na domenu problema, odgovarajući tip oznaka su pravokutnici. Što se tiče izlaznog formata anotiranih podataka, alat podržava YOLO, VOC XML, VGG JSON i CSV format. Budući da je YOLOv5 jedan od modela korištenih u eksperimentima ovoga rada, anotacije su izvezene u YOLO formatu. Svaka slika jedinstvenog naziva ima pridruženu tekstualnu datoteku u kojoj su zapisane oznake objekata za tu sliku, pri čemu je svaka oznaka u svome retku. Oznake slijedno sadrže razred objekta, centroid objekta po x-osi, centroid objekta po y-osi, te širinu i visinu objekta. Koordinate okvira moraju biti normalizirane, tj. u rasponu od 0 do 1. Stoga su x koordinata centroida i širina okvira podijeljene sa širinom cijele slike, a y koordinata okvira i visina okvira sa visinom cijele slike. Oznake razreda indeksirane su od nule, pri čemu u vlastitom skupu podataka oznaka 0 označava dijete, a oznaka 1 odraslu osobu. Objašnjene opisanog YOLO formata prikazano je na slici (Slika 5.1) koja nije dio vlastitog skupa podataka i služi isključivo za lakšu vizualizaciju.



Slika 5.1: Skica slike i pripadajućih oznaka u YOLO formatu. (preuzeto dana 21.06.2022. s: https://github.com/ultralytics/yolov5/issues/2293)

Također, s obzirom na to da je za potrebe ovog rada korištena biblioteka Detectron2 (koji će biti detaljnije objašnjen u poglavlju 5.3), vlastiti skup podataka prilagođen je u skladu s definiranim specifikacijama biblioteke. Napravljena je posebna verzija skupa podataka u kojoj uređena četvorka koja definira okvir nije normalizirana i ne sadrži centroid objekta kako je u YOLO formatu, već se sastoji redom od nenormaliziranih x i y koordinata gornjeg lijevog ruba te širine i visine. Iako se ovakva prilagodba oznaka u format koji Detectron2 podržava mogla raditi i direktno pri učitavanju skupa podataka, to je napravljeno unaprijed kako bi se izbjegli nepotrebni dodatni izračuni prilikom učitavanja podatkovnog skupa.

5.2. Sastav vlastitog skupa podataka

Vlastito izrađeni skup podataka sastoji se od ukupno 1200 slika. Detaljniji sastav skupa podataka vidljiv je u tablici (Tablica 1).

Tablica 1: Sastav vlastito izrađenog skupa podataka koji je korišten za eksperimentalni dio rada.

	BROJ SLIKA	**BROJ OZNAKA**
DIJETE	565	1234
ODRASLA OSOBA	570	1157
UKUPNO	1135 + 65 slika na kojima su i djeca i odrasle osobe = 1200	2391

Validacijski skup podataka, a tako i testni skup, stvoren je na način da je za svaki od njih uzeto otprilike 10% od ukupnog broja oznaka kako bi skupovi za učenje (80%), validaciju (10%) i testiranje (10%) bili približno balansirani po udjelu oznaka pojedinih klasa kako je prikazano u grafikonu na slici (Slika 5.2). Na slici je prikazan broj oznaka pojedinog razreda te njihov udio u podatkovnim skupovima za učenje, odnosno validaciju i testiranje.

Slika 5.2: Graf sastava skupa za učenje (lijevo), validacijskog skupa (sredina), te skupa za testiranje (desno). Prikazani su brojevi oznaka pojedinih razreda, a zatim i njihovi udjeli u pojedinim skupovima.

5.3. Biblioteka Detectron2

Za potrebe provođenja eksperimentalnog dijela rada korištena je biblioteka Detctron2. Detectron2 je biblioteka koju je razvio Facebook-ov AI Research i koja pruža implementirane modele današnjeg stanja tehnike računalnog vida za probleme detekcije objekata, segmentacije instanci, detekcije ključnih točaka osoba te panoptičke segmentacije.

Arhitekture za problem detekcije objekata naučeni su na COCO skupu podataka. Jedna od arhitektura koju biblioteka podržava je Faster R-CNN koja je ujedno i korištena za provođenje eksperimenata u radu. Biblioteka nudi tri različite opcije za odabir kralježnice modela. Prva opcija (FPN) je korištenje mreže ResNet uz arhitekturu FPN i standardne potpuno povezane glave za predviđanje okvira. Ova opcija zadržava najbolji kompromis između brzine i točnosti. Druga opcija (C4) zapravo predstavlja model kakav je originalno predložen u Faster R-CNN znanstvenom radu. Posljednja opcija (DC5, odnosno Dilated-C5)

implementira ResNet kralježnicu uz korištenje dilatacije nad određenim konvolucijskim slojevima te standardne potpuno povezane glave za predikciju okvira, kako je predloženo u znanstvenom radu [20].

Arhitekture za detekciju objekata učene su duljinom treniranja 3x (otprilike 37 COCO epoha) ili 1x (otprilike 12 COCO epoha). Neki od modela Faster R-CNN arhitekture i njihove performanse prikazane su u tablici (Tablica 2).

Tablica 2: (preuzeto dana 22.06.2022. s: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md)

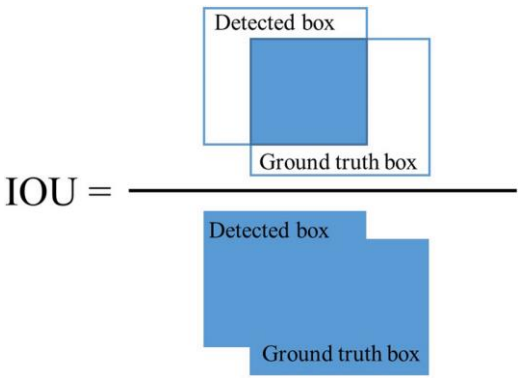
Arhitektura	način učenja	vrijeme zaključivanja (s)	AP
R50-FPN	1x	0.038	37.9
R50-C4	3x	0.104	38.4
R50-DC5	3x	0.070	39.0
R50-FPN	3x	0.038	40.2
R101-C4	3x	0.139	41.1
R101-DC5	3x	0.086	40.6
R101-FPN	3x	0.051	42.0

6. Eksperimenti i rezultati na vlastitom skupu podataka

Cilj provođenja eksperimenata na vlastitom skupu podataka bio je naučiti model da detektira osobe te pri tome razlikuje djecu od odraslih osoba. S obzirom na to da su svi korišteni modeli predtrenirani na COCO skupu podataka, logično je očekivati da će oni i bez dodatnog učenja moći detektirati odrasle osobe iz razloga što je jedan od razreda u COCO skupu podataka upravo klasa osoba (engl. *person*). Također, to znači da predtrenirani modeli već imaju naučene određene značajke specifične za ljude koje se mogu iskoristiti i time olakšati razlikovanje djece i odraslih osoba.

6.1. Evaluacijske mjere

Osnovne metrike koje se koriste za evaluaciju modela za detekciju objekata su preciznost, odziv, prosječna preciznost te srednja prosječna preciznost. Kako bi se ove mjere mogle izračunati, potrebno je definirati na koji način se određuje je li neka detekcija točna ili ne. Modeli za detekciju objekata lokaliziraju objekte na način da ih uokvire pravokutnikom i svrstaju u određenu klasu. Točnost detekcije objekta nad područjem interesa određuje se omjerom presjeka i unije okvira (već spomenuti IoU) konceptualno skiciranog na slici (Slika 6.1).



Slika 6.1: Ilustracija formule za izračun omjera presjeka i unije (IoU). (preuzeto dana 22.06.2022. s: https://www.researchgate.net/figure/Illustration-of-intersection-over-union-IOU_fig5_346512249)

Detektirani okvir koji se savršeno preklapa s oznakom imat će IoU jednak jedan, dok pomaknuti okviri ovisno o pomaku imaju vrijednost manju od jedan. Tijekom evaluacije modela detekcije objekata najčešće se smatra da je svaki predviđeni okvir koji je točno klasificiran i ima IoU vrijednost veću od 0.5 točna detekcija.

Evaluacijske metrike definiraju se pomoću broja točno pozitivnih (engl. *true positive*), pogrešno pozitivnih (engl. *false positive*) i pogrešno negativnih (engl. *false negative*) primjera. Pri tome su točno pozitivni primjeri oni kojima okviri točno detektiraju objekt prema uvjetima koje postavlja IoU , pogrešno pozitivni oni koji nisu zadovoljili taj uvjet, a pogrešno negativni oni koje model za detekciju uopće nije prepoznao.

Prva spomenuta evaluacijska mjera je preciznost (engl. *precision*). Ona je definirana formulom (6.1) te govori o tome koliki je udio točno klasificiranih primjera u skupu pozitivno klasificiranih primjera.

$$P = \frac{TP}{TP + FP} \tag{6.1}$$

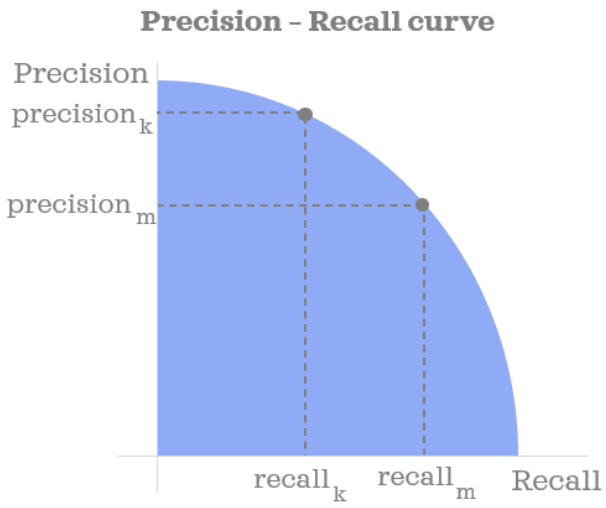
Odziv (engl. *recall*) je definiran formulom (6.2) te govori o tome koliki je udio točno klasificiranih primjera u skupu svih pozitivnih primjera.

$$R = \frac{TP}{TP + FN} \tag{6.2}$$

Na konceptualnoj razini, preciznost daje odgovor na pitanje koliko od točnih detekcija je relevantno, dakle pokazuje sposobnost modela da identificira samo relevantne elemente, a odziv odgovara na pitanje koliko je relevantnih elemenata točno detektirano, dakle pokazuje sposobnost modela da pronađe sve relevantne elemente.

Evaluacijska mjera koja povezuje preciznost i odziv naziva se prosječna preciznost.

Ona se temelji na krivulji preciznost-odziv (engl. *precision-recall curve*) koja se crta na način da se za određeni odziv računa preciznost. Pri tome se vrijednost odziva kontrolira nekim parametrom detekcije, većinom pragom sigurnosti modela detekcije. Na samom početku se prag sigurnosti postavlja na visoku vrijednost tako da ne bude točno klasificiranih primjera i time vrijednost odziva bude jednaka nuli. Zatim se prag sigurnosti postupno smanjuje, a time vrijednost odziva raste, dok istovremeno vrijednost preciznosti pada. Ova krivulja prikazana je na slici (Slika 6.2).



Slika 6.2: Skica krivulje preciznost-odziv. (preuzeto dana 22.06.2022. s: https://miro.medium.com/max/1024/1*KZu3UEBx3UIgOvdS6V_h_A.png)

Prosječna preciznost (engl. *average precision*) definirana je formulom (6.3) te ona predstavlja površinu ispod krivulje preciznost-odziv.

$$AP = \int_0^1 p(r)dr \tag{6.3}$$

Budući da se preciznost i odziv kreću u intervalu od nule do jedinice uključeno, iznos prosječne preciznosti također će biti u tom intervalu. Prosječna preciznost koristi se samo kod binarne klasifikacije.

Srednja prosječna preciznost (mAP, engl. *mean average precision*) je aritmetička sredina prosječne preciznosti te se koristi u slučaju višeatributne klasifikacije (engl. *multi-label classification*). Računa se formulom (6.4).

$$mAP = \frac{1}{K}\sum_{i=0}^{K-1} AP_i \tag{6.4}$$

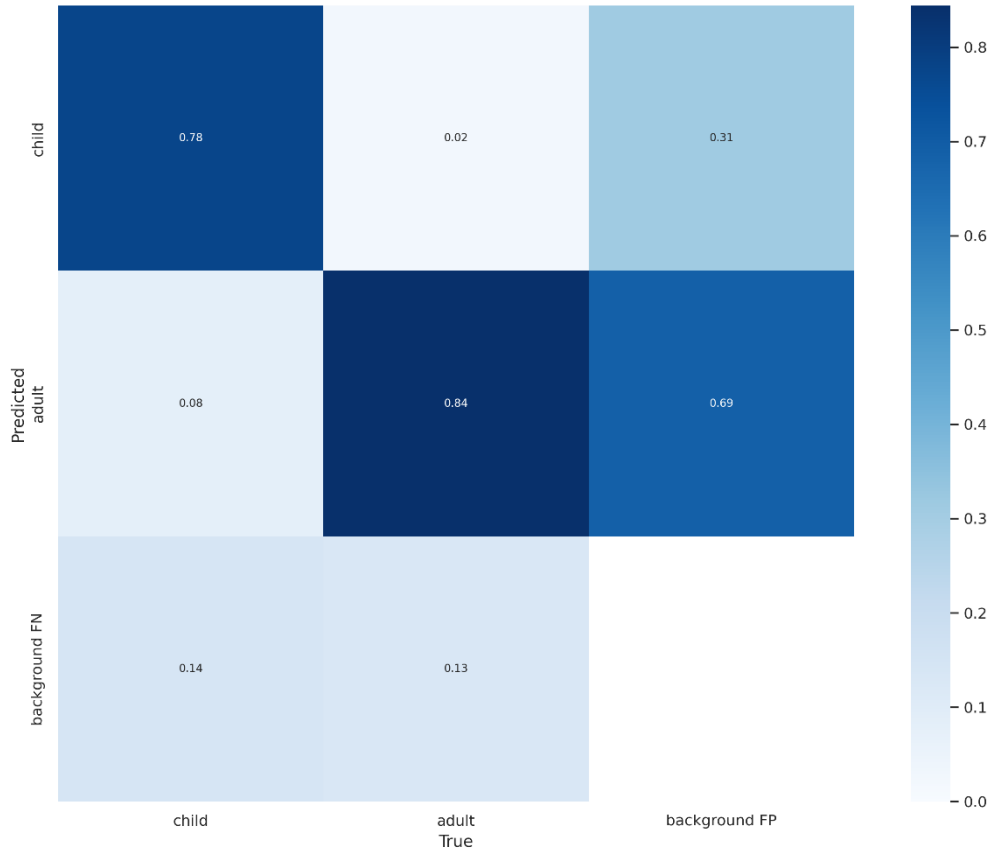
Često se računa evaluacijska metrika koja uprosječuje srednju prosječnu preciznost (mAP) za različite IoU pragove od 0.5 do 0.95. Ova metrika označava se kao $mAP@[0.5{:}\,0.95]$.

6.2. Eksperimenti nad arhitekturom YOLOv5

6.2.1. Učenje svih slojeva mreže

Prvi provedeni eksperiment je učenje modela YOLOv5, točnije verzije YOLOv5x koja je najkompleksnija i ima 86.7 milijuna parametara. Za inicijalizaciju modela korištene su odgovarajuće težine predtrenirane na COCO skupu podataka kako je prethodno napomenuto. Učenje modela provedeno je na način da su se učili svi slojevi mreže, odnosno da nijedan sloj mreže nije bio zamrznut. Model je treniran na 250 epoha uz veličinu mini-grupe 4 te uz korištenje Adam optimizatora s podrazumijevanim hiperparametrima. Početni korak učenja veličine je 0.01 te se smanjuje nakon svake epohe sukladno pravilima kosinusnog kaljenja (engl. *cosine annealing scheduler*) [21] do minimalne vrijednosti od 10^{-4}. Prigušenje težina (engl. *weight decay*) je vrijednosti $5 \cdot 10^{-4}$.

Dobiveni su sljedeći rezultati. Na slici (Slika 6.3) prikazana je matrica zabune dobivena nad skupom za validaciju.



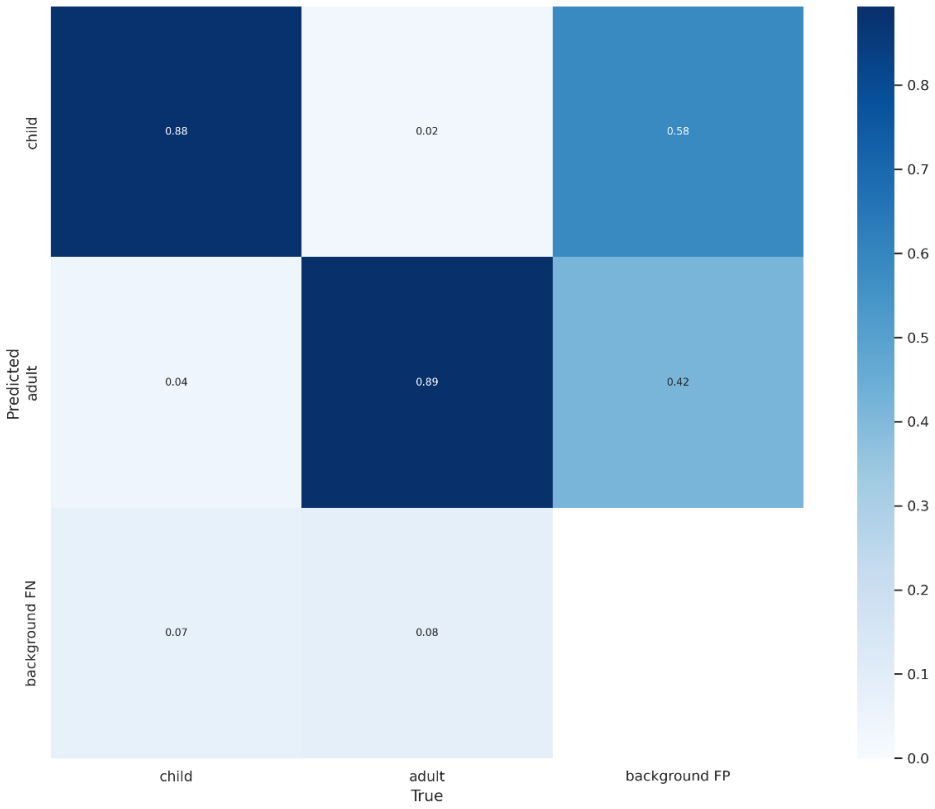
Slika 6.3: Matrica zabune dobivena nad validacijskim skupom podataka.

Na slici (Slika 6.4) prikazani su primjeri predviđanja modela na nekima od slika iz validacijskog skupa.

Slika 6.4: Neki od primjera izlaza modela za nekolicinu slika iz validacijskog skupa.

6.2.2. Učenje sa zamrzavanjem kralježnice modela

Ovaj eksperiment podrazumijeva učenje modela YOLOv5x na način da se zamrzne kralježnica modela, odnosno prvih 10 slojeva mreže. Učenje je provedeno na način da su svi hiperparametri ostali identični onima u prethodnom eksperimentu iz odjeljka 6.2.1. Na slici (Slika 6.5) prikazana je matrica zabune na validacijskom skupu, a na slici (Slika 6.6) primjeri izlaza modela na nekima od slika iz validacijskog skupa.



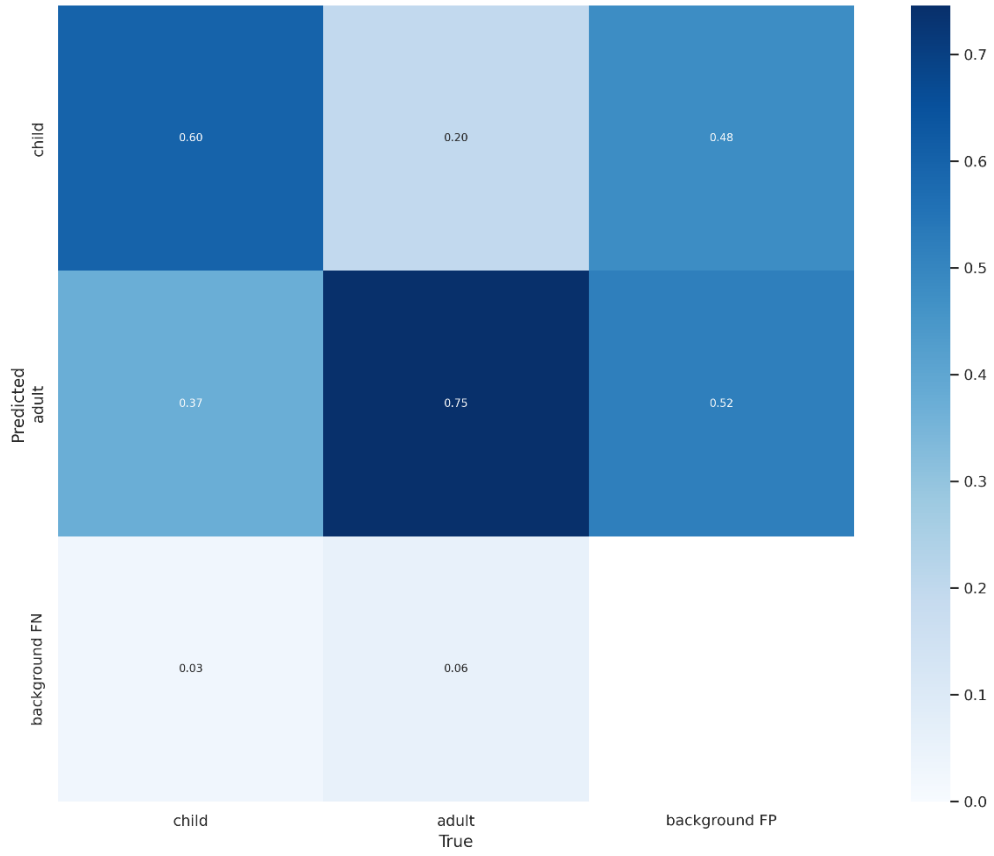
Slika 6.5: Matrica zabune dobivena nad validacijskim skupom podataka.

Slika 6.6: Primjeri izlaza modela za neke od slika iz validacijskog skupa podataka.

6.2.3. Učenje sa zamrzavanjem svih osim posljednjeg sloja

Sljedeći provedeni eksperiment također je učenje modela YOLOv5x, no s razlikom u tome da su zamrznuti svi slojevi mreže osim posljednjeg izlaznog konvolucijskog sloja u detekcijskoj glavi. Dakle, zamrznuta su 24 sloja mreže. Ideja iza ovakvog *„transfer learninga“* sa zamrznutim slojevima proizašla je iz činjenice da modelu koji je već predtreniran da detektira ljude nije potrebno previše mijenjanja težina kako bi naučio razaznati i djecu od odraslih osoba. Ovaj eksperiment proveden je na 60 epoha s obzirom na to da je ovakvoj vrsti učenja sa zamrznutim slojevima potrebno puno manje epoha do konvergencije. Svi ostali hiperparametri su isti kao u prethodnom eksperimentu iz odjeljka 6.2.1.

Na slici (Slika 6.7) prikazana je matrica zabune dobivena nad skupom za validaciju.



Slika 6.7: Matrica zabune dobivena nad validacijskom skupu podataka.

Na slici (Slika 6.8) prikazani su primjeri predviđanja modela na nekima od slika iz validacijskog skupa.

Slika 6.8: Primjeri izlaza modela za neke od slika iz validacijskog skupa podataka.

6.2.4. Usporedba dvaju eksperimenata na mreži YOLOv5x

U tablici (Tablica 3) prikazani su preciznost, odziv, te AP (za pragove 0.5 te 0.95) za oba razreda iz validacijskog podatkovnog skupa za eksperimente iz 5.1.1 i 5.1.2 usporedno, pri čemu sivi stupci označavaju rezultate eksperimenta 5.1.1 (učenje na svim slojevima).

Tablica 3: Rezultati dobiveni na validacijskom skupu pri evaluaciji oba YOLOv5x naučena modela. S lijeva na desno stupci označavaju rezultate učenja na svim slojevima, zatim učenje uz zamrznutu kralježnicu i naposljetku učenje uz zamrzavanje svih osim posljednjeg sloja.

	preciznost (%)			**odziv (%)**			**AP@0.5 (%)**			**AP@0.5:0.95 (%)**		
dijete	79.3	96.1	91.8	83.0	88.4	79.6	87.2	96.1	91.2	47.1	64.1	68.3
odrasla osoba	82.1	97.2	66.9	82.8	86.8	81.1	89.0	96.1	81.8	52.1	66.4	56.4
ukupno	80.7	96.6	79.3	82.9	87.6	80.4	88.1	96.1	86.5	49.6	65.3	62.3

Tablica 4: Rezultati dobiveni na testnom skupu pri evaluaciji oba YOLOv5x naučena modela. S lijeva na desno stupci označavaju rezultate učenja na svim slojevima, zatim učenje uz zamrznutu kralježnicu i naposljetku učenje uz zamrzavanje svih osim posljednjeg sloja.

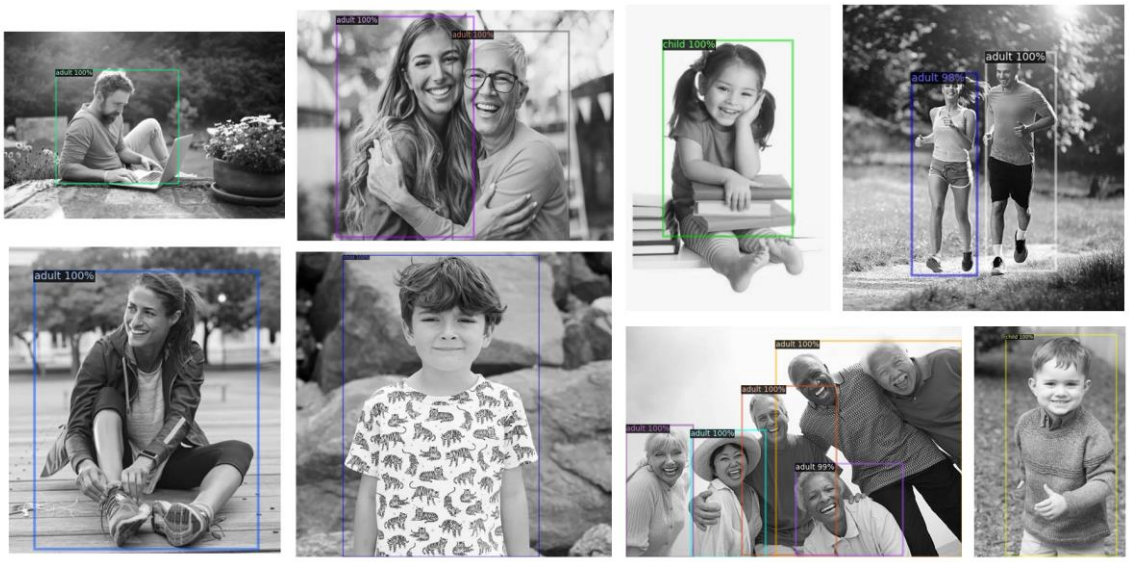
	preciznost (%)			**odziv (%)**			**AP@0.5 (%)**			**AP@0.5:0.95 (%)**		
dijete	75.2	92.7	71.5	83.7	90.8	74.5	84.1	94.2	81.5	41.5	63.9	53.6
odrasla osoba	73.6	84.7	73.3	80.8	80.8	76.9	80.0	88.2	80.8	41.5	59.0	53.8
ukupno	74.4	88.7	72.4	82.2	85.8	75.7	82.0	91.2	81.2	41.5	61.5	53.7

Tablica pokazuje kako model kojemu su zamrznuti slojevi kralježnice prilikom učenja postiže bolje rezultate od ostalih modela. Model kojemu su bili zamrznuti svi slojevi osim posljednjeg znatno je lošiji. Objašnjenje iza toga jest da posljednji sloj nema dovoljno kapaciteta za učenje. Model kojemu su učeni svi slojevi postiže najgore rezultate, a iza toga stoji zaboravljanje značajki. Naime, s obzirom na to da su težine bile prednaučene na COCO skupu koji sadrži klasu osobe, mijenjanje svih značajki tijekom učenja dovelo je do lošijih rezultata, dok je zamrzavanje kralježnice sačuvalo neke od korisnih i već naučenih značajki.

6.3. Eksperimenti nad arhitekturom Faster R-CNN

6.3.1. Učenje svih slojeva mreže

Ovaj eksperiment odnosi se na učenje mreže Faster R-CNN u sklopu biblioteke Detectron2 koja je predtrenirana na skupu podataka COCO uz kralježnicu ResNet-50 i FPN modul s načinom učenja 3x (objašnjeno u poglavlju 5.3). Učenje se provodilo nad svim slojevima mreže, na 250 epoha uz veličinu mini-grupe jednaku 4 te SGD optimizator uz korak učenja od 10^{-4}. Na slici (Slika 6.9) prikazana su predviđanja modela na nekima od slika iz validacijskog skupa.



Slika 6.9: Izlazi modela za neke od slika iz validacijskog skupa.

6.3.2. Učenje sa zamrznutom kralježnicom modela

Nakon učenja svih slojeva modela Faster R-CNN, provedeno je učenje istog modela uz zamrzavanje kralježnice modela kako bi se postigao *„transfer learning“*. Model je analogno eksperimentu opisanom u odjeljku 6.2.1 inicijaliziran težinama naučenima na COCO skupu podataka, učen uz veličinu mini-grupe jednaku 4 i SGD optimizator. Učenje je provedeno na 250 epoha. Na slici (Slika 6.10) prikazani su izlazi modela na odabranim slikama iz validacijskog skupa.

Slika 6.10: Izlazi modela za neke od slika iz validacijskog skupa.

6.3.3. Usporedba dvaju eksperimenata na mreži Faster R-CNN

U tablici (Tablica 5) prikazana je usporedba rezultata dobivenih na validacijskom skupu podataka pri učenju Faster R-CNN arhitekture sa i bez zamrzavanja slojeva mreže.

Tablica 5: Usporedba rezultata modela Faster R-CNN dobivenih učenjem na svim slojevima i uz zamrzavanje slojeva na validacijskom skupu.

	učenje svih slojeva (5.2.1)	**učenje sa zamrznutim slojevima (5.2.2)**
odziv (%)	72.22	74.53
AP@0.5:0.95 dijete (%)	68.30	68.21
AP@0.5:0.95 odrasla osoba (%)	61.00	66.74
mAP@0.5:0.95 (%)	64.65	67.47
mAP@0.5 (%)	91.21	91.14

Tablica 6: Usporedba rezultata modela Faster R-CNN dobivenih učenjem na svim slojevima i uz zamrzavanje slojeva na testnom skupu.

	učenje svih slojeva (5.2.1)	**učenje sa zamrznutim slojevima (5.2.2)**
odziv (%)	67.91	68.83
AP@0.5:0.95 dijete (%)	62.85	64.86
AP@0.5:0.95 odrasla osoba (%)	57.40	58.06
mAP@0.5:0.95 (%)	60.12	61.46
mAP@0.5 (%)	90.40	90.05

Iz tablice je vidljivo da prema svim praćenim metrikama model kojemu su učeni svi slojevi ima lošije performanse od modela kojemu je zamrznuta kralježnica. Interpretacija iza toga analogna je onoj za učenje modela YOLOv5.

6.4. Usporedba dobivenih rezultata dvaju modela

Naposljetku možemo usporediti razliku među rezultatima na testnom skupu koje daju modeli YOLOv5x i Faster R-CNN prikazanu u tablici (Tablica 7). Usporedba dobivenih metrika napravljena je nad mrežama kojima je zamrznuta kralježnica modela, odnosno nad eksperimentima provedenima u poglavljima 6.2.2 i 6.3.2, s obzirom na to da su ti modeli dali najbolje rezultate. Brzina zaključivanja za pojedinu sliku mjerena je uz korištenje grafičke kartice NVIDIA Quadro P5000.

Tablica 7: Usporedba rezultata modela YOLOv5 i Faster R-CNN.

	mreža YOLOv5	**mreža Faster R-CNN**
AP@0.5:0.95 dijete (%)	63.9	64.86
AP@0.5:0.95 odrasla osoba (%)	59.0	58.06
mAP@0.5:0.95 (%)	**61.5**	**61.46**
mAP@0.5 (%)	91.2	90.05
vrijeme zaključivanja (ms)	**30.12**	**1207.51**

Iz tablice se može vidjeti kako mreža Faster R-CNN i mreža YOLOv5 imaju otprilike jednake performanse u vidu srednje prosječne preciznosti od mreže YOLOv5, iako je bilo za očekivati da će model Faster R-CNN imati veću prosječnu preciznost s obzirom na to da dvoprolazni detektori općenito imaju bolju točnost od jednoprolaznih detektora. Ipak, ako promatramo eksperimente u kojima se uče svi slojevi mreže, tada Faster R-CNN zaista postiže puno bolje rezultate.

Također, vidljivo je da model YOLOv5 ima znatno veću brzinu zaključivanja, što je očekivano za jednoprolazne detektore. Model YOLOv5 bio bi pogodan za rad u realnom vremenu, dok bi model Faster R-CNN imao preveliko kašnjenje.

Zaključak

Ovaj rad se bavi jednim od glavnih problema računalnog vida, detekcijom objekata, čija je svrha lokalizirati sve objekte postojećih razreda te ih svrstati u ispravne klase. U radu su objašnjeni koncepti na kojem počivaju konvolucijske neuronske mreže, operacija konvolucije i operacija sažimanja, na kojima se temelji većina dubokih modela današnjeg stanja tehnike.

U posljednje vrijeme vrlo dobre rezultate u rješavanju problema detekcije objekata postižu jednoprolazni i dvoprolazni modeli koje ovaj rad razmatra. Napravljena je kratka usporedba obje vrste modela te je navedeno kako jednoprolazni detektori iz ulazne slike uče vjerojatnosti pojedinih razreda i koordinate okvira objekata. Objašnjeno je da dvoprolazni detektori imaju dvije faze, od kojih prva izlučuje regije interesa, a druga iskorištava navedene regije radi klasifikacije te regresije okvira. Navedeno je kako dodatan prolaz generalno poboljšava performanse dvoprolaznih detektora u vidu točnosti, no brzina zaključivanja im je znatno manja.

Cilj rada bio je uhodati učenje jednim jednoprolaznim i jednim dvoprolaznim modelom, evaluirati naučene modele, prikazati postignutu točnost odgovarajućim evaluacijskim mjerama, izmjeriti brzinu zaključivanja i usporediti dobivene rezultate. Od jednoprolaznih modela, u radu je izdvojen model YOLOv5, a od dvoprolaznih Faster R-CNN, pri čemu su detaljno opisane i obrađene obje arhitekture te ideje iza njih. Navedene dvije arhitekture korištene su u eksperimentalnom dijelu rada.

U sklopu rada, izrađen je vlastiti skup podataka u svrhu učenja modela da detektira osobe te razlikuje odrasle osobe od djece. Eksperimentalni dio rada obuhvaća učenje modela YOLOv5 i Faster R-CNN na vlastitom skupu podataka uz učenje svih slojeva mreže te zamrzavanje pojedinih slojeva mreže.

Pokazano je kako su rezultati u pogledu evaluacijskih mjera otprilike jednako dobri kod modela YOLOv5 i modela Faster R-CNN, iako je bilo očekivano da će model Faster R-CNN davati bolje metrike, s obzirom na to da je Faster R-CNN dvoprolazni model. Također, mjerenjem brzine zaključivanja modela, zaključeno je da je model YOLOv5 kao jednoprolazni model doista brži od dvoprolaznog Faster R-CNN-a, što ga čini pogodnim za rad u realnom vremenu.

Literatura

[1] I. Fabijanić, "Duboki konvolucijski modeli za praćenje objekata," Diplomski rad, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2017.

[2] L. Pleše, „R-FCN i Soft-NMS G," Seminarski rad, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2021.

[3] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[4] D. Miko, "Duboki konvolucijski modeli za lokalizaciju objekata," Diplomski rad, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2018.

[5] W. Zhe, "YOLOv5 (6.0/6.1) brief summary" https://github.com/ultralytics/yolov5/issues/6998, datum pristupa: 10.06.2022.

[6] C. Wang, H. M. Liao, I. Yeh, Y. Wu, P. Chen, and J. Hsieh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," CoRR, vol. abs/1911.11929, 2019.

[7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," CoRR, vol. abs/1804.02767, 2018.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," CoRR, vol. abs/1406.4729, 2015.

[9] T. Lin, P. Dollar, R. Girshick, K. He, B. Haiharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," CoRR, vol. abs/1612.03144, 2017.

[10] S. Liu, L. Qi, H. Qin, J. Shi, and J.Jia, "Path Aggregation Network for Instance Segmentation," CoRR, vol. abs/1803.01534v4, 2018.

[11] M. D. Zeiler, and R. Fergus, "Visualizing and Understanding Convolutional Networks," CoRR, vol. abs/1311.2901, 2013.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," CoRR, vol. abs/1311.2524, 2014.

[13] J. R.R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," 2012.

[14] R. Gandhi, “R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms” https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e, datum pristupa: 19.06.2022.

[15] R. Girshick, “Fast R-CNN,” CoRR, vol. abs/1504.08083, 2015.

[16] A. Yelisetty, “Understanding Fast R-CNN and Faster R-CNN for Object Detection.” https://medium.com/towards-data-science/understanding-fast-r-cnn-and-faster-r-cnn-for-object-detection-adbb55653d97, datum pristupa: 19.06.2022.

[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” CoRR. Vol. abs/1506.01497v3, 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” The 14th European Conference on Computer Vision, 2016.

[19] J. Bratulić, “Semantička segmentacija kolničkih trakova,” Završni rad, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2020.

[20] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Yu, and Y. Wei, “Deformable Convolutional Networks,” CoRR, vol. abs/1703.06211, 2017.

[21] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts.,” CoRR, vol. abs/1608.03983, 2016.

Sažetak

Vrednovanje metoda za detekciju osoba u slikama

U ovom radu obrazložen je problem detekcije objekata te princip funkcioniranja jednoprolaznih i dvoprolaznih detektora. Objašnjene su razlike među njima te su detaljno opisane arhitekture modela YOLOv5 i Faster R-CNN kao predstavnika obje vrste detektora. Uhodano je učenje navedenih mreža na vlastitom skupu podataka s ciljem razvoja modela koji detektira ljude te razlikuje djecu od odraslih osoba. Napravljena je usporedba rezultata koje daju oba modela te su oni navedeni na kraju rada.

Ključne riječi: detekcija objekata, jednoprolazni detektori, dvoprolazni detektori, YOLOv5, Faster R-CNN

Abstract

Evaluation of methods for detection of people in images

This paper explains the problem of object detection and the principle of functioning of single-shot and two-shot detectors. The differences between them are explained in the paper and the architectures of YOLOv5 and Faster R-CNN models are described in detail, as they are representatives of both types of detectors. These models are trained on a custom dataset with the aim of developing a model that detects people and distinguishes children from adults. A comparison of the results given by both models is made and they are listed at the end of the paper.

Keywords: object detection, single-shot detectors, two-shot detectors, YOLOv5, Faster R-CNN