Automatska detekcija voća

Perić, Luka

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:612252

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij računarstvo

AUTOMATSKA DETEKCIJA VOĆA

Diplomski rad

Luka Perić

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 14.09.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Luka Perić
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1150R, 13.10.2020.
OIB studenta:	79217241990
Mentor:	izv. prof. dr. sc. Emmanuel Karlo Nyarko
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Ivan Vidović
Član Povjerenstva 1:	izv. prof. dr. sc. Emmanuel-Karlo Nyarko
Član Povjerenstva 2:	prof. dr. sc. Robert Cupec
Naslov diplomskog rada:	Automatska detekcija voća
Znanstvena grana diplomskog rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Zadatak diplomskog rada:	Potrebno je razviti sustav koji na temelju slike dobivene kamerom detektira voće koje je pakirano u otvorenoj kutiji. Osim detekcije voća, sustav treba odrediti njihov broj te procijeniti veličinu. (Tema rezervirana za: Luka Perić)
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	14.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 02.10.2023.

Ime i prezime studenta:	Luka Perić
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1150R, 13.10.2020.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Automatska detekcija voća**

izrađen pod vodstvom mentora izv. prof. dr. sc. Emmanuel Karlo Nyarko

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1
2. PREGLED PODRUČJA TEME 3
3. PODATKOVNI SKUP 5
3.1. Prikupljanje slika 5
3.2. Anotacija slika 6
3.3. Augmentacija slika 7
4. DETEKCIJA VOĆA 9
4.1. Faster RCNN 9
4.2. YOLOv4 14
5. PROCJENA VELIČINE I BROJANJE VOĆA 17
5.1. Segmentacija slike pomoću boje 17
5.2. Houghova detekcija krugova 18
6. REZULTATI 20
7. ZAKLJUČAK 25
LITERATURA 26
SAŽETAK 31
ABSTRACT 32
PRILOZI 33

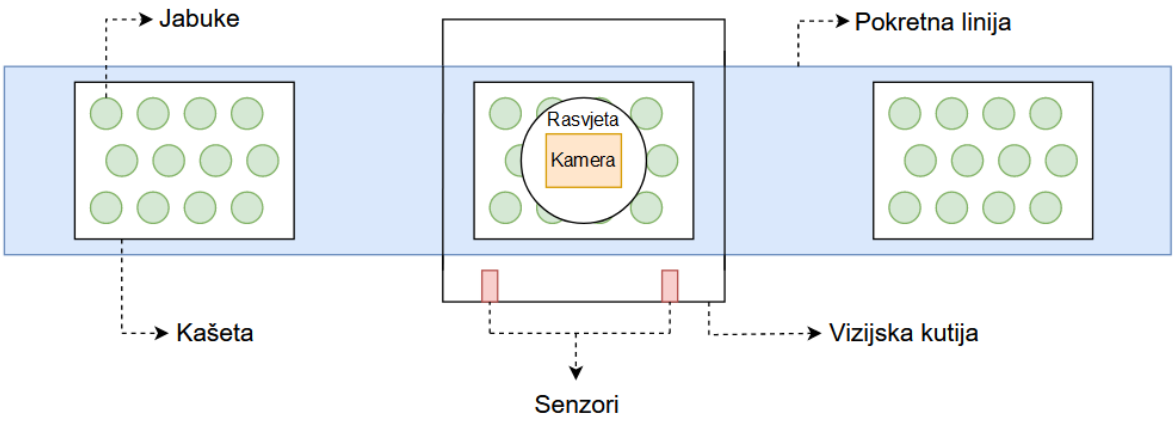
1. UVOD

Kontrola proizvodnje nezaobilazni je čimbenik u industrijskim postrojenjima. Ona omogućuje rano prepoznavanje neispravnog proizvoda te tako smanjuje lošu kvalitetu i troškove. Time se osigurava efektivna, efikasna i ekonomična proizvodnja. Pojavom automatizacije sve više se smanjuje potreba za ljudskom snagom u tom sektoru. Jedna od industrija koja koristi kontrolu proizvodnje jest prehrambena odnosno voćarska industrija. Nakon što se voće ubere (na poljima, u plastenicima ili slično), postoje razne opcije gdje ono može završiti. Voće se nakon branja može skladištiti u hladnjačama, pakirati za transport ili direktno stavljati na tržište. Pri tome voće prolazi kroz kontrolu kakvoće. Pravilnik o kakvoći voća propisuje zahtjeve kakvoće koje se odnose na klasifikaciju, kategorizaciju i senzorna svojstva, uvjete pakiranja, prezentiranja i način označavanja [1]. Sve više kontrolu proizvodnje (kakvoće), umjesto ljudi, nadziru strojevi. Polja, voćnjake i plastenike počinju nadzirati roboti i bespilotne letjelice, dok se u industrijskim postrojenjima na pokretnim linijama postavljaju strojevi koji kontroliraju kakvoću voća [2]. Voće se po pokretnoj liniji najčešće kreće (rasuto) jedno po jedno ili u kartonskim (ili plastičnim) kutijama. U kutijama (pogotovo onih Ekstra klase) se nalaze kartonski podlošci koji služe za zaštitu od mehaničkih oštećenja prilikom transporta, za lakše skladištenje voća te radi prezentiranja (Slika 1.1.).

Slika 1.1. *Primjer kartonske kutije za odlaganje voća*

Pomoću transportne linije, kutija s voćem se kreće prema sustavu gdje se određuje njezina kakvoća. Sustavi se najčešće sastoje od nekoliko senzora, koji detektiraju kutiju na određenom mjestu te kamere koja uz pomoć određenog algoritma računalnog vida ili neuronskih mreža,

procjenjuju kakvoću voća u kutiji. Nakon procjene kakvoće, kutija se šalje u daljnji postupak. Na slici 1.2. je prikazan jedan takav sustav na koji će se oslanjati ovaj rad.



Slika 1.2. *Primjer sustava za kontrolu kakvoće voća*

U ovom diplomskom radu testirati će se nekoliko algoritama računalnog vida i neuronskih mreža koje bi se mogle primijeniti na ranije opisan sustav (Slika 1.2.). Na temelju slike dobivene kamerom potrebno će biti detektirati voće koje se nalazi u kutiji. Osim detekcije voća, potrebno će biti odrediti njihov broj te procijeniti njihovu veličinu. Pregled područja ove problematike bit će dan u poglavlju 2. U poglavlju 3 bit će opisan način prikupljanja, anotacije i augmentacije slika potrebnih za izradu podatkovnog skupa koji će se primijeniti za neuronske mreže opisane u poglavlju 4. Opis testiranih algoritama za procjenu veličine voća odnosno brojanje je dan u poglavlju 5. Poglavlje 6 opisuje korištene alate te daje osvrt na rezultate testiranih algoritama korištenih u radu. Na temelju rezultata će se dati zaključak u zadnjem (7) poglavlju.

2. PREGLED PODRUČJA TEME

Pojavom i razvojem računalnog vida i neuronskih mreža omogućena je njihova primjena u raznim industrijskim granama. Jedna od industrija u kojoj pronalaze primjenu je voćarska industrija. Aktualni znanstveni i praktični dosezi često ih koriste i analiziraju na poljima, voćnjacima ili plastenicima. Kako bi ih se moglo koristiti, za njih se izrađuju podatkovni skupovi. Na internetu se mogu pronaći razni podatkovni skupovi za klasifikaciju, segmentaciju i detekciju voća. Od malih i jednostavnih skupova podataka [3, 4] (koje izrađuju entuzijasti) pa do velikih i zahtjevnijih koji su korišteni u svrhu znanstvenih radova [5, 6, 7]. Prema [6], podatkovni skup sadrži preko 40000 anotiranih instanci objekta na 1000 slika, dok [5] sadrži 1120 slika jabuka, 1964 manga i 620 badema. Skupovi se koriste u voćnjacima pa je broj slika veći kako bi neuronske mreže mogle bolje generalizirati scenu. Prije pojave neuronskih mreža, rane metode su se uglavnom oslanjale na statičke pragove boja (engl. *static color thresholds*) čije su nedostatke nadopunjavali dodatni senzori, kao što su termalne ili bliske infracrvene (NIR) kamere [8]. Neuronske mreže koje se najčešće koriste za detekciju voća su Faster RCNN i YOLO. Faster RCNN se pojavljuje 2015. godine u radu znanstvenika Rena i dr. [9] te postaje jedan od poznatijih metoda prepoznavanja objekata. Sa i dr. [10] uz pomoć Faster RCNN-a te spajanjem slika dobivenih iz dva modaliteta, RGB i NIR, prepoznaju slatke paprike. Ovaj model ostvaruje F1 mjeru od 0,838. Koristeći skup za jabuke i manga [5], Bargoti i Underwood ostvaruju F1 mjeru veću od 0,9 [11]. Koristeći Faster RCNN za detekciju breskvi, Assunção i dr. [12] ostvaruju 0,90 prosječnu preciznost (engl. *average precision*) s omjerom presjeka i unije od 0,5 (engl. *intersection over union*). YOLO algoritam se pojavljuje 2016. godine [13] te zbog svoje brzine i jednostavnosti se uvelike počinje koristiti za detekciju objekata u stvarnom vremenu. Danas postoje mnoge poboljšane varijante YOLO algoritma (v2, v3, v4 itd.). Prema [14], koristeći YOLOv4 algoritam ostvario je prosječnu preciznost od 0,9995 prilikom detekcija banana u plantažama. Parico i Ahamed [7] koriste također YOLOv4 varijante pri detekciji krušaka u stvarnom vremenu za primjenu na mobilnim telefonima. Oni ostvaruju prosječnu preciznost od 98%. Koristeći YOLO algoritam za detekciju limuna, u [15] se ostvaruje prosječna preciznost od 96,28%, a u [16] za detekciju manga 98,3% s F1 mjerom od 0,968. Rezultat od 99,5% za prosječnu preciznost ostvaren je u radu [17] za detekciju rajčice. Jako je malo istraženih i opisanih radova u području izračuna ili procjene veličine voća. Najčešće opisani algoritmi za procjenu veličine voća koriste klasifikaciju [18] ili tehnike praga (engl. *thresholding technique*) [18, 19]. Prema [18], primjenom tehnike praga, ostvarena je točnost klasifikacije rajčice od 80%. Što se tiče algoritama za brojanje voća također se često koriste tehnike praga kao što je segmentacija objekata po boji [20]. Kako

navodi [8], najviše se primjenjuje Houghova detekcija krugova [21] ili se sumiraju izlazi (prepoznati objekti) koje daje neuronska mreža.

3. PODATKOVNI SKUP

Kao što je ranije rečeno, postoje razni skupovi podataka za klasifikaciju, segmentaciju i detekciju voća. Često su podatkovni skupovi za detekciju voća prikupljeni unutar voćnjaka ili su slikani radi vlastitih primjera. U ovom diplomskom radu bit će korišten vlastito izrađen podatkovni skup. Voće za koje će se raditi podatkovni skup su jabuke sorte Granny Smith. U prilogu P.3.1. se može pronaći podatkovni skup korišten u ovom radu.

3.1. Prikupljanje slika

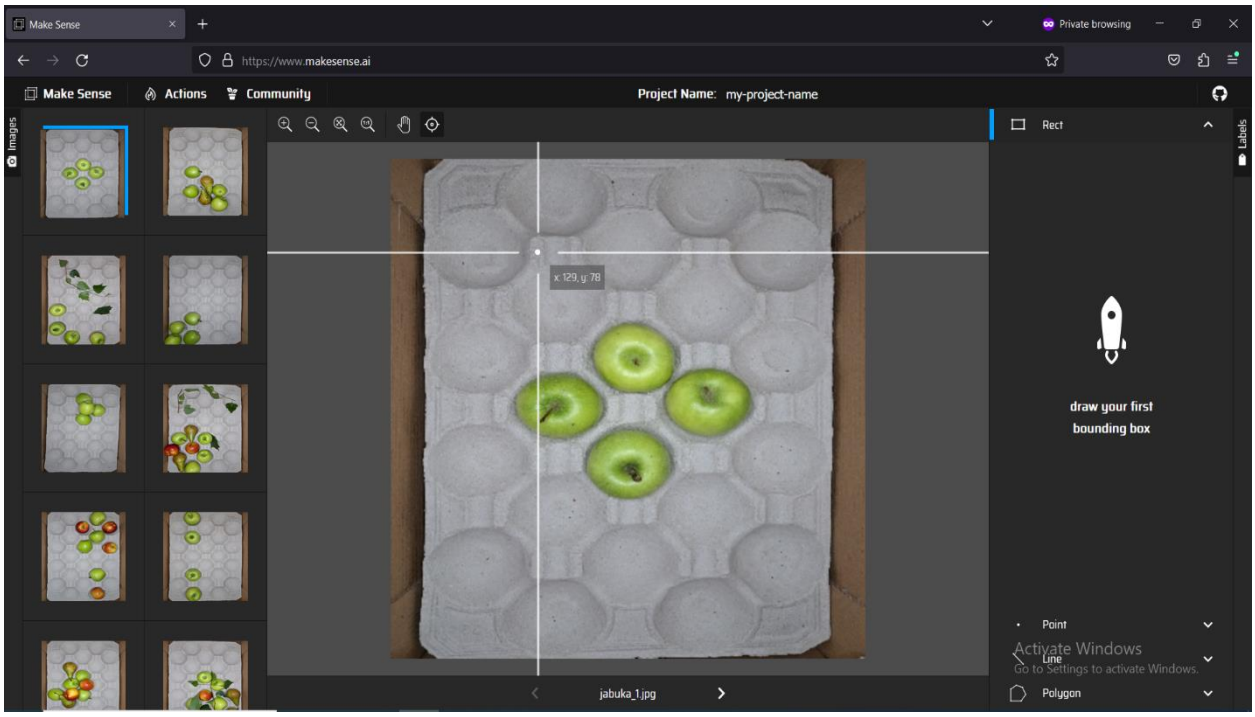
Slika 3.1. prikazuje maketu izrađenu za potrebe prikupljanja slika. Maketa je napravljena od drveta u obliku slova "C" gdje je na njezinom vrhu učvršćena prsten rasvjeta radi kontroliranog osvjetljenja. Na prsten rasvjetu je postavljen mobitel s kojim su prikupljene slike jabuka koje su se nalazile u kutiji. Uslikano je 100 slika različito raspoređenih jabuka. Osim jabuka, u kutiji su bili postavljeni i strani predmeti poput krušaka, jabuka druge sorte i lišća (Slika 3.2.). Maketa je napravljena kao primjer sustava iz industrijskog postrojenja (Slika 1.2.).

Slika 3.1. *Prikaz izrađene makete za prikupljanje slika*

Slika 3.2. *Izgled dobivenih slika za podatkovni skup*

3.2. Anotacija slika

Kako bi se jabuke mogle detektirati potrebno je anotirati slike. Anotiranje je proces označavanja podataka na slikama [22]. Tehnike dubokog učenja koriste tekstualno napisane informacije o podacima za trening algoritama. Pomoću njih sustav „nauči“ prepoznati objekte na slici. Alat s kojim je izvršena anotacija naziva se Make Sense [23]. Make Sense je besplatan web baziran alat otvorenog koda za anotiranje podatkovnih skupova (Slika 3.3.). Nakon ubacivanja slika u alat, jabuke su pravokutno označene čime se dobivaju granični okviri (engl. *bounding boxes*). Granični okviri nose informaciju o označenom objektu napisanu u određenom formatu. Anotacije su na kraju izvezene u YOLO, VOC XML i CSV formatu.



Slika 3.3. *Prozor alata Make Sense*

3.3. Augmentacija slika

Prema [24], augmentacija slika je procedura pri kojoj se skup podataka umjetno proširi primjenjivanjem različitih transformacija na slike. Neke od prednosti augmentacije su ubrzanje dobivanja dovoljno podataka za podatkovni skup, sprječavanje pretjeranog usklađivanja nad podacima (engl. *overfitting*) te poboljšanje dobivenih rezultata modela. Neke od vrsta augmentacija slika su vertikalno ili horizontalno zakretanje, izrezivanje dijelova slike, rotacija, mijenjanje kontrasta i svjetline, promjena veličine ili oštrine slike i tako dalje. One se dijele u dvije kategorije, afine augmentacije (rotacija, vertikalno zakretanje, promjena veličine slike itd.) i augmentacije na razini piksela (mijenjanje kontrasta, svjetline, oštrine itd.) [24, 25]. Kako bi se augmentacija slika provela korištena je Python biblioteka naziva Albumentations. Augmentacija s Albumentations bibliotekom je brza i fleksibilna te je široko upotrebljavana u industrijama, istraživanjima i natjecanjima iz računalnog vida i dubokog učenja, projektima otvorenog koda [26]. Albumentations sadrži više od 60 različitih metoda augmentacija slika koje se mogu koristiti za klasifikaciju, segmentaciju, prepoznavanje objekata i tako dalje. Pri augmentaciji slika za prepoznavanje objekata osim augmentacija slika moraju se augmentirati i granični okviri. Kao što je u prijašnjem potpoglavlju rečeno, anotacije su izvezene u različitim formatima. Formati koji će biti korišteni u ovom radu su VOC XML i YOLO. U VOC formatu koordinate graničnih okvira prikazuje se kao [*x_min, y_min, x_max, y_max*]. One predstavljaju vrijednosti u pikselima gdje *x_min* i *y_min* su koordinate gornjeg lijevog, a *x_max* i *y_max* donjeg desnog kuta graničnog okvira. Granični okvir u YOLO formatu se prikazuje kao [*object_id*, *x_center, y_center, width, height*]. *Object_id* predstavlja oznaku klase objekta. *X_center* i *y_center* predstavljaju (vrijednosti u pikselima) koordinate centra graničnog okvira gdje je x-os (*x_center*) podijeljen sa širinom, a y-os (*y_center*) s visinom slike kako bi vrijednosti bile normalizirane. *Width* i *height* predstavljaju širinu i visinu graničnog okvira te su također normalizirani. Albumentations podržava Pascal VOC, Albumentations, COCO i YOLO formate [27]. U ovom radu su korištene složene augmentacije kako bi se augmentirane slike što bolje razlikovale od izvornih. Slika 3.4. prikazuje primjer programskog koda složene augmentacije.

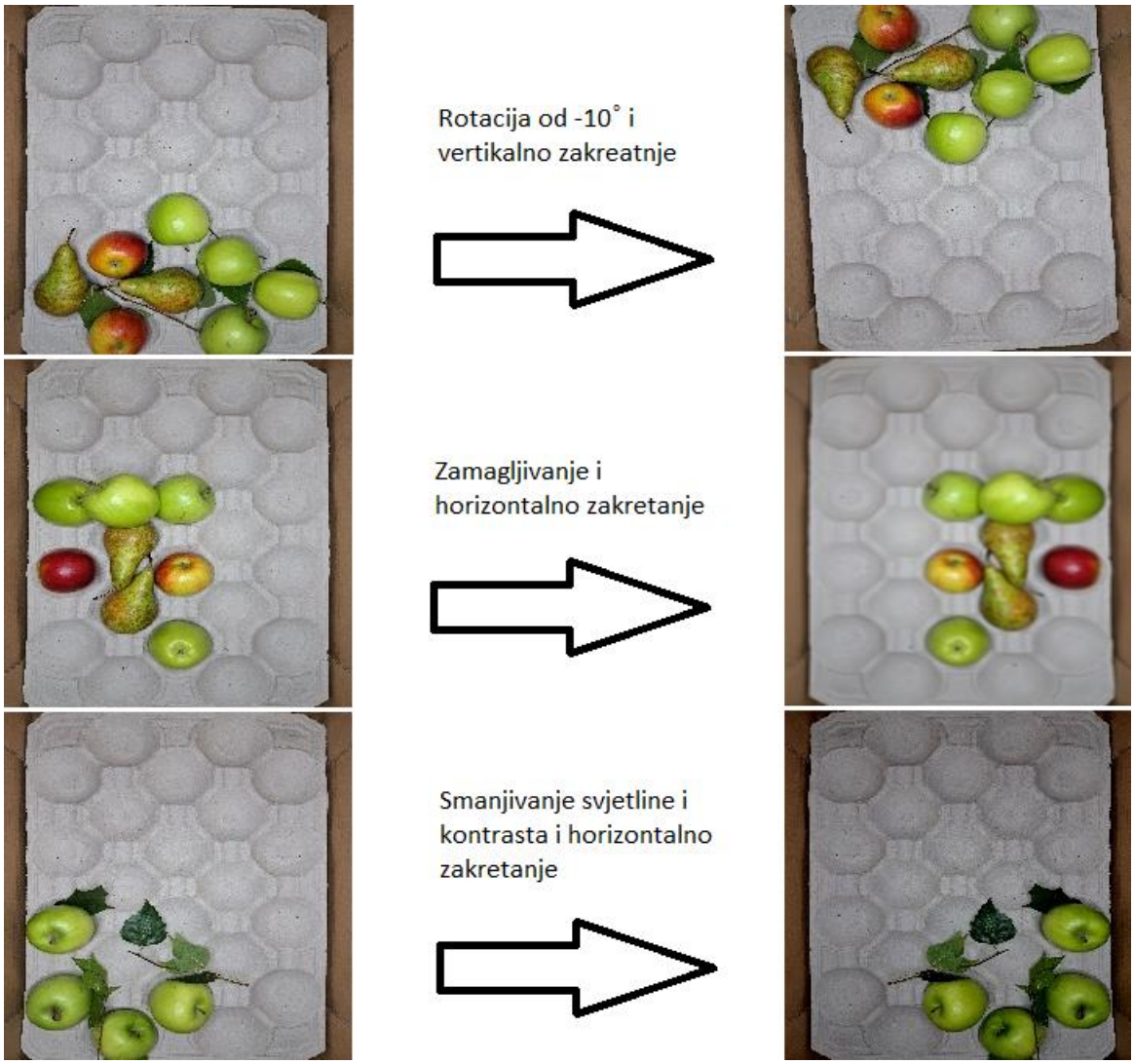
```
transform = A.Compose([
    A.Blur(blur_limit=51, always_apply=False, p=0.5),
    A.VerticalFlip(p=0.5)],
    bbox_params=A.BboxParams(format='pascal_voc', label_fields=['category_ids']),)
```

Slika 3.4. *Primjer koda složene augmentacije slike*

Složena augmentacija predstavlja primjenu više vrsta augmentacije na izvornu sliku. Složene augmentacije koje su korištene u ovom radu su:

- Rotacija od +10˚ (Rotate) i horizontalno zakretanje (HorizontalFlip),
- Rotacija od -10˚ i vertikalno zakretanje (VerticalFlip),
- Zamagljivanje (Blur) i vertikalno zakretanje,
- Zamagljivanje i horizontalno zakretanje,
- Smanjivanje svjetlosti i kontrasta (RandomBrightnessContrast) i horizontalno zakretanje te
- Povećavanje svjetlosti i kontrasta i vertikalno zakretanje.

Augmentacijom podatkovni skup je povećan na 200 slika čime je povećana njegova robusnost. Slika 3.5. prikazuje neke od provedenih augmentacija slika.



Slika 3.5. *Augmentacija slika*

4. DETEKCIJA VOĆA

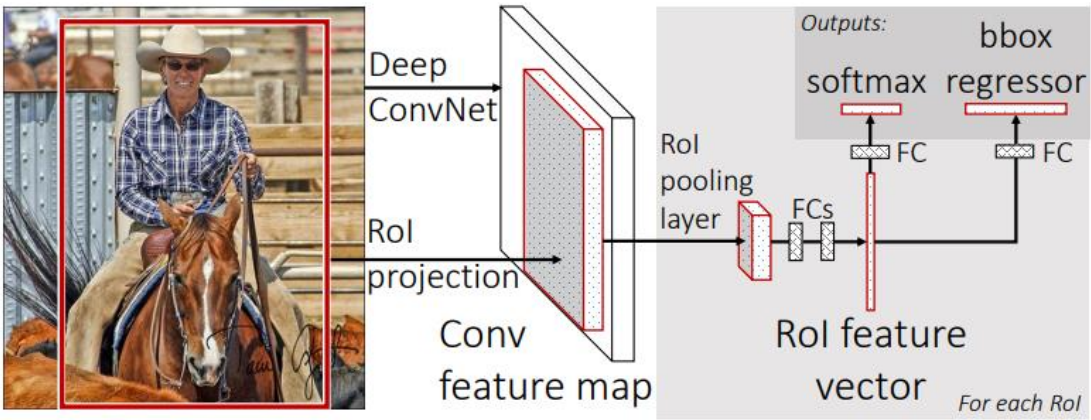
Jedan od čestih i važnih problema s kojim se suočava područje računalnog vida i umjetne inteligencije je detekcija voća. Kao što je i ranije rečeno, taj se problem najčešće rješava implementacijom neuronskih mreža. Iako većina radova koristi neuronske mreže za detekciju voća unutar voćnjaka ili plastenika, u ovom radu će oni biti testirani za sustav opisan u uvodnom poglavlju. Najčešće korišteni modeli neuronskih mreža za detekciju voća su Faster RCNN i YOLO te će oni biti opisani u ovom poglavlju. Faster RCNN spada u kategoriju dvofaznog (engl. *two-stage detector*), a YOLO jednofaznog (engl. *one-stage detector*) detektora. Dvofazna detekcija je metoda dobivanja rezultata sekvencijalnim izvođenjem klasifikacije (engl. *classification*) i prijedloga regije (engl. *regional proposal*) [28]. Prijedlog regije predstavlja isječak slike koji bi mogao predstavljati neki objekt. S druge strane jednofazna detekcija klasifikaciju i prijedlog regije radi istovremeno. Zbog svoje arhitekture dvofazne detekcije imaju manju brzinu detekcije i veću točnost, a jednofazne detekcije imaju veću brzinu detekcije i manju točnost [28]. U nastavku će biti dan opis Faster RCNN i YOLO mreže korišten u radu.

4.1. Faster RCNN

Faster RCNN mreža pripada kategoriji regijski temeljene konvolucijske neuronske mreže (RCNN) koju je razvio i opisao Ross Girshick i drugi [29] u radu iz 2014. godine. RCNN tada postaje trenutno dostignuće (engl. *state-of-the-art*) u području prepoznavanja i segmentaciji objekata. RCNN za ulaznu sliku kreira prijedloge regija koristeći selektivnu pretragu (engl. *selective search*). Selektivna pretraga iterativno segmentira sliku te svakom segmentu (regiji) slike dodaje granični okvir [30]. Segmenti (regije) se grupiraju na temelju sličnosti boja, teksture, oblika i veličina. Prijedlozi regija zatim služe kao ulazi u konvolucijsku neuronsku mrežu. Konvolucijska neuronska mreža izdvaja značajke (engl. *feature map*) koje stroj sa potpornim vektorima (engl. Support Vector Machines) koristi za klasifikaciju regije, a linearni regresijski model za podešavanje graničnih okvira. Nedostaci ovog modela su što zahtjeva puno memorijskog prostora, a treniranje modela i detekcija objekata na slici je spora [31].

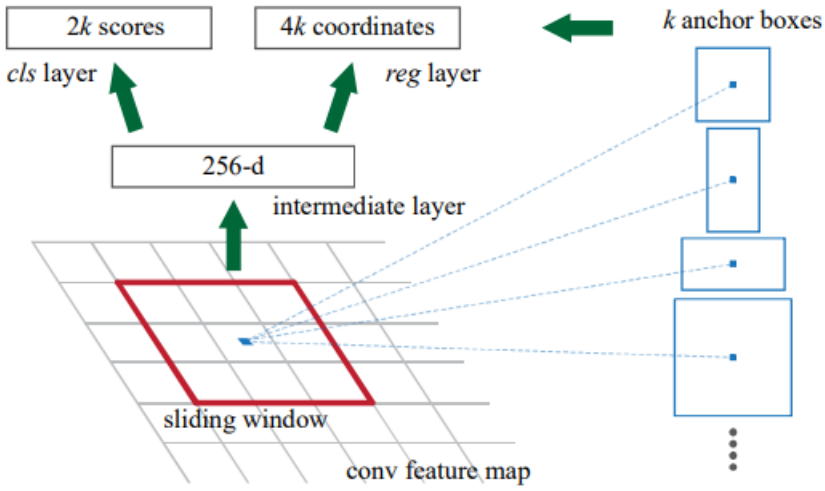
Pojavom Fast RCNN-a [32] smanjuje se potreba za memorijskim prostorom, a vrijeme treniranja modela i detekcije objekta se ubrzalo. Treniranje je jedno-stupanjsko (engl. *single stage*) te osvježava sve parametre mreže. Također je veća preciznost u odnosu na RCNN. U Fast RCNN-u izdvajanje značajki se vrši jednom pomoću konvolucijske neuronske mreže. U Fast RCNN-u ulaz u konvolucijsku neuronsku mrežu je cijela slika te prijedlozi regija, za razliku od RCNN-a

koji svaki prijedlog regije šalje na ulaz mreže. Svaki se prijedlog regije zatim projicira u područje od interesa (engl. *Region of Interest - RoI*) u izdvojenim značajkama. Nakon toga slijedi RoI sloj sažimanja koji za svako područje od interesa (prijedlog regije) stvara polje vektora značajki fiksne visine i širine te se na vektore značajki unutar polja primjenjuje max-sažimanje operacija (engl. *max-pooling*). Za svako polje dobiveno RoI slojem sažimanja, izračunava se vektor značajki pomoću potpuno povezanih slojeva. Iz tog vektora značajki ponovno nekoliko potpuno povezanih slojeva izračunava vrijednost pripadnosti objekta nekoj klasi te poboljšanje graničnog okvira. Za klasifikaciju prijedloga regije koristi se softmax metoda, a za podešavanje graničnog okvira regresijski model. Slika 4.1. prikazuje Fast RCNN arhitekturu.

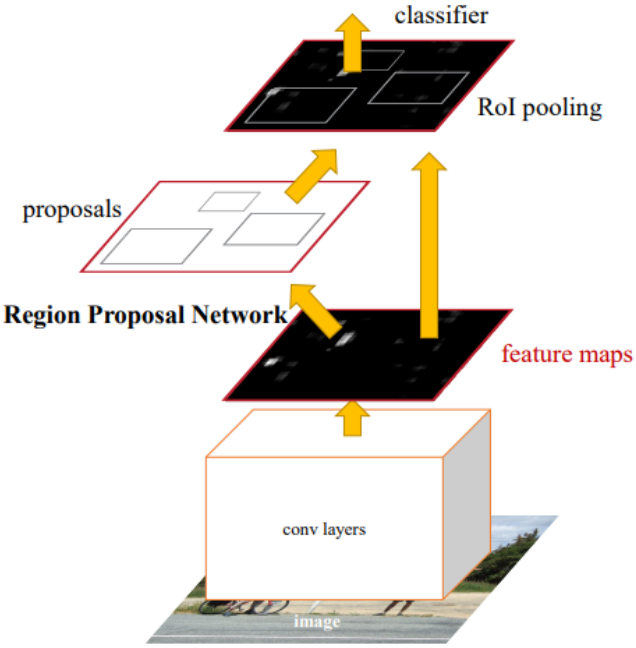


Slika 4.1. *Arhitektura Fast RCNN-a* [32]

Nedostatak Fast RCNN-a je što se puno vremena potroši na algoritam selektivne pretrage. Faster RCNN [33] rješava taj problem zamjenom selektivne pretrage mrežom koja prepoznaje područje od interesa (engl. *Region Proposal Network - RPN*). RPN dobije izdvojene značajke iz okosne konvolucijske neuronske mreže (engl. *backbone network*) po kojima se pomiče prozor dimenzija npr. 3 x 3 (Slika 4.2.) koji svakom vektoru značajki pridružuje niže dimenzionalan vektor značajki [33, 34]. Pomoću vektora značajki i potpuno povezanog sloja mjeri se vjerojatnost pronalaska objekta te se predlaže granični okviri (engl. *anchor boxses*). Na kraju se pomoću metode ne-maksimum potiskivanja (engl. *Non-Maximum Suppression*) dobije N najviše rangiranih prijedloga regija. Osim RPN-a, Faster RCNN se sastoji i od Fast RCNN-a. RPN prosljeđuje prijedloge regija Fast RCNN-u koji onda za svaki prijedlog određuje klasu i poboljšava granični okvir. Slika 4.3. prikazuje arhitekturu Faster RCNN-a.



Slika 4.2. *Region Proposal Network – RPN* [33]



Slika 4.3. *Arhitektura Faster RCNN-a* [33]

U ovom diplomskom radu testirati će se Faster RCNN mreža s različitim okosnim konvolucijskim neuronskim mrežama. Okosne neuronske mreže se odnose na mreže koje obrađuju ulazne podatke za izdvajanje značajki [35]. Postoji puno konvolucijskih neuronskih mreža koje se koriste kao okosnica, a u ovom radu će se koristi VGG16 i ResNet-50-FPN.

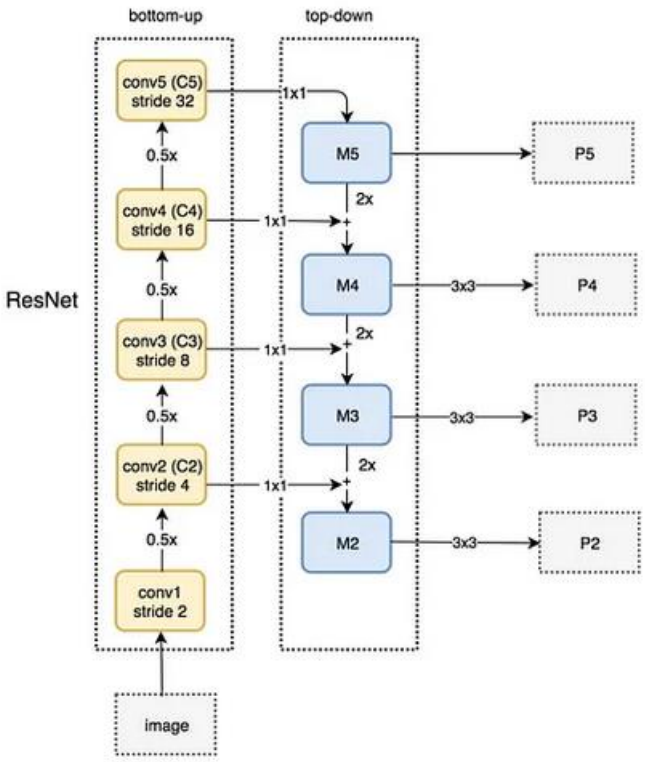
VGG16 mreža je predstavljena 2014. godine u radu Karena Simonyana i Andrewa Zissermana [36]. Tada je bila dostignuće u području mreža za izdvajanje značajki te se smatrala vrlo dubokom mrežom. Mreža se sastoji od 16 težinskih slojeva (Slika 4.4.). VGG16 sadrži 13 konvolucijskih slojeva, 5 max-sažimanja i 3 potpuno povezana sloja. Kao ulaz u mrežu uzima se

tenzor 224x224x3. VGG16 koristi 3 x 3 konvloucijski filter uz korak uzorkovanja (engl. *stride*) 1, ispunjenje rubova (engl. *padding*) mu je 1, a od sloja sažimanja filter ima dimenziju 2 x 2 uz korak uzorkovanja 2. Na krajevima mreže se nalaze 3 potpuno povezana sloja gdje prva 2 imaju 4096 kanala, a zadnji 1000. Poslije potpuno povezanih slojeva slijedi softmax metoda pri izlazu iz mreže. Nedostaci VGG16 mreže što je jako spora za treniranje te što zauzima jako puno memorijskog prostora i širinu propusnog pojasa (engl. *bandwidth*) [37].

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

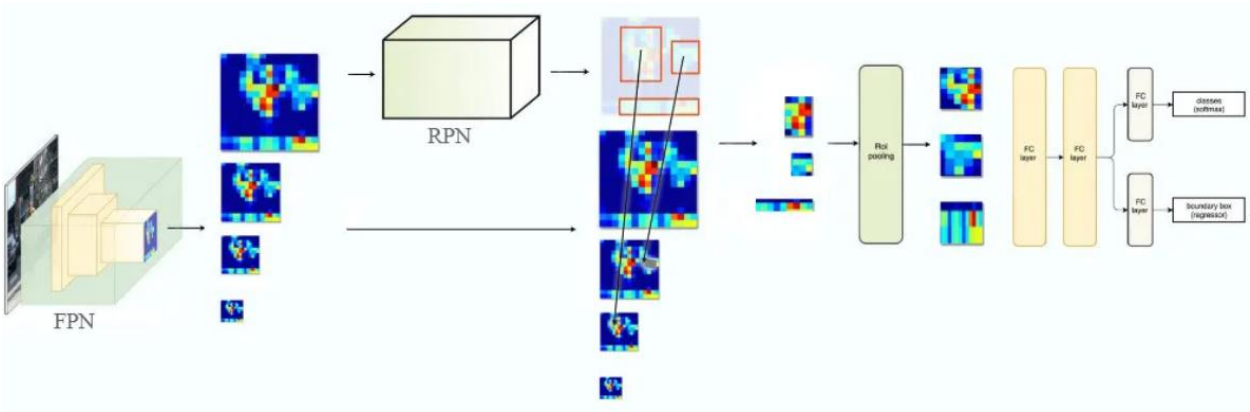
Slika 4.4. *Primjeri arhitektura VGG-a mreža* [36]

ResNet-50 [38] mreža se sastoji od 50 težinskih slojeva. Ona rješava problem dodavanja više konvolucijskih slojeva u mrežu, bez dobivanja problema nestajanja gradijenta (engl. *vanishing gradient problem*) [39]. Taj problem se riješio prečac vezom (engl. *shortcut connections*) koja preskače određene slojeve mreže. ResNet-50 prečac vezom preskače 3 sloja te koristi 1 x 1 konvolucijski sloj s kojim smanjuje broj parametara. Tako mreža postaje puno brža za treniranje. U ovom radu ResNet-50 se koristi s piramidalnom mrežom za značajke (engl. *Feature Pyramid Network - FPN*) [40]. FPN je mreža piramidalnog koncepta za izvlačenje značajki sa slika. Ona daje višestruki sloj izdvojenih značajki čime se povećava točnost i kvaliteta informacije. FPN se sastoji od uzlaznog (engl. *bottom-up*) i silaznog (engl. *top-down*) puta [41]. Uzlazni put koristi ResNet mrežu za izdvajanje značajki koja se sastoji od nekoliko konvolucijskih modula čiji se izlazi kasnije koriste pri silaznom putu (Slika 4.5.).



Slika 4.5. Izgled piramidalne mreže za značajke [41]

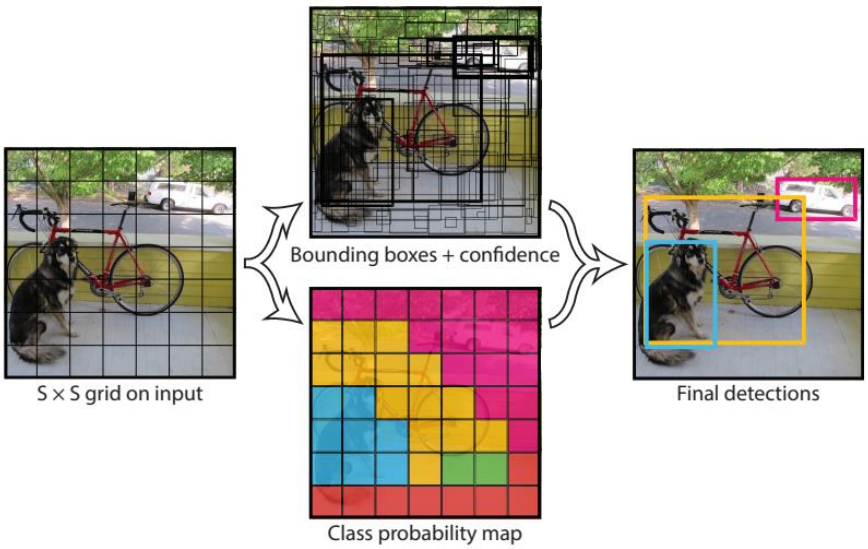
Nakon svakog konvolucijskog modula prostorna dimenzija se upola smanji. Na zadnjem konvolucijskom modulu primjenjuje se konvolucijski filter dimenzija 1 x 1 kako bi se smanjila dubina kanala prvih izdvojenih značajki u silaznom putu. Krećući se silaznim putem, svaka se izdvojena značajka dvostruko uzorkuje (pomoću algoritma najbližih susjeda) te se spaja s odgovarajućim filtriranim izlazom konvolucijskog modula. Na novonastalim izdvojenim značajkama se primjenjuje konvolucijski filter (dimenzija 3 x 3) radi uklanjanja alias efekta. Za prvi konvolucijski modul se ne koriste ranije opisane operacije. Slika 4.6. prikazuje izgled Faster RCNN mreže koristeći ResNet-50-FPN okosnicu.



Slika 4.6. Faster RCNN s ResNet-50-FPN okosnicom [41]

4.2. YOLOv4

YOLO (engl. *You Only Look Once*) algoritam (kao što mu i samo ime govori) pripada u kategoriju jednofaznih detektora. Svojom pojavom u radu iz 2016. godine [13] (radi svoje brzine) postaje trenutno dostignuće u području detekcije objekata u stvarnom vremenu. Osnovni princip YOLO algoritma je jednostavan. Jedna konvolucijska mreža istovremeno radi predikciju koordinata graničnog okvira i klasifikaciju za te okvire [13]. Prvo se slika podjeli u mrežu s N ćelija gdje svaka ćelija ima istu dimenziju. U svakoj ćeliji se vrši predikcija graničnih okvira, vjerovatnost pojavljivanja objekta unutar graničnih okvira i vjerovatnost pripadanja objekta određenoj klasi. Na kraju se primjenjuje ne-maksimum potiskivanje gdje se prvo provjerava jesu li preklapajući granični okviri iste klase. Ako su granični okviri iste klase, računa se omjer presjeka i unije nakon čega se izbacuju oni s malim omjerom.

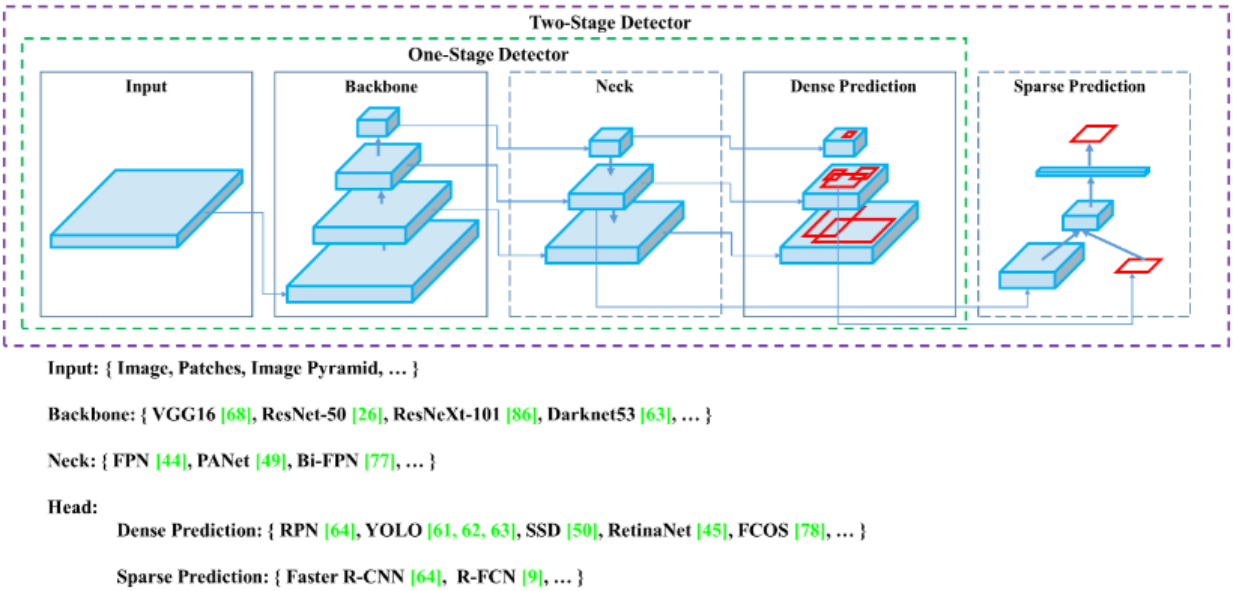


Slika 4.7. *Model YOLO algoritma* [13]

S godinama YOLO algoritam je dobivao svoje inačice. 2016. godine izlazi YOLOv2 [42] koji uvodi „anchor-box“ regije za predikciju čime je omogućeno predviđanje više graničnih okvira za svaku ćeliju [43]. S ovom opcijom poboljšano je detektiranje malih objekata. Nakon YOLOv2 dolazi verzija YOLOv3 [44], koja umjesto DarkNet-19 koristi Darknet-53 okosnicu te radi predikciju graničnih okvira u tri različita omjera. YOLOv4 [45] se pojavljuje 2020. godine te će se zbog svoje učestale primjene u detekciji voća koristiti i biti opisan u daljnjem dijelu rada.

Arhitektura YOLOv4 mreže se sastoji od nekoliko cjelina. Nakon što slika uđe u YOLOv4 arhitekturu, ulazi u cjelinu koju čini okosna neuronska mreža koja zajedno sa sljedećom cjelinom,

vratom arhitekture (engl. *neck*), izdvaja značajke sa slike. Na kraju se nalazi glava arhitekture. Inače, glavu arhitekture čine jednofazne ili dvofazne mreže koje služe za predikciju (Slika 4.8.). YOLOv4 sačinjavaju: CSPDarknet53 mreža kao okosnica, prostorno piramidalno sažimanje (engl. *Spatial Pyramid Pooling - SPP*) s putnom agregacijskom mrežom (engl. *Path Aggregation Network - PANet*) čini vrat mreže, a glavu čini YOLOv3 koji se koristi za predikciju.



Slika 4.8. *Općeniti prikaz cjelina mreža za prepoznavanje objekata* [45]

CSPDarknet53 je baziran na DenseNet mreži. DenseNet [46] mreža se sastoji od nekoliko potpuno povezanih slojeva podijeljenih u blokove te prijelaznog sloja. Prijelazni sloj ima funkciju da izlaz iz i-tog bloka bude spojen s ulazom i-tog bloka te tako spojeni ishod postaje ulaz (i + 1)-tog bloka [47]. Kod CSP arhitekture (engl. *Cross Stage Partial architecture*) i-ti ulaz se dijeli na dva dijela. Prvi ulaz prolazi kroz blok potpuno povezanih slojeva, a drugi ulaz taj blok preskače te se spaja s rezultatom prvog ulaza nakon njegovog prolaska kroz blok. Vrat mreže sakuplja slike izdvojenih značajki te ih obrađuje. Vrat se sastoji od dva dijela: prostornog piramidalnog sažimanja i putne agregacijske mreže. Prostorno piramidalno sažimanje koristi slojeve sažimanja kako bi, bez obzira na dimenzije izdvojenih značajki, generirala značajke fiksne veličine. Putna agregacijska mreža poboljšava proces segmentacije instance čuvanjem prostornih informacija koje pomažu u pravilnoj lokalizaciji piksela za predviđanje maske [48]. Ova modificirana mreža ulančava susjedne slojeve umjesto da ih dodaje kod sloja sažimanja adaptivnih značajki (engl. *adaptive feature pooling*). Glava mreže služi za konačnu predikciju objekata, koju u ovom slučaju čini YOLOv3 mreža. U YOLOv4 radu dodane su i opisane neke nove značajke nazvane „Bag of freebies“ i „Bag of specials“. „Bag of freebies“ se najviše odnose na načine augmentacije podataka ne povećavajući

trošak zaključivanja (engl. *inference cost*). Neke metode koje spadaju u „Bag of freebies“ su: mozaik i „Cut-Mix“ augmentacija podataka, IoU gubitak, „DropBlock“ regularizacija itd. „Bag of specials“ samo malo povećavaju trošak zaključivanja, ali značajno povećavaju točnost predviđanja objekata [45]. U ovom radu će se osim YOLOv4 mreže testirati i YOLOv4-tiny verzija mreže. YOLOv4-tiny je manja verzija YOLOv4 mreže te zbog svoje arhitekture ima veću brzinu detekcije. YOLOv4-tiny ima smanjen broj konvolucijskih slojeva, ima dva YOLO sloja (YOLOv4 ima tri) te je manji broj graničnih okvira za predikciju [49].

5. PROCJENA VELIČINE I BROJANJE VOĆA

Veličina voća jako je bitan faktor pri kontroli kvalitete voća. U ovom poglavlju je opisana implementirana metoda za procjenu veličine voća. Metoda koja se koristila je segmentacija slike pomoću boje. Nakon detekcije voća, voće je potrebno prebrojati. Znanje o količini voća je bitno kod skladištenja i inventure robe. Za prebrojavanje voća u kašeti isprobana je Houghova detekcija krugova.

5.1. Segmentacija slike pomoću boje

Segmentacija slike je operacija podjele piksela slike na disjunktne podskupove, pri čemu jedan podskup predstavlja objekt od interesa na slici, a ostali predstavljaju pozadinu odnosno druge objekte na sceni [50]. Objekte možemo segmentirati po boji, geometrijskim oblicima i sl. U našem slučaju zelene jabuke sorte Granny Smith se znatno razlikuju od sive kartonske podloge pa je segmentacija po boji dobar izbor. RGB sliku (gdje R predstavlja crvenu, G zelenu i B plavu komponentu slike) je moguće transformirati u binarnu sliku. Binarna slika se dobije sljedećom formulom [50]:

$$c_m = \begin{cases} 1, \ ako\ je\ r_{min} \leq r_m \leq r_{max} \wedge g_{min} \leq g_m \leq g_{max} \wedge b_{min} \leq b_m \leq b_{max} \\ 0, \ inače \end{cases} \tag{5-1}$$

gdje su

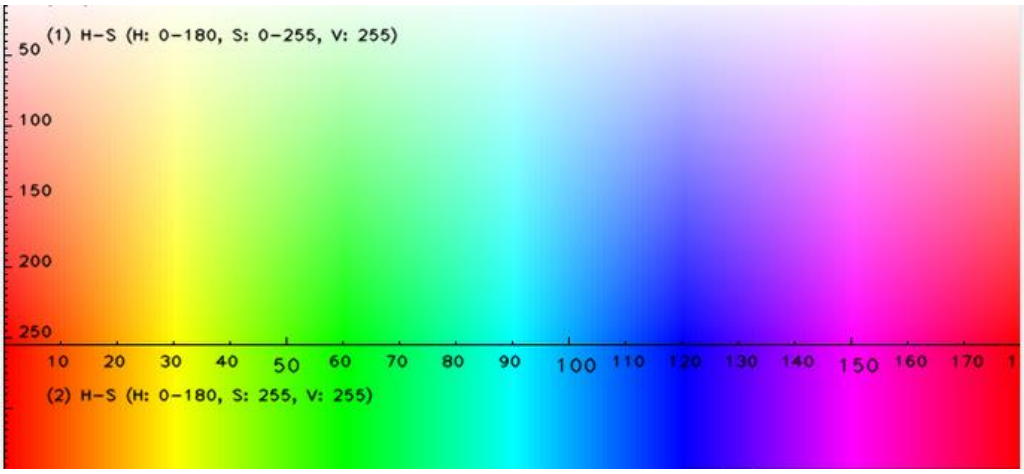
c_m - piksel m binarne slike,

r_m, g_m, b_m - crvena, zelena i plava komponenta boje piksela m,

r_{min}, r_{max} - donja i gornja granica intervala intenziteta crvene boje objekta od interesa,

g_{min}, g_{max} - donja i gornja granica intervala intenziteta zelene boje objekta od interesa,

b_{min}, b_{max} - donja i gornja granica intervala intenziteta plave boje objekta od interesa.



Slika 5.1. *HSV prostor boja* [51]

Osim RGB prostora boja postoji i HSV prostor boja (Slika 5.1.). U HSV prostoru boja, H element predstavlja boju u frekvencijskom vidljivom spektru, S element predstavlja zasićenje neke boje (što je veća to je intenzivnija, što je manja to je sivlja) i V element predstavlja svjetlinu neke boje. Odabrana boja u HSV 3D prostoru bit će bolje lokalizirana i vizualno odvojiva [52]. Korištenjem određenih programskih biblioteka realizirat će se segmentiranje jabuka pomoću HSV prostora boja. Segmentiranje se primjenjuje na cijelu sliku dobivenu kamerom. Kada se segmentacijom dobije binarna slika moći će se odrediti veličina detektiranih područja odnosno objekata. Područje detektiranog objekta bit će izraženo u pikselima, ali primjenom „piksel po metrici“ kalibracije (engl. *pixels per metric*) [53] veličina objekta će biti izražena u milimetrima. Time će se riješiti problem određivanja veličina jabuka u kutiji.

5.2. Houghova detekcija krugova

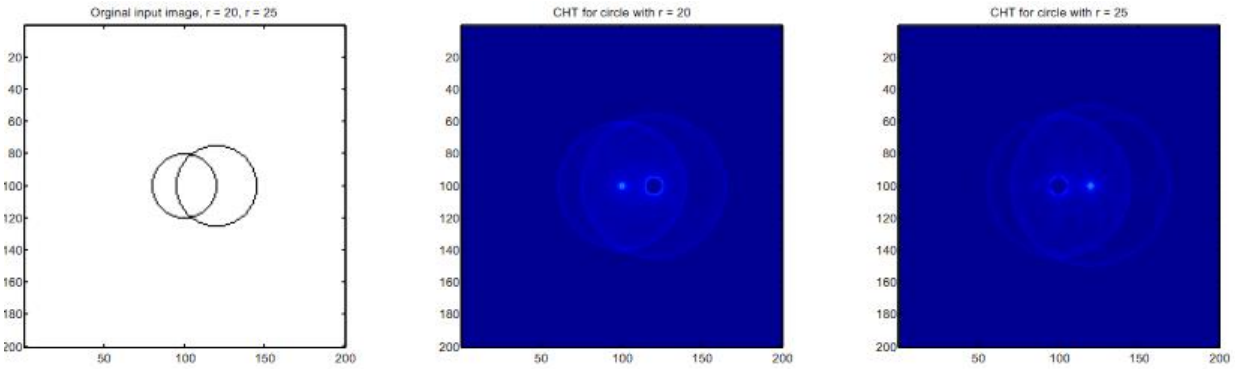
Često se u računalnom vidu nailazi na problem pronalaska raznih oblika na slici. Problem pronalaska linija može se riješiti Houghovom transformacijom. Houghova transformacija može se opisati kao transformacija točke iz x-y ravnine u prostor parametara [21]. Umjesto Kartezijevim koordinatnim sustavom, linija je prikazana njezinom normalom odnosno kutom (θ) i dužinom (ρ):

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{5-2}$$

Često je umjesto linija na slikama potrebno prepoznati krugove. Krug je jednostavnije prikazati u prostoru parametara (jer mu se parametri direktno prenose u prostor parametara) s radijusom (r) i centrom (a, b):

$$x = a + r\cos(\theta) \tag{5-3}$$
$$y = b + r\sin(\theta)$$

Houghova transformacija za krugove je kompleksnija jer prostor parametara za krug spada u R^3. Houghova transformacija za krugove započinje pronalaskom svih rubova na slici. To se obradi nekim algoritmom za pronalazak rubova npr. Canny algoritmom. Nakon toga na svakoj točki ruba (koja će predstavljati središte kruga) će se nacrtati krug radijusa r (Slika 5.2.). Pri tome se inkrementiraju vrijednosti u akumulacijskoj matrici. Najveće vrijednosti u akumulacijskoj matrici predstavljaju središta krugova na slici. Pronalaze se varijable a, b i r za prikaz krugova.



(a) The input image. (b) Hough transform where r = 20. (c) Hough transform where r = 25.

Slika 5.2. *Primjer Houghove transformacije za krugove* [21]

6. REZULTATI

Programski jezik koji se koristio za izradu diplomskog rada je Python [54]. Python je besplatni programski jezik, jednostavne sintakse s velikom korisničkom zajednicom. Osim toga, Python ima veliki izbor kvalitetnih biblioteka za digitalnu obradu slike, znanost o podacima i strojno učenje potrebnih za izradu diplomskog rada. Neke od biblioteka korištenih u ovom radu su: Numpy (biblioteka za matematičke zadatke), Matplotlib (biblioteka za vizualizaciju), OpenCV (biblioteka za obradu slike), Torch (biblioteka za strojno učenje)… Za treniranje neuronskih mreža korišten je Google Colab razvojno okruženje [55]. Google Colab razvojno okruženje je usluga u oblaku koja nudi pisanje i izvršavanje koda u pregledniku. Namjenjen je za rješavanje zadataka iz strojnog učenja (neuronske mreže). Nudi besplatni pristup računalnim resursima poput grafičkih kartica (Tesla T4). Pri treniranju podatkovni skup je podijeljen na trening, validacijski i test skup u omjeru 70:20:10. Nakon treniranja, modeli su evaluirani uz sljedeće metrike:

TP (Stvarno pozitivni) – klasa 1 je predviđena i točno je predviđanje.
TN (Stvarno negativni) – klasa -1 je predviđena i točno je predviđanje.
FP (Lažno pozitivni) – klasa 1 je predviđena i netočno je predviđanje.
FN (Lažno negativni) – klasa -1 je predviđena i netočno je predviđanje.

1. Točnost (engl. *accuracy*) – pokazuje broj ispravno klasificiranih uzoraka.

$$A = \frac{TP+TN}{TP+FP+TN+FN} \tag{6-1}$$

2. Preciznost (engl. *precision*) – broj uzoraka koji pripadaju klasi 1 od svih za koje je model predvidio da su klasa 1.

$$P = \frac{TP}{TP+FP} \tag{6-2}$$

3. Odziv (engl. *recall*) – pokazuje točno predviđanje klase 1 kada je klasa 1.

$$R = \frac{TP}{TP+FN} \tag{6-3}$$

4. F1-mjera (engl. *F1-score*) – harmonijska sredina preciznosti i odziva.

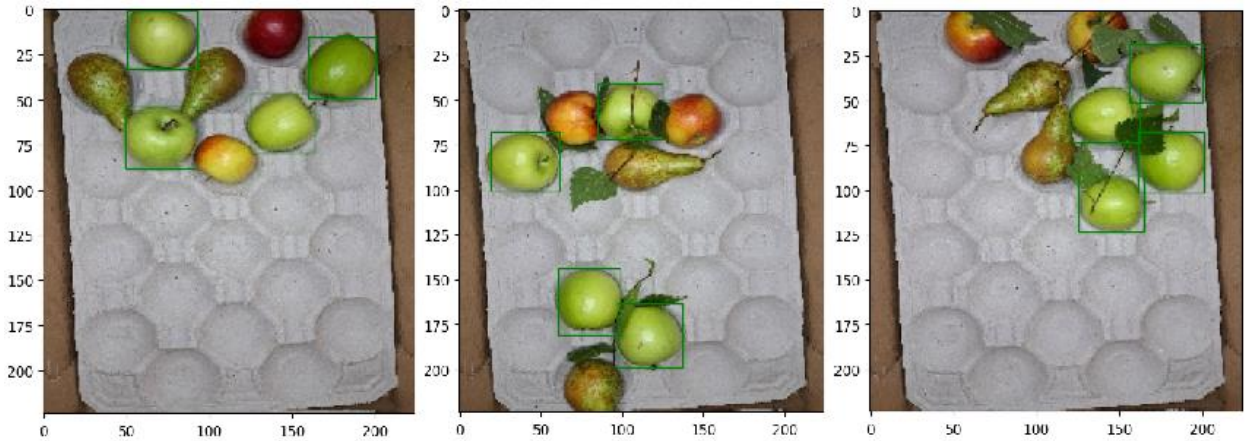
$$F = \frac{2PR}{P+R} \tag{6-4}$$

U tablici 6.1. prikazani su rezultati evaluiranih neuronskih mreža. Kod koji se koristio za treniranje mreža je [56] i [57].

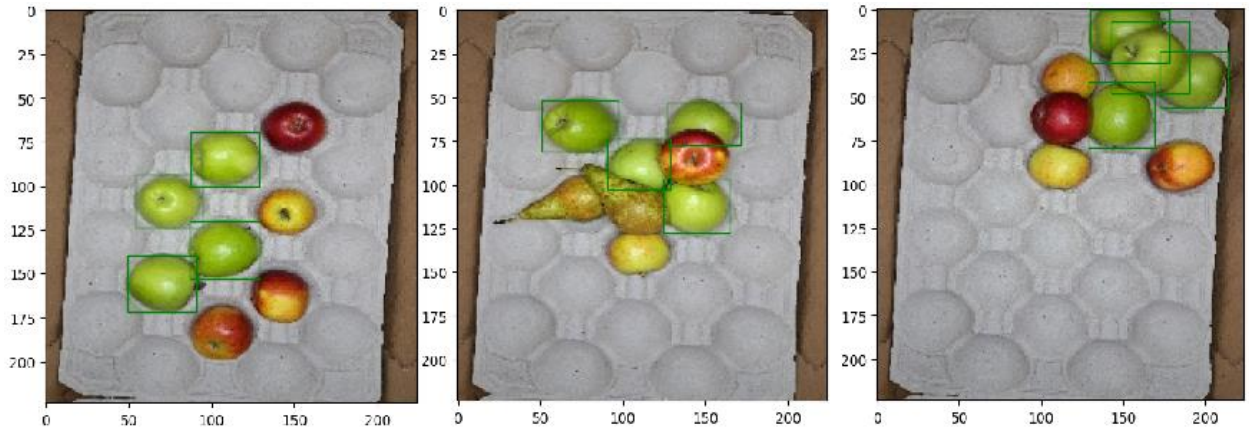
Tab. 6.1. *Evaluirani rezultati neuronskih mreža*

Mreža	Arhitektura	Točnost	Srednja točnost	Preciznost	Odziv	F1-mjera
Faster RCNN	VGG16	0,9873	0,9873	0,9753	0,9875	0,9814
	ResNet-50-fpn	1	1	0,9877	1	0,9938
YOLO	v4	1	1	1	1	1
	v4-tiny	1	1	0,98	1	0,99

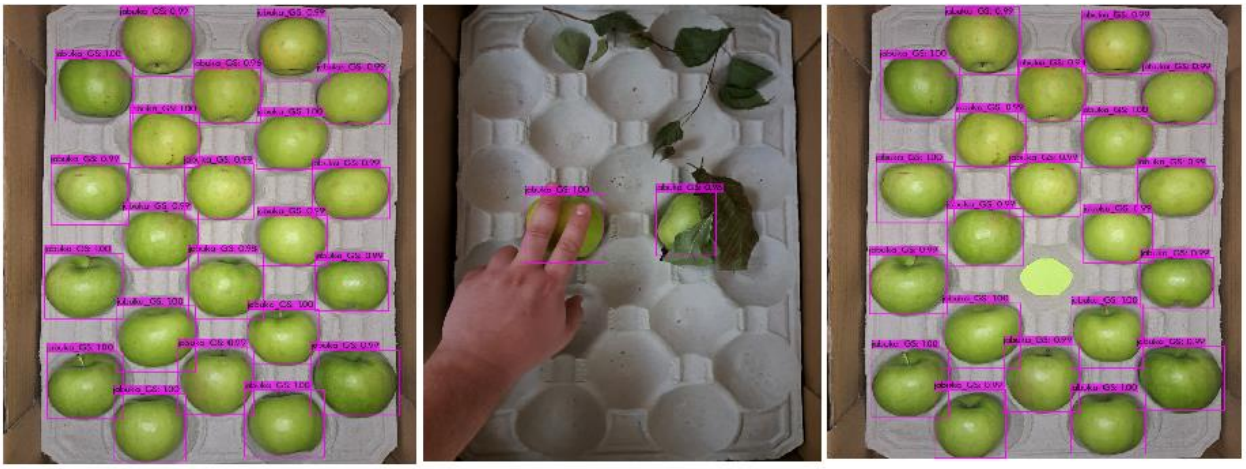
Rezultati istreniranih neuronskih mreža pokazuju izvrsne rezultate. Faster RCNN mreža s VGG16 okosnicom daje nešto lošije rezultate u odnosu s ResNet-50-fpn okosnicom. To je razumljivo jer je ResNet-50-fpn mreža dublja od VGG16 mreže. Što se tiče YOLO mreža, obje arhitekture daju vrhunske rezultate. YOLOv4-tiny je reducirana verzija YOLOv4 mreže (manje arhitekture) pa daje nešto lošiju preciznost. S obzirom da ove mreže u voćnjacima i plastenicima, gdje su scene za predikciju voća nešto složenije, daju vrhunske rezultate (Poglavlje 2), rezultati dobiveni u Tab. 6.1. su opravdani. Slike 6.1., 6.2., 6.3., i 6.4. prikazuju rezultate predikcije za pojedine mreže.



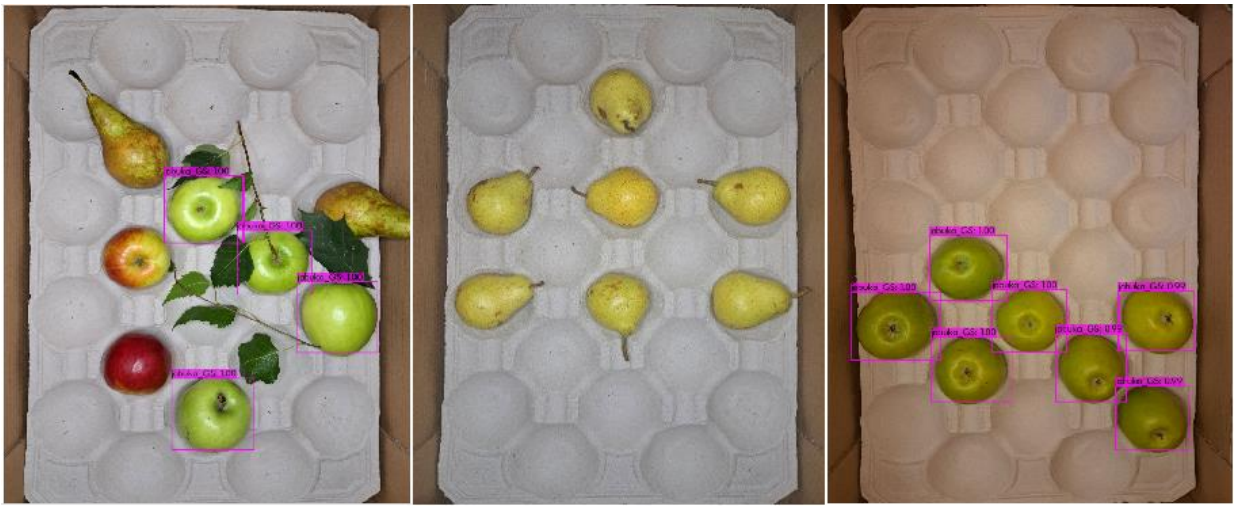
Slika 6.1. *Rezultati – Faster RCNN (VGG16)*



Slika 6.2. *Rezultati – Faster RCNN (ResNet-50-fpn)*



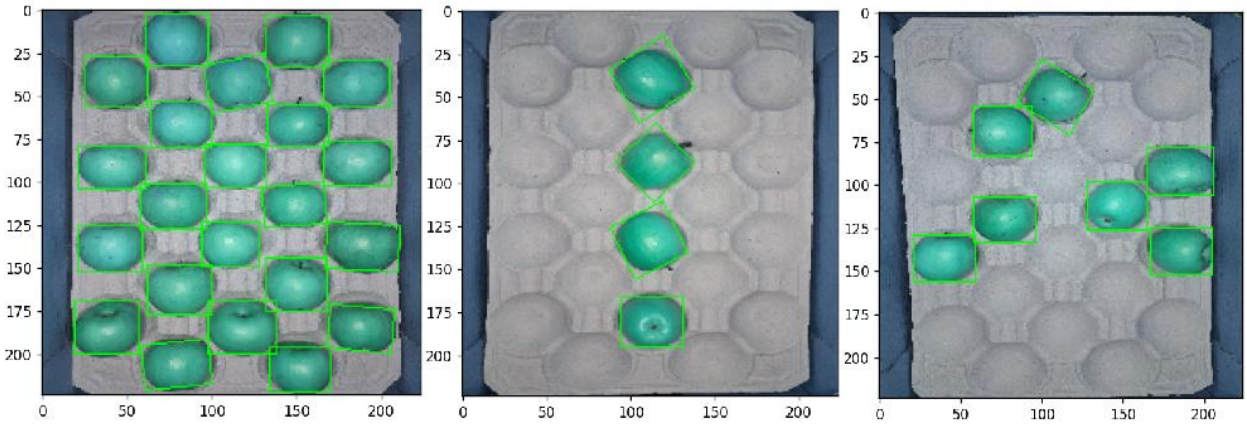
Slika 6.3. *Rezultati – YOLOv4*



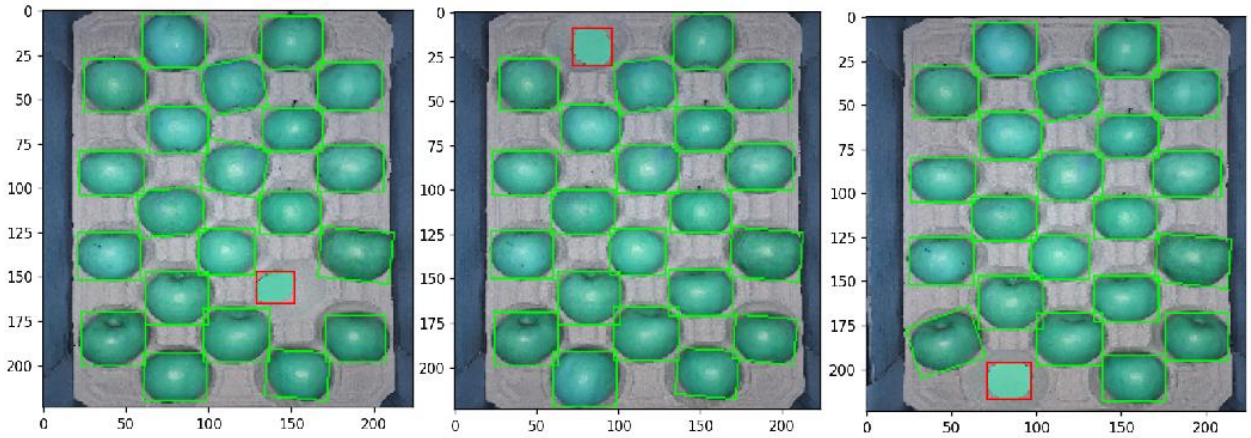
Slika 6.4. *Rezultati – YOLOv4-tiny*

Segmentacija slike pomoću boje korištena je za određivanje veličine jabuka. Prema [1], za sorte jabuka svih klasa minimalna veličina određena promjerom iznosi 60 mm. Za mjerenje veličine jabuka korištena je OpenCV biblioteka. Segmentiranjem slike po zelenoj boji dobije se binarna slika gdje se računaju dimenzije dobivenih kontura. Pomoću OpenCV funkcije *minAreaRect()* dobivaju se dimenzije dobivenih kontura koje ne smiju prelaziti dimenziju 60x60 mm. Pošto su dimenzije koje daje funkcija u pikselima određeno je koliko 60 mm iznosi piksela „piksel po metrici“ kalibracijom [53]. Ako je kontura manja od 60x60 mm, bit će iscrtana crvenim (Slika 6.6.), a u suprotnom zelenim okvirom (Slika 6.5.). Osim što algoritam procjenjuje veličinu jabuka, može i prepoznati jesu li jabuke dobro posložene. Ako jabuke nisu dobro posložene (jedna na drugoj, između dvije jabuke i sl.), segmentiranjem slike dobit će se kontura velikih dimenzija koja se također iscrtava crvenim okvirom (Slika 6.7.). Problem segmentiranja slike po boji je taj što može doći predmet iste boje kao i predmet od interesa, ali taj problem se može riješiti

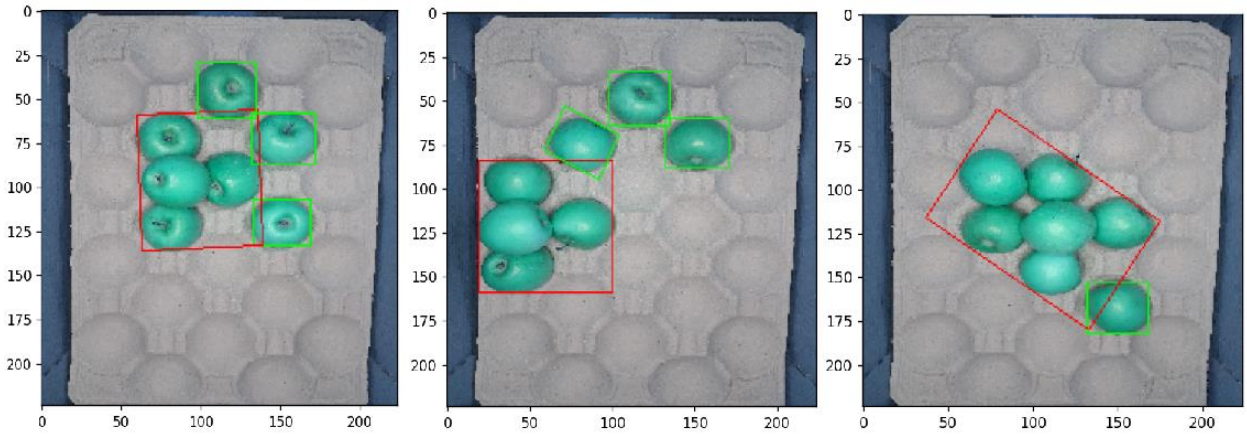
uspoređivanjem broja predikcija koje daje neuronska mreža s brojem dobivenih kontura nastale segmentacijom te tako uvidjeti nepravilnost. Omjer presjeka i unije dobivenih graničnih okvira segmentacijom i anotiranih graničnih okvira iznosi 0,8391 što je vrlo dobro.



Slika 6.5. *Predikcija kada su sve jabuke veće od 60x60 mm*

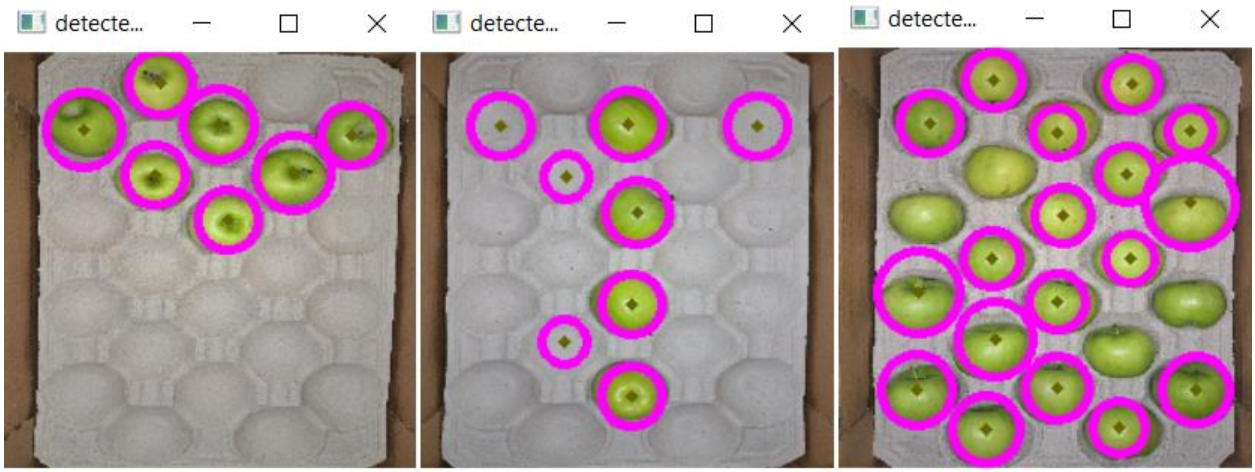


Slika 6.6. *Predikcija kada nisu sve jabuke veće od 60x60 mm*

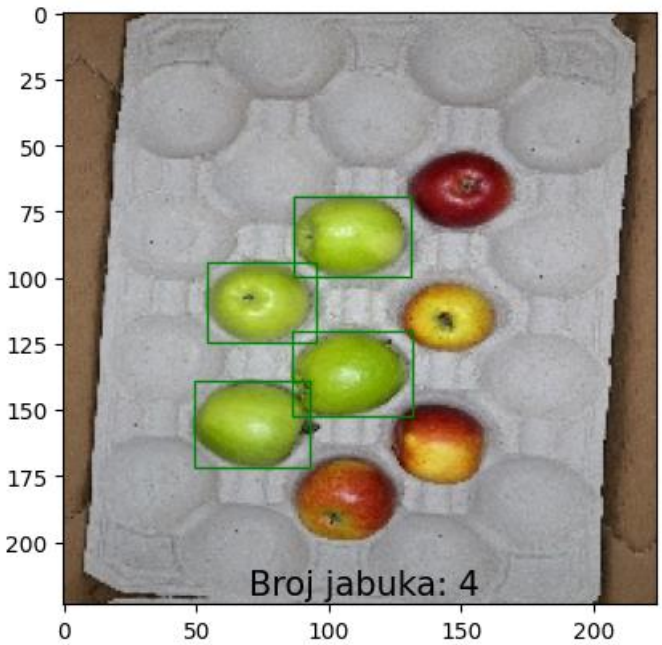


Slika 6.7. *Predikcija kada jabuke nisu dobro posložene*

Houhova detekcija krugova je testirana za prebrojavanje jabuka te se pokazala kao loš odabir za ovaj zadatak. Na Slici 6.8. vidi se kako se za iste parametre transformacije dobiju različiti rezultati za relativno istu scenu. Kako navodi [8], jedan od glavnih nedostataka Houghove transformacije za krugove su oslanjanje na točne segmentacije, nemogućnost rukovanja okluzijama te potreba za finim podešavanjem parametara. [8] koristi i navodi kako se često sumiraju detektirani objekti (izlazi) iz neuronske mreže kao način prebrojavanja voća. Kako u ovom radu neuronske mreže daju vrhunske rezultate, sumirane detekcije jabuka su dobar izbor za prebrojavanje jabuka. Sumirane detekcije jabuka, ovisno o neuronskoj mreži, imale bi iste evaluacijske metrike kao i neuronska mreža koja se koristi za predviđanje jabuka. Na Slici 6.9. su se zbrojile detekcije jabuka (izlazi) te se zbroj ispisao na dnu slike.



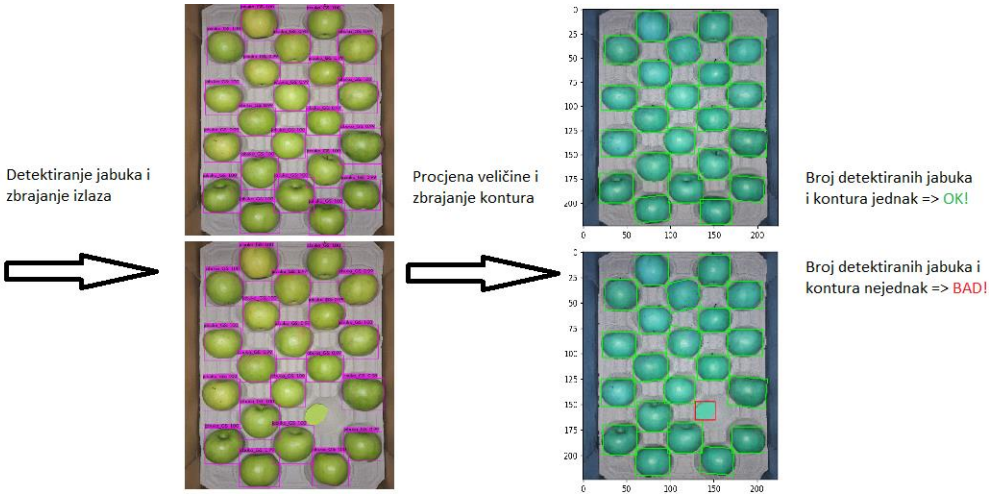
Slika 6.8. *Houghova detekcija krugova*



Slika 6.9. *Sumiranje prepoznatih jabuka (objekata)*

7. ZAKLJUČAK

U ovom diplomskom radu cilj je bio testirati nekoliko algoritama računalnog vida i umjetnih neuronskih mreža koje bi se mogle primijeniti pri prepoznavanju voća u kartonskim kutijama. Voće koje se prepoznavalo su jabuke sorte Granny Smith. Izrađena je maketa s kojom su se prikupljale slike za podatkovni skup. Nakon prikupljanja, slike su anotirane i augmentirane. Izrađeni podatkovni skup se pokazao pouzdanim za treniranje umjetnih neuronskih mreža. Umjetne neuronske mreže koje su se koristile imale su Faster RCNN i YOLO arhitekturu. Sve mreže su dale izvrsne rezultate. Iako Faster RCNN arhitektura s VGG16 i ResNet-50-fpn okosnicom daje vrhunske evaluacijske metrike, YOLOv4 i YOLOv4-tiny umjetne neuronske mreže imaju bolja vremena predikcije te su prigodne za korištenje u industrijskim postrojenjima gdje je brzina kontrole kvalitete jako bitna. Za procjenu veličine jabuka korištena je segmentacija slike po boji. Segmentacijom slike po zelenoj boji, uz pomoću OpenCV biblioteke, mogu se izračunati dimenzije kontura na dobivenoj binarnoj slici. Ako je dimenzija konture manja od zadane vrijednosti, jabuka ne zadovoljava veličinu, a ako je kontura veća to može značiti da jabuke nisu dobro posložene. Uspoređivanjem broja predikcija koje daje neuronska mreža s brojem dobivenih kontura nastale segmentacijom može se riješiti problem ako se u kutiji nalazi predmet iste boje kao i jabuka. Houghova transformacija za krugove nije zadovoljila zadatak jer je za različite slike potrebno podešavati nove parametre transformacije. Kao bolji odabir pokazalo se zbrojiti detektirane jabuke (izlaze) iz umjetnih neuronskih mreža jer mreže daju odlične evaluacijske metrike. Kao najbolji odabir za detektiranje jabuka pokazalo se koristiti YOLOv4 (ili YOLOv4-tiny, ako hoćemo još brže vrijeme predikcije) arhitekturu neuronske mreže, za procjenu veličine jabuka segmentaciju po boji te za prebrojavanje jabuka zbrojiti detektirane jabuke (izlaze) koje daje neuronska mreža (Slika 7.1.).



Slika 7.1. *Primjer sustava kontrole kvalitete jabuka*

LITERATURA

[1] Narodne novine: Pravilnik o kakvoći voća, NN 114/2008, dostupno na: https://narodne-novine.nn.hr/clanci/sluzbeni/2008_10_114_3322.html (pristupljeno: 15.8.2023.)

[2] Amazon: Machine Learning: Using algorithms to sort fruit, 30.5.2018., dostupno na: https://www.aboutamazon.eu/news/innovation/machine-learning-using-algorithms-to-sort-fruit (pristupljeno: 14.7.2023.)

[3] Kaggle: Fruit Images for Object Detection, Kaggle, dostupno na: https://www.kaggle.com/datasets/mbkinaci/fruit-images-for-object-detection (pristupljeno: 1.8.2023.)

[4] Kaggle: Fruit Detection, Kaggle, dostupno na: https://www.kaggle.com/datasets/andrewmvd/fruit-detection (pristupljeno: 5.8.2023.)

[5] Bargoti, Suchet and Underwood, James, Deep Fruit Detection in Orchards, ACFR Orchard Fruit Dataset, 2016., dostupno na: https://data.acfr.usyd.edu.au/ag/treecrops/2016-multifruit/ (pristupljeno: 2.8.2023.)

[6] Nicolai Häni, Pravakar Roy, Volkan Isler, MinneApple: A Benchmark Dataset for Apple Detection and Segmentation, Papers with code, 2020., dostupno na: https://paperswithcode.com/paper/minneapple-a-benchmark-dataset-for-apple (pristupljeno: 1.8.2023.)

[7] Parico, A.I.B.; Ahamed, T. Real Time Pear Fruit Detection and Counting Using YOLOv4 Models and Deep SORT. Sensors 2021, 21, 4803., dostupno na: https://doi.org/10.3390/s21144803 (pristupljeno: 15.7.2023.)

[8] Nicolai Häni, Pravakar Roy, Volkan Isler, A Comparative Study of Fruit Detection and Counting Methods for Yield Mapping in Apple Orchards, 6.3.2019., dostupno na: https://doi.org/10.48550/arXiv.1810.09499 (pristupljeno: 11.7.2023.)

[9] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 6.1.2016., dostupno na: https://doi.org/10.48550/arXiv.1506.01497 (pristupljeno: 15.7.2023.)

[10] Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. DeepFruits: A Fruit Detection System Using Deep Neural Networks. Sensors 2016, 16, 1222., dostupno na: https://doi.org/10.3390/s16081222 (pristupljeno: 10.7.2023.)

[11] Suchet Bargoti, James Underwood, Deep Fruit Detection in Orchards, 18.9.2017., dostupno na: https://doi.org/10.48550/arXiv.1610.03677 (pristupljeno: 28.6.2023.)

[12] Assunção, E.T.; Gaspar, P.D.; Mesquita, R.J.M.; Simões, M.P.; Ramos, A.; Proença, H.; Inacio, P.R.M. Peaches Detection Using a Deep Learning Technique—A Contribution to Yield Estimation, Resources Management, and Circular Economy. Climate 2022, 10, 11. dostupno na: https://doi.org/10.3390/cli10020011 (pristupljeno: 1.8.2023.)

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection, 9.5.2016., dostupno na: https://doi.org/10.48550/arXiv.1506.02640 (pristupljeno: 25.6.2023.)

[14] Fu, Lanhui & Duan, Jieli & Zou, Xiangjun & Lin, Jiaquan & Zhao, Lei & Li, Jinhui & Yang, Zhou. (2020). Fast and Accurate Detection of Banana Fruits in Complex Background Orchards. IEEE Access. 8. 196835-196846. 10.1109/ACCESS.2020.3029215, dostupno na: https://www.researchgate.net/publication/347127242_Fast_and_Accurate_Detection_of_Banana_Fruits_in_Complex_Background_Orchards (pristupljeno: 27.6.2023.)

[15] Guojin Li, Xiaojie Huang, Jiaoyan Ai, Zeren Yi, Wei Xie, Lemon-YOLO: An efficient object detection method for lemons in the natural environment, 8.3.2021., dostupno na: https://doi.org/10.1049/ipr2.12171 (pristupljeno: 10.7.2023.)

[16] Koirala, A., Walsh, K.B., Wang, Z. et al. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO'. Precision Agric 20, 1107–1135 (2019), dostupno na: https://doi.org/10.1007/s11119-019-09642-0 (pristupljeno: 27.7.2023.)

[17] Lawal, M.O. Tomato detection based on modified YOLOv3 framework. Sci Rep 11, 1447 (2021), dostupno na: https://doi.org/10.1038/s41598-021-81216-5 (pristupljeno: 2.8.2023.)

[18] Robert G. de Luna, Elmer P. Dadios, Argel A. Bandala, Ryan Rhay P. Vicerra, Size Classification of Tomato Fruit Using Thresholding, Machine Learning, and Deep Learning Techniques, listopad 2019., dostupno na: https://agrivita.ub.ac.id/index.php/agrivita/article/view/2435 (pristupljeno: 27.7.2023.)

[19] Nandi, Chandra & Tudu, Bipan & Koley, Chiranjib. (2013). Machine Vision Based Techniques for Automatic Mango Fruit Sorting and Grading Based on Maturity Level and Size. Sensing Technology: Current Status and Future Trends II. 8. 27-46, dostupno na: https://www.researchgate.net/publication/303002510_Machine_Vision_Based_Techniques_for_Automatic_Mango_Fruit_Sorting_and_Grading_Based_on_Maturity_Level_and_Size (pristupljeno: 5.8.2023.)

[20] Tech(B)log of Dr. Shrishail Gajbhar, [OpenCV] Detecting and Counting Apples in Real World Images using OpenCV and Python, 7.2.2021., dostupno na: https://shrishailsgajbhar.github.io/post/OpenCV-Apple-detection-counting (pristupljeno: 20.7.2023.)

[21] Pedersen, S. J. K. (2007). Circular Hough Transform. Aalborg University, Vision, Graphics, and Interactive Systems, 123:6, dostupno na: https://cdn.manesht.ir/9961___Simon_Pedersen_CircularHoughTransform.pdf (pristupljeno: 1.8.2023.)

[22] Medium, What Is Annotation In AI / ML ?, 1.2.2019., dostupno na: https://medium.com/@orendaminds/what-is-annotation-in-ai-ml-df63af8599e8 (pristupljeno: 28.6.2023.)

[23] MakeSense, dostupno na: https://www.makesense.ai/ (pristupljeno: 1.2.2023.)

[24] Antonio Parunov, Augmentacija slika koristeći TensorFlow, Megatrend, 21.2.2022., dostupno na: https://www.megatrend.com/augmentacija-slika-koristeci-tensorflow/ (pristupljeno: 15.2.2023.)

[25] Albumentations, Full API Reference on a single page, dostupno na: https://albumentations.ai/docs/api_reference/full_reference/ (pristupljeno: 22.2.2023.)

[26] Albumentations, dostupno na: https://albumentations.ai/ (pristupljeno: 22.2.2023.)

[27] Albumentations, Bounding boxes augmentation for object detection, dostupno na: https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/ (pristupljeno: 24.2.2023.)

[28] PseudoLab Tutorial Book, 1. Introduction to Object Detection, dostupno na: https://pseudo-lab.github.io/Tutorial-Book-en/chapters/en/object-detection/Ch1-Object-Detection.html (pristupljeno: 24.7.2023.)

[29] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, 22.10.2014., dostupno na: https://doi.org/10.48550/arXiv.1311.2524 (pristupljeno: 7.7.2023.)

[30] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders, Selective Search for Object Recognition, University of Trento, Italy, University of Amsterdam, the Netherlands, 2012., dostupno na: http://www.huppelen.nl/publications/selectiveSearchDraft.pdf (pristupljeno: 5.7.2023.)

[31] Geeksforgeeks, R-CNN vs Fast R-CNN vs Faster R-CNN | ML, dostupno na: https://www.geeksforgeeks.org/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-ml/ (pristupljeno: 3.7.2023.)

[32] Ross Girshick, Fast R-CNN, 27.9.2015., dostupno na: https://doi.org/10.48550/arXiv.1504.08083 (pristupljeno: 15.7.2023.)

[33] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 6.1.2016., dostupno na: https://doi.org/10.48550/arXiv.1506.01497 (pristupljeno: 20.7.2023.)

[34] Detekcija i klasifikacija objekata primjenom strojnog učenja, Predavanja iz kolegija Računalna geometrija i robotski vid

[35] Enes Zvornicanin, What Does Backbone Mean in Neural Networks?, Baeldung, 17.12.2020., dostupno na: https://www.baeldung.com/cs/neural-network-backbone (pristupljeno: 28.7.2023.)

[36] Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 10.4.2015., dostupno na: https://doi.org/10.48550/arXiv.1409.1556 (pristupljeno: 15.7.2023.)

[37] Rohini G, Everything you need to know about VGG16, 23.9.2021., dostupno na: https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918 (pristupljeno: 15.7.2023.)

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 10.12.2015., dostupno na: https://doi.org/10.48550/arXiv.1512.03385 (pristupljeno: 21.7.2023.)

[39] Datagen, ResNet-50: The Basics and a Quick Tutorial, dostupno na: https://datagen.tech/guides/computer-vision/resnet-50/ (pristupljeno: 20.8.2023.)

[40] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, Feature Pyramid Networks for Object Detection, 19.4.2017., dostupno na: https://doi.org/10.48550/arXiv.1612.03144 (pristupljeno: 20.8.2023.)

[41] Medium, Understanding Feature Pyramid Networks for object detection (FPN), 27.3.2018., dostupno na: https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c (pristupljeno: 22.8.2023.)

[42] Joseph Redmon, Ali Farhadi, YOLO9000: Better, Faster, Stronger, 25.12.2016., dostupno na: https://doi.org/10.48550/arXiv.1612.08242 (pristupljeno: 28.6.2023.)

[43] Davor Širinić, Primjena YOLO algoritma računalnog vida, Megatrend, 8.9.2022., dostupno na: https://www.megatrend.com/primjena-yolo-algoritma-racunalnog-vida/ (pristupljeno: 1.8.2023.)

[44] Joseph Redmon, Ali Farhadi, YOLOv3: An Incremental Improvement, 8.4.2018., dostupno na: https://doi.org/10.48550/arXiv.1804.02767 (pristupljeno: 4.8.2023.)

[45] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, 23.4.2020., dostupno na: https://doi.org/10.48550/arXiv.2004.10934 (pristupljeno: 5.8.2023.)

[46] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, Densely Connected Convolutional Networks, 28.1.2018., dostupno na: https://doi.org/10.48550/arXiv.1608.06993 (pristupljeno: 18.7.2023.)

[47] Medium, Explanation of YOLO V4 a one stage detector, 7.9.2020., dostupno na: https://becominghuman.ai/explaining-yolov4-a-one-stage-detector-cdac0826cbd7 (pristupljeno: 20.7.2023.)

[48] Opengenus, YOLOv4 model architecture, dostupno na: https://iq.opengenus.org/yolov4-model-architecture/ (pristupljeno: 12.8.2023.)

[49] Jacob Solawetz, Samrat Sahoo, Train YOLOv4-tiny on Custom Data - Lightning Fast Object Detection, 1.7.2020., dostupno na: https://blog.roboflow.com/train-yolov4-tiny-on-custom-data-lighting-fast-detection/ (pristupljeno: 5.8.2023.)

[50] Segmentacija slike, Predavanja iz kolegija Računalna geometrija i robotski vid

[51] Cvexplained, Computer Vision, dostupno na: https://cvexplained.wordpress.com/2020/04/28/color-detection-hsv/ (pristupljeno: 10.8.2023.)

[52] Rebecca Stone, Image Segmentation Using Color Spaces in OpenCV + Python, dostupno na: https://realpython.com/python-opencv-color-spaces/ (pristupljeno: 7.8.2023.)

[53] Adrian Rosebrock, Measuring size of objects in an image with OpenCV, 28.3.2016., dostupno na: https://pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/ (pristupljeno: 20.7.2023.)

[54] Python, dostupno na: https://www.python.org/ (pristupljeno: 1.7.2023.)

[55] Google Colab, Google Colaboratory, dostupno na: https://colab.google/ (pristupljeno: 1.7.2023.)

[56] AlexeyAB-darknet, GitHub, dostupno na: https://github.com/AlexeyAB/darknet (pristupljeno: 20.7.2023.)

[57] PseudoLab Tutorial Book, Faster R-CNN, dostupno na: https://pseudo-lab.github.io/Tutorial-Book-en/chapters/en/object-detection/Ch5-Faster-R-CNN.html (pristupljeno: 20.7.2023.)

SAŽETAK

U ovom diplomskom radu testirano je nekoliko algoritama računalnog vida i umjetnih neuronskih mreža koje bi se mogle primijeniti pri prepoznavanju voća u kartonskim kutijama. Prikupljene su slike za podatkovni skup za prepoznavanje jabuka sorte Granny Smith. Podatkovni skup je anotiran i augmentiran. Za detektiranje jabuka korištene su Faster RCNN i YOLO arhitekture neuronskih mreža. Sve mreže su imale odlične evaluacijske metrike. Pri procjeni veličine jabuka korištena je segmentacija slike po boji. Segmentiranjem slike po boji, uz korištenje OpenCV biblioteke, prepoznaju se jabuke manjih dimenzija. Za prebrojavanje jabuka unutar kutije testirana je Houghova transformacija za krugove. Zbrojiti prepoznate jabuke (izlaze) koje daju umjetne neuronske mreže pokazao se kao bolji izbor za prebrojavanje jabuka unutar kutije. Testirani algoritmi mogu poslužiti pri kontroli kvalitete voća unutar kutija u industrijskim postrojenjima.

Ključne riječi: detekcija jabuka, podatkovni skup, prebrojavanje jabuka, procjena veličine jabuka, umjetne neuronske mreže

ABSTRACT

In this paper, several computer vision algorithms and artificial neural networks that could be applied in the recognition of fruit in cardboard boxes were tested. Images were collected for the Granny Smith apple recognition dataset. The dataset is annotated and augmented. Faster RCNN and YOLO neural network architectures were used to detect apples. All networks had excellent evaluation metrics. Color image segmentation was used to estimate the size of apples. By using color image segmentation and OpenCV library, apples of smaller dimensions are recognized. To count the apples inside the box, the Hough transform for circles was tested. Summing recognized apples (outputs) provided by artificial neural networks proved to be a better choice for counting the apples inside the box. The tested algorithms can be used to control the quality of fruit inside boxes in industrial plants.

Keywords: fruit detection, dataset, fruit counting, estimate fruit size, artificial neural networks

PRILOZI

P.3.1. – Podatkovni skup (priloženo na DVD-u uz rad)