Razvoj metode za automatsku detekciju brodova

Arapović, Andrea

Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija
Diplomski studij – Poslovna informatika

Andrea Arapović

Razvoj metode za automatsku detekciju brodova

Diplomski rad

Mentor: izv. prof. dr. sc. Marina Ivašić-Kos

Rijeka, prosinac 2022.

Zadatak za diplomski rad

Rijeka, 15.2.2022.

Zadatak za diplomski rad

Pristupnik: **Andrea Arapović**

Naziv diplomskog rada**: Razvoj metode za automatsku detekciju brodova**

Naziv diplomskog rada na eng. jeziku: **Development of a model for automatic ship detection**

Sadržaj zadatka:

Proučiti i opisati osnovne koncepte strojnog i dubokog učenja te metode učenja i biblioteke funkcija koje se koriste kod rješavanja zadataka računalnog vida. Objasniti razliku između zadataka klasifikacije slika i detekcija objekta na slici.

Istražiti alate koji se koriste za učenje modela neuronske mreže, odabrati arhitekture neuronskih mreža koja će se koristiti za detekciju brodova te prikazati postupak podešavanje tehnološkog stoga i korištenja mreže za zadatke detekcije.

Prikupiti slike brodova s javno dostupnih baza slika te opisati postupak učenja modela neuronske mreže za detekciju brodova te evaluirati model kvantitativno pomoću odgovarajućih metrika i kvalitativno na nekoliko testnih primjera.

Mentor:

Izv. prof. dr. sc. Marina Ivašić-Kos

Voditeljica za diplomske radove:

prof. dr. sc. Ana Meštrović

Zadatak preuzet: 27.02.2022

(potpis pristupnika)

Sažetak i ključne riječi

U radu su opisani pojmovi umjetne inteligencije, strojnog učenja i računalnog vida. Nadalje je opisano kako se umjetna inteligencija koristi ili planira u budućnosti koristiti u pomorstvu, budući da se rad bazira na razvoju metode za detekciju brodova.

Veći fokus pridan je samoj detekciji objekata te je detaljno opisano što ta tehnika radi i kako funkcionira. Napravljena je i usporedba detekcije objekata na slici i klasifikacije slika budući da su to dva pojma koji se jako često spominju uz pojam računalnog vida.

Kao praktični dio rada, opisan je razvoj metode za detekciju objekata na slikama u Pythonu koristeći OpenCV biblioteku, u programskom okruženju PyCharm i s prethodno treniranim modelom. Napravljena je detekcija objekata i na statičkim slikama i na slikama direktno iz web kamere.

Ključne riječi: umjetna inteligencija, detekcija objekata, strojno učenje

Sadržaj:

1. Uvod

Izraz umjetna inteligencija prvi put je korišten 1956. godine, ali AI je danas postao popularniji zahvaljujući povećanim količinama podataka, naprednim algoritmima i poboljšanjima računalne snage i pohrane.

Rana istraživanja umjetne inteligencije u 1950-ima istraživala su teme poput rješavanja problema i simboličkih metoda [14]. U 1960-ima se Ministarstvo obrane SAD-a zainteresiralo za ovu vrstu posla i počelo se trenirati računala da oponašaju osnovno ljudsko razmišljanje. Na primjer, Agencija za napredna obrambena istraživanja (DARPA) dovršila je projekte mapiranja ulica 1970-ih. DARPA je 2003. proizvela i inteligentne osobne asistente, mnogo prije nego što su Siri, Alexa ili Cortana postali poznati.

Ovaj rani rad otvorio je put automatizaciji i formalnom zaključivanju koje danas vidimo u računalima, uključujući sustave za podršku odlučivanju i pametne sustave pretraživanja koji se mogu dizajnirati da nadopune i povećaju ljudske sposobnosti.

Dok holivudski filmovi i znanstveno-fantastični romani prikazuju umjetnu inteligenciju kao čovjekolike robote koji preuzimaju svijet, trenutna evolucija AI tehnologija nije tako zastrašujuća - ili baš toliko pametna [14]. Umjesto toga, AI se razvio kako bi pružio mnoge specifične prednosti u svakoj industriji.

Rad se sastoji od dva dijela, teoretskog i praktičnog. U teoretskom dijelu objašnjeni su pojmovi umjetne inteligencije, strojnog učenja i detekcije objekata, dok je u praktičnom dijelu opisan razvoj metode za detekciju objekata (brodova).

2. Umjetna inteligencija

Umjetna inteligencija (eng. *Artificial Intelligence*) odnosi se na simulaciju ljudske inteligencije u strojevima, tj. sposobnost strojeva da oponašaju inteligentno ljudsko ponašanje [3]. Takvi strojevi mogu obrađivati informacije te učiti i donositi odluke na temelju logike i razmišljanja. Specifične primjene umjetne inteligencije uključuju ekspertne sustave, obradu prirodnog jezika, prepoznavanje govora te strojni i računalni vid.

Idealna karakteristika umjetne inteligencije je njezina sposobnost racionalizacije i poduzimanja radnji koje imaju najbolje izglede za postizanje određenog cilja. Podskup umjetne inteligencije je strojno učenje (eng. *Machine Learning*), koje se odnosi na koncept da računalni programi mogu automatski učiti i prilagođavati se novim podacima bez ljudske pomoći [4]. Tehnike koje omogućuju takvo automatsko učenje kroz apsorpciju ogromnih količina nestrukturiranih podataka kao što su tekst, slike ili video, spadaju pod metode dubokog učenja (eng. *Deep Learning*) [4].

Kako umjetna inteligencija zapravo funkcionira?
Općenito, sustavi umjetne inteligencije funkcioniraju tako da koriste velike količine podataka za učenje, analiziraju te skupove da bi pronašli korelacije među podacima i koriste te uzorke i zaključke za predviđanje budućih stanja [3]. Kao primjere možemo uzeti chatbot koji je treniran primjerima tekstualnih razgovora iz kojih može naučiti stvarati realistične razgovore s ljudima ili alat za prepoznavanje slika koji može naučiti identificirati i opisati objekte na slikama treniranjem na milijunima primjera.

Mogućnosti primjene umjetne inteligencije u drugim sektorima i industrijama su velike i sve popularnije. Možemo uočiti veliki porast u modernizaciji unutar raznih sektora ljudskih djelatnosti što nerijetko uključuje i upotrebu umjetne inteligencije za izvršavanje raznih zadataka. Kao i svaka tehnologija, umjetna inteligencija ima svoje prednosti i mane. Pod prednosti spadaju [3]:

- Velika korist u poslovima koji pridonose veliku pažnju detaljima
- Smanjeno vrijeme obrade ogromnih količina podataka
- Konzistentnost rezultata

- Visoka točnost predikcija

Kada govorimo o manama sustava, često se spominju stvari poput [3]:

- Korištenje takvih tehnologija zahtijeva visoku stručnost
- Ograničena ponuda radnika koji imaju dovoljno znanja i stručnosti za razvoj kompleksnijih AI alata
- AI alat zna samo ono što mu je pokazano i što je naučio

Unatoč svim tim navodima, umjetna inteligencija može se pronaći u svim područjima ljudskog života kao što su npr. zdravstvo, obrazovanje, promet i ostali. U ovom radu fokus je na dva podskupa umjetne inteligencije, a to su strojno učenje i računalni vid.

2.1. Strojno učenje

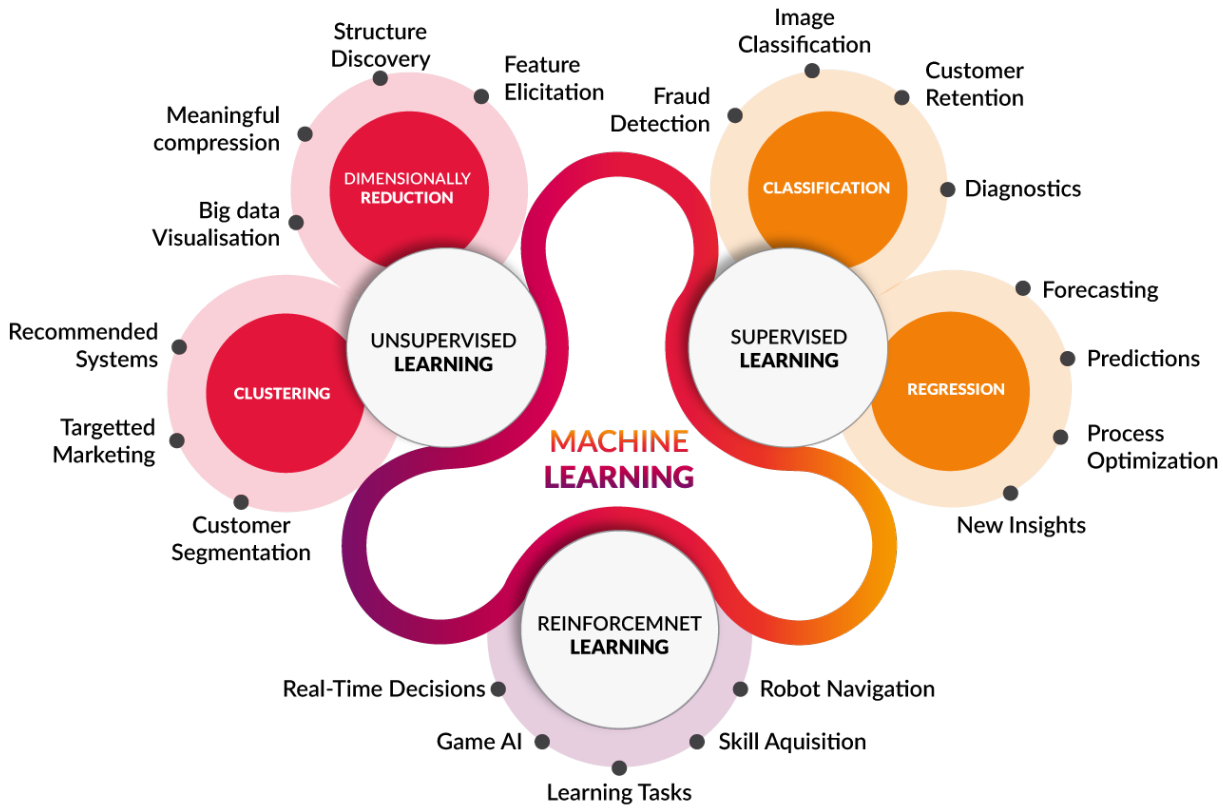
Kao disciplina, strojno učenje (eng. *Machine Learning)* istražuje analizu i konstrukciju algoritama koji mogu učiti iz podataka i predviđati ih [4]. ML se pokazao vrijednim jer može riješiti probleme velikom brzinom i u razmjeru kojeg ljudski um ne može sam kopirati. Uz ogromne količine računalne sposobnosti iza jednog zadatka ili više specifičnih zadataka, strojevi se mogu osposobiti za prepoznavanje uzoraka i odnosa između ulaznih podataka i automatiziranje rutinskih procesa.

Algoritmi koji pokreću strojno učenje ključni su za uspjeh. ML algoritmi su matematički modeli temeljeni na uzorcima podataka poznatijim kao „podaci za učenje“ te kao takvi mogu donositi odluke ili izvršavati predviđanja bez da su eksplicitno programirani za to [4]. Takve metode mogu otkriti uzorke ili trendove u podacima koje tvrtke mogu koristiti za poboljšanje donošenja odluka, optimizaciju učinkovitosti i prikupljanje velikog broja korisnih podataka.

ML pruža temelj za AI sustave koji automatiziraju procese i autonomno rješavaju poslovne probleme temeljene na podacima. Također omogućuje tvrtkama da zamijene ili povećaju određene ljudske sposobnosti unutar svog poslovanja. Uobičajene aplikacije za strojno učenje koje se mogu

pronaći u raznim sektorima ljudske djelatnosti uključuju chatbotove, samovozeće automobile, prepoznavanje govora i ostale.

Strojno učenje je vrlo širok pojam i kao takav ima više različitih pristupa i metoda kao što su nadzirano učenje, nenadzirano učenje i poboljšano učenje (slika 1) [4].



Slika 1: Metode strojnog učenja (izvor - zadnje posjećeno 05. prosinca 2022.: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d)

Nadzirani algoritmi strojnog učenja (eng. *Supervised Learning*) primjenjuju ono što je naučeno u prošlosti na nove podatke pomoću označenih primjera za predviđanje budućih događaja. Nakon uspješnog terniranja sustav može dati rezultate za svaki novi unos sličan podacima koji su bili u skupu za učenje.

Nenadzirano učenje (eng. *Unsupervised Learning*) je tehnika strojnog učenja u kojoj korisnici ne moraju nadzirati model [4]. Umjesto toga, omogućuje modelu da samostalno radi na otkrivanju obrazaca i informacija koje prije nisu bile otkrivene. Uglavnom se bavi neoznačenim podacima.

Poboljšano učenje (eng. *Reinforcement Learning*) bavi se učenjem putem interakcije i povratne informacije, odnosno, učenjem rješavanja zadatka pokušajem i pogreškom, djelovanjem u okruženju i primanjem nagrade za to [4]. U osnovi je izgrađen agent (ili nekoliko njih) koji može percipirati i interpretirati okolinu u kojoj se nalazi, štoviše, može poduzimati radnje i komunicirati s njom.

Važno je razumjeti što strojno učenje može, a što ne može. Suprotno uvriježenom mišljenju, strojno učenje ne može postići inteligenciju na ljudskoj razini. Strojeve pokreću podaci, a ne ljudsko znanje. Kao rezultat toga, „inteligencija“ je određena količinom podataka potrebnom za treniranje modela [4].

2.2. Računalni vid

Računalni vid je polje računalne znanosti koje se fokusira na repliciranje dijelova složenosti sustava ljudskog vida i omogućavanje računalima da identificiraju i obrađuju objekte u slikama i video zapisima na isti način na koji to rade ljudi [1]. Donedavno je računalni vid radio samo u ograničenom kapacitetu, no zahvaljujući napretku u umjetnoj inteligenciji i inovacijama u dubokom učenju i neuronskim mrežama, to polje je posljednjih godina uspjelo napraviti veliki skok i nadmašiti ljude u nekim zadacima povezanima s detekcijom i označavanjem objekata.

Jedan od pokretačkih čimbenika koji stoje iza rasta popularnosti računalnog vida je količina podataka koje danas generiramo, a koji se zatim koriste za učenje i poboljšanje rezultata algoritama računalnog vida. U manje od jednog desetljeća, točnost klasifikacije današnjih sustava porasla je s 50% na visokih 99%, što pokazuje veliki pomak u razvoju ove tehnologije [12]. Primjer korištenja računalnog vida u detekciji objekata nalazi se na slici 2.

Slika 2: Primjer rezultata algoritma računalnog vida za detekciju objekata u praksi (izvor – zadnje posjećeno 05. prosinca 2022.: https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e)

Računalni vid nije nova tehnologija. Prvi eksperimenti s računalnim vidom započeli su 1950-ih, a tada se koristio za tumačenje tipkanog i rukom pisanog teksta [5]. U to su vrijeme postupci analize računalnog vida bili relativno jednostavni, ali su zahtijevali puno rada ljudskih operatera koji su morali ručno osigurati i pripremiti uzorke podataka za analizu te je bilo teško manualno unijeti i označiti velike količine podataka. Osim toga, računalna snaga nije bila dovoljno dobra, tako da je margina pogreške za ovu analizu bila prilično visoka.

Algoritmi računalnog vida koje danas koristimo temelje se na prepoznavanju uzoraka [5]. Računalne algoritme učimo na ogromnoj količini vizualnih podataka - računala obrađuju slike, označavaju objekte na njima i pronalaze uzorke u tim objektima. Na primjer, ako mu pošaljemo milijun slika cvijeća, računalo će ih analizirati, identificirati uzorke koji su slični svim cvjetovima i na kraju ovog procesa izradit će model „cvijeta“. Kao rezultat toga, računalo će moći točno otkriti je li određena slika cvijet svaki put kada bude imao takav primjer slike u upitu.

Računalni vid danas je prisutan u mnogim aspektima ljudskog života kao što su proširena stvarnost (VR naočale), autonomna vozila, prepoznavanje osoba na slikama i snimkama, detekcija

raka i ostalih zloćudnih slučajeva kod pacijenata i još mnogi drugi. Ogromne količine podataka koje svakodnevno stvaramo, za koje neki ljudi misle da su prokletstvo naše generacije, zapravo se koriste za našu dobrobit tj. podaci mogu naučiti računala da vide i razumiju objekte.

2.3. Usporedba strojnog učenja i računalnog vida

Jednostavno rečeno, računalni vid je tehnologija koja pokušava osposobiti računala da prepoznaju obrasce u vizualnim podacima na sličan način kao što to rade ljudi. S druge strane, strojno učenje je proces koji omogućuje računalima da nauče kako obraditi i reagirati na unose podataka. Ukratko, strojno učenje je općenitije i ne uključuje nužno vizualne podatke [6].

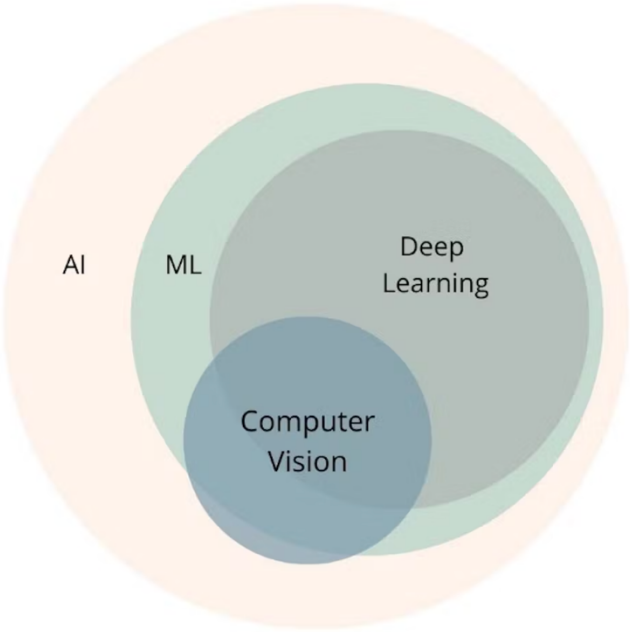
I kod računalnog vida i kod strojnog učenja, cilj je da računalni sustavi nauče kako obraditi podatke i reagirati na njih u određenoj situaciji. Međutim, računalni vid je puno više fokusiran na slike i vizualne podatke, dok se strojno učenje fokusira na druge vrste podataka i ima za cilj rješavanje klasifikacije slika, detekcije objekata, segmentacije objekata te praćenja objekata u videozapisima. U oba slučaja ishod je isti, no vrsta ulaznih podataka određuje koji će proces učenja najbolje funkcionirati. Ključni ishod je da bi računalni sustav trebao moći učiti iz prethodnih podataka.

Kada govorimo o računalnom vidu, podaci se koriste u širokom spektru aplikacija te su potrebne više od jedne vrste metoda analize podataka da bi ih računala mogla koristiti i izvoditi zaključke [6]. Primjeri uključuju medicinske dijagnostičke postupke, poljoprivredu i pomoć pri autonomnoj vožnji (vizualni aspekti). Nasuprot tome, strojno učenje također se koristi kada se podaci koji se analiziraju sastoje od podataka temeljenih na tekstu ili govoru. Primjeri za to uključuju prepoznavanje govora, analizu financijskih podataka, analizu podataka o prometu, analizu e-pošte i slično.

Računalni vid i strojno učenje važni su aspekti umjetne inteligencije. Već su poboljšali najsuvremenije razine točnosti i izvedbe u nekoliko zadataka kao što su klasifikacija slika, detekcija objekata i segmentacija.

Dodatno, kombinacija računalnog vida i strojnog učenja usmjerava i pojednostavljuje stvaranje učinkovitih tehničkih pristupa, aplikacija i sustava za sve glavne industrije i poslovne sektore.

Strojno učenje i računalni vid dva su napredna tehnička područja koja su evoluirala i postala usko povezana. Strojno učenje poboljšalo je računalni vid u pogledu praćenja i prepoznavanja. Osim toga, nudi učinkovite tehnike za prikupljanje podataka, obradu digitalne slike i detekciju objekata - tehnike koje se sve koriste unutar računalnog vida [6]. U usporedbi, strojno učenje je šire tehničko područje, a njegovi općenitiji algoritmi mogu se koristiti u drugim područjima i poljima (slika 3).



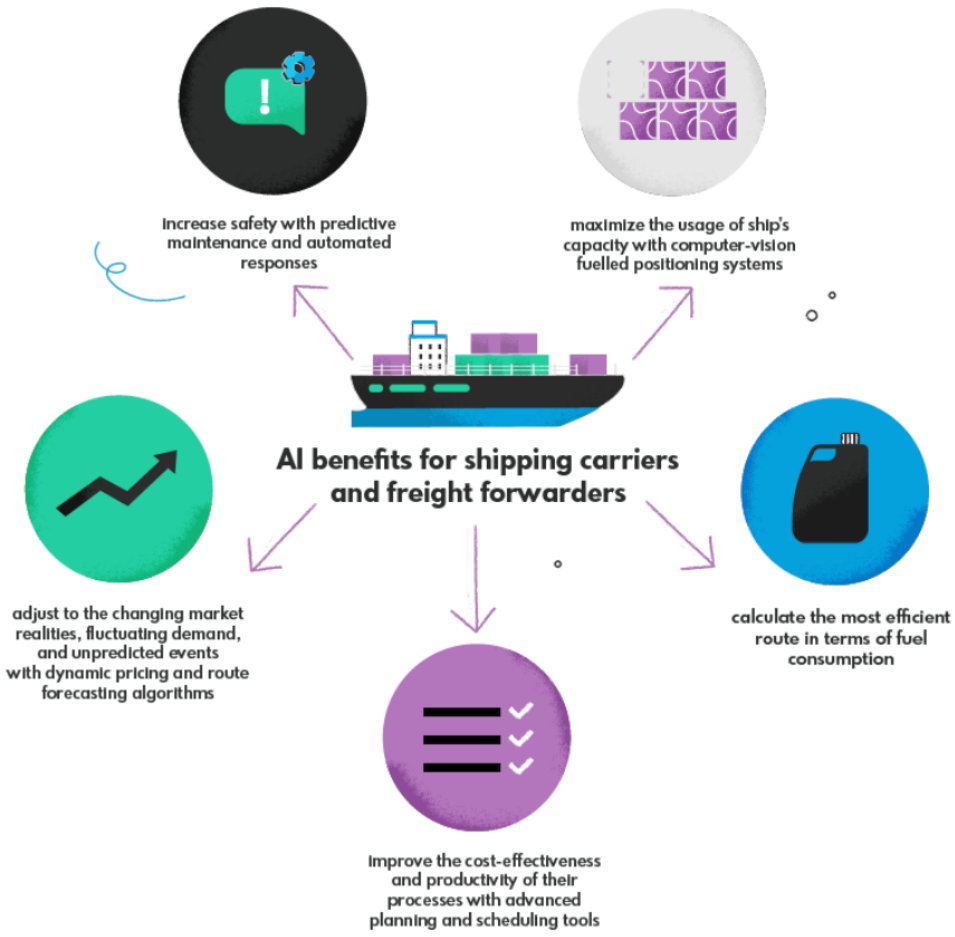
Slika 3: Podskupi umjetne inteligencije (izvor – zadnje posjećeno 05. prosinca 2022.: https://peltarion.com/blog/applied-ai/what-is-computer-vision)

3. Umjetna inteligencija u pomorstvu

Umjetna inteligencija (AI) počinje ulaziti u pomorsku industriju, s potencijalom za uvođenje autonomnih sustava, učinkovitijim procesima i povećanjem sigurnosti. Kao jedan od najstarijih sektora na svijetu, pomorstvo se tradicionalno uvelike oslanjalo na ljude i njihovo iskustvo. No, velike promjene dolaze unutar tog sektora.

Pomorska industrija idealna je za algoritme strojnog učenja s obzirom na to da se goleme količine podataka generiraju svaki dan u obliku otpremnih dokumenata, informacija o emisijama plinova, operativnih metrika i druge dokumentacije [7]. Sustavi umjetne inteligencije mogu se učinkovito baviti svim ovim aspektima. Također je važno uzeti u obzir visoke troškove povezane s mnogim procesima otpreme, gdje čak i najmanja optimizacija dovodi do značajnih ušteda.

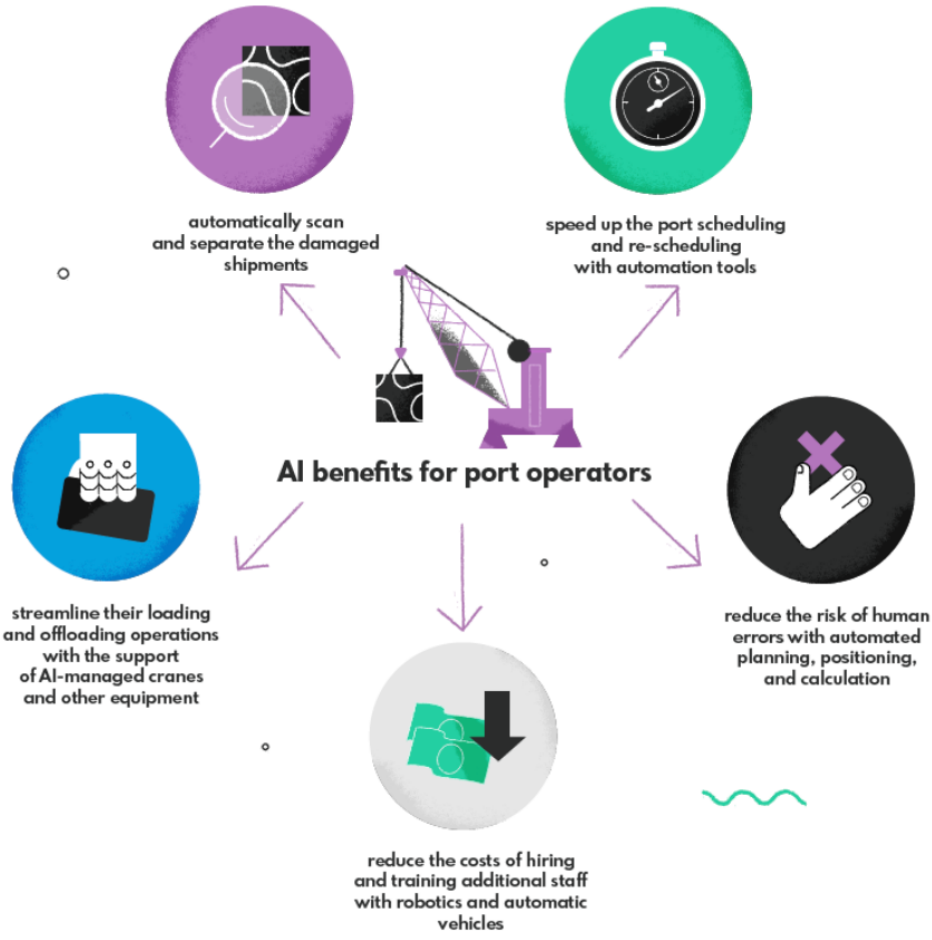
AI bi trebala ozbiljno uzdrmati sektor komercijalnog brodarstva u narednim desetljećima, s „pametnim brodom“ (potpuno digitaliziranim, autonomnim brodom bez posade) kao jednim od glavnih ciljeva koje treba postići. Sigurnost je još jedan važan cilj s obzirom na to da je između 75% i 95% svih nesreća na moru rezultat ljudske pogreške i stoga bi se uvelike mogle izbjeći upotrebom umjetne inteligencije [7]. Benefiti korištenja umjetne inteligencije u brodarstvu prikazani su na slici 4.



Slika 4: Benefiti korištenja umjetne inteligencije u brodarstvu (izvor – zadnje posjećeno 05. prosinca 2022.: https://nexocode.com/blog/posts/ai-in-maritime-artificial-intelligence-solutions-in-the-shipping-sector/)

U lukama, umjetna inteligencija je usko povezana s robotikom, budući da mnogi sustavi umjetne inteligencije imaju robotske elemente koji omogućuju autonomno izvršavanje zadataka. Roboti su postavljeni da rade uz vozila kako bi raspakiravali transportne kontejnere i obavljali druge zadatke [7]. Slično konceptu pametnog broda, krajnji cilj ovih razvoja je potpuno digitalizirana pametna luka, gdje se podaci razmjenjuju kroz cijeli logistički lanac, a AI se koristi za optimizaciju procesa. Središnji koncept ovdje je „digitalni blizanac“, koji uključuje potpunu digitalnu kartu svih procesa koji se odvijaju unutar postavke luke. Kada je u pitanju brodogradnja, AI je u početku postavljena da se pretežno koristi za dizajne i optimizaciju procesa. Ubuduće se AI neće oslanjati samo na praćenje podataka senzora. Također će se koristiti u dronovima kao

način nadzora brodova i lučkih objekata [7]. Benefiti korištenja umjetne inteligencije u lučkom prometu prikazani su na slici 5.



Slika 5: Benefiti korištenja umjetne inteligencije u lučkom prometu (izvor – zadnje posjećeno 05. prosinca 2022.: https://nexocode.com/blog/posts/ai-in-maritime-artificial-intelligence-solutions-in-the-shipping-sector/)

Poslovne prednosti umjetne inteligencije u pomorskoj industriji su neosporive, potiču rast poslovanja i stvaraju prostor za razvoj i nova ulaganja. Odustajanjem od ovih tehnologija tvrtke gube prednost pred konkurencijom koja će u bliskoj budućnosti prihvatiti AI.

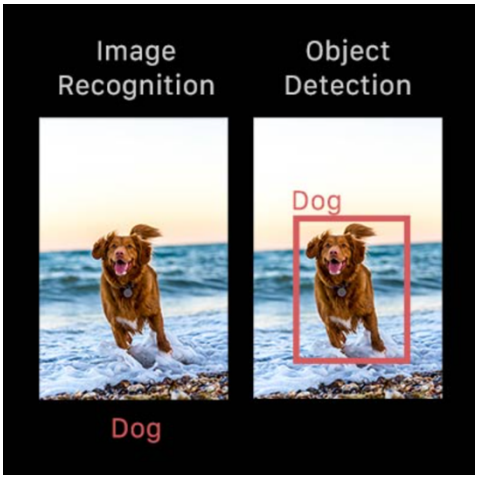
Malo je vjerojatno da će se svijet odmaknuti od globalizirane trgovine - stoga je ključno pomorsku industriju učiniti zelenijom. Njegov značajan utjecaj na okoliš zbog emisija CO_2 i zagađenja može se smanjiti algoritmima strojnog učenja koji sugeriraju najodrživiju upotrebu

resursa, rute koje troše najmanje goriva i automatiziranu konfiguraciju spremnika koja upravljanje prostorom dovodi na višu razinu [7]. To čini umjetnu inteligenciju još perspektivnijom u kontekstu pomorske industrije.

4. Detekcija objekata

Detekcija objekata tehnika je računalnog vida koja radi na identificiranju i lociranju objekata unutar slike ili videa. Točnije, detekcija objekata iscrtava granične okvire oko tih otkrivenih objekata, što nam omogućuje da lociramo gdje se navedeni objekti nalaze (ili kako se kreću kroz) određenu scenu [8].

Detekcija objekata obično se uspoređuje s raspoznavanjem slike, pa je važno razjasniti razlike između njih. Raspoznavanje slike dodjeljuje oznaku slici. Na primjer, slika psa dobiva oznaku „pas“. Slika na kojoj su dva psa, još uvijek dobiva oznaku „pas“. Detekcija objekata, s druge strane, crta kutiju oko svakog psa i označava kutiju oznakom „pas“ (slika 6). Model predviđa gdje se svaki objekt nalazi i koju oznaku treba primijeniti. Na taj način detekcija objekata daje više informacija o slici nego prepoznavanje [8].



Slika 6: Primjer prepoznavanja slike i detekcije objekta na slici (izvor – zadnje posjećeno 05. prosinca 2022.: https://www.fritz.ai/object-detection/)

Općenito, detekcija objekata može se podijeliti na pristupe temeljene na strojnom učenju i pristupe temeljene na dubokom učenju. U tradicionalnijim pristupima koji se temelje na ML-u, tehnike računalnog vida koriste se za promatranje različitih značajki slike, poput histograma boja ili rubova, kako bi se identificirale grupe piksela koji mogu pripadati objektu. Te se značajke zatim unose u regresijski model koji predviđa lokaciju objekta zajedno s njegovom oznakom. S druge strane, pristupi koji se temelje na dubokom učenju koriste konvolucijske neuronske mreže (CNN)

za izvođenje end-to-end nenadzirane detekcije objekta, u kojoj se značajke ne moraju definirati i ekstrahirati zasebno [8].

Detekcija objekata povezana je s drugim sličnim tehnikama računalnog vida kao što su raspoznavanje slike i segmentacija slike, jer nam pomaže razumjeti i analizirati scene u slikama ili videu. Ipak, postoje važne razlike. Raspoznavanje slike daje samo oznaku klase za identificirani objekt, a segmentacija slike svaki piksel na slici pridružuje odgovarajućem objektu i omogućuje raspodjelu elemenata scene u objekte na razini piksela. Ono što odvaja detekciju objekata od ovih drugih zadataka je njegova jedinstvena sposobnost lociranja objekata unutar slike ili videa. To nam onda omogućuje brojanje i praćenje tih objekata [8].

4.1. Korištenje detekcije objekata u stvarnom svijetu

Video nadzor:

Budući da najsuvremenije tehnike detekcije objekata mogu točno identificirati i pratiti više instanci neke klase u sceni, ove tehnike prirodno se mogu koristiti za automatizaciju sustava video nadzora. Na primjer, modeli detekcije objekata sposobni su pratiti više ljudi odjednom, u stvarnom vremenu, dok se kreću kroz određenu scenu ili kroz video okvire. Od maloprodajnih trgovina do industrijskih tvornica, ova vrsta praćenja mogla bi pružiti veliki doprinos u očuvanju sigurnosti kako radnika, kupaca i smanjenje troškova od krađa u maloprodajnom prometu i slično [8].

Prebrojavanje ljudi u gomili:

Prebrojavanje ljudi u gomili ili u velikim grupama još je jedna korisna primjena detekcije objekata. Za gusto naseljena područja kao što su tematski parkovi, trgovački centri i gradski trgovi, detekcija objekata može pomoći tvrtkama i općinama da učinkovitije mjere različite vrste prometa - bilo pješice, u vozilima ili na neki drugi način. Ova mogućnost lokalizacije i praćenja ljudi dok se kreću kroz različite prostore mogla bi pomoći tvrtkama da optimiziraju bilo što, od logističkih cjevovoda i upravljanja zalihama, do radnog vremena skladišta, rasporeda smjena i ostalo [8].

Otkrivanje anomalija:

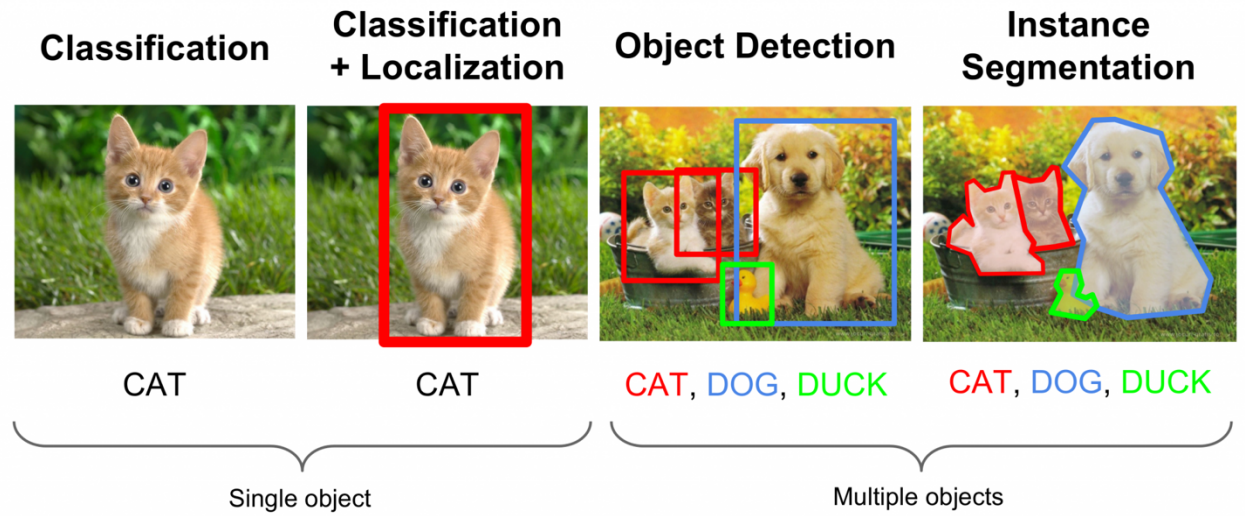
Otkrivanje anomalija je slučaj upotrebe detekcije objekata koji se najbolje objašnjava kroz konkretne primjere industrije. U poljoprivredi, na primjer, prilagođeni model detekcije objekata mogao bi točno identificirati i locirati potencijalne slučajeve biljnih bolesti, omogućujući poljoprivrednicima da otkriju prijetnje njihovim prinosima koje inače ne bi bile vidljive golim ljudskim okom. U zdravstvu bi se detekcija objekata mogla koristiti za pomoć u liječenju stanja koja imaju specifične i jedinstvene simptomatske lezije. Jedan takav primjer dolazi u obliku njege kože i liječenja akni - model detekcije predmeta mogao bi locirati i identificirati primjere akni u nekoliko sekundi. Ono što je osobito važno i uvjerljivo u ovim potencijalnim slučajevima upotrebe jest kako pružaju znanje i informacije koje su općenito dostupne samo poljoprivrednim stručnjacima, odnosno liječnicima [8].

Autonomna vozila:

Modeli detekcije automobila u stvarnom vremenu ključni su za uspjeh autonomnih sustava vozila. Ovi sustavi moraju biti sposobni identificirati, locirati i pratiti objekte oko sebe kako bi se kretali svijetom sigurno i učinkovito. Dok se zadaci poput segmentacije slike mogu (i često jesu) primijeniti na autonomna vozila, detekcija objekata ostaje temeljna zadaća koja podupire trenutni rad na pretvaranju autonomnih vozila u stvarnost [8].

4.2. Usporedba klasifikacije slike i detekcije objekata

Tehnike klasifikacije slika i detekcije objekata važne su metode kada se radi o domeni računalnog vida. Ove tehnike pomažu strojevima razumjeti i identificirati objekte i okruženja u stvarnom vremenu uz pomoć digitalnih slika kao ulaznih podataka (slika 7). Tijekom godina, tehnike računalnog vida korištene su u nekoliko sektora, uključujući zdravstvo, proizvodnju, maloprodaju i mnoge druge [9].



Slika 7: Razlike između najpoznatijih metoda računalnog vida (izvor – zadnje posjećeno 05. prosinca 2022.: https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852)

Budući da se metode kao što su klasifikacija slika i detekcija objekata vrte oko identificiranja objekata u digitalnim slikama, često se postavlja pitanje: što su te dvije tehnike zapravo i kako se te dvije tehnike razlikuju jedna od druge?

Jednostavnim riječima, klasifikacija slike je tehnika koja se koristi za klasifikaciju ili predviđanje klase određenog objekta na slici [9]. Glavni cilj ove tehnike je točno identificirati značajke na slici. Općenito, tehnike klasifikacije slika mogu se kategorizirati kao parametarske i neparametarske ili nadzirane i nenadzirane. Nadzirana klasifikacija daje rezultate temeljene na stvorenoj granici odluke, koja se uglavnom oslanja na ulazne i izlazne podatke dane tijekom učenja modela. U slučaju nenadzirane klasifikacije, tehnika daje rezultat temeljen na analizi vlastitog ulaznog skupa podataka u kojem traži neke pravilnosti ili uzorke [9].

Glavni koraci uključeni u tehnike klasifikacije slika su određivanje prikladnog sustava klasifikacije, izdvajanje značajki, odabir dobrih uzoraka za učenje, prethodna obrada slike i odabir odgovarajuće metode klasifikacije, obrada nakon klasifikacije i konačno procjenjivanje ukupne točnosti [9]. U ovoj tehnici, ulazi su obično slika određenog objekta, a izlazi su predviđene klase koje definiraju i odgovaraju ulaznim objektima. Konvolucijske neuronske mreže (CNN) najpopularniji su model neuronske mreže koji se koristi za problem klasifikacije slika.

U nadziranim i nenadziranim tehnikama klasifikacije slika, nedostaci su velika količina vremena potrebna tijekom faze obuke i nisu prikladne za rad s velikim podacima.

Definicija problema detekcije objekta je odrediti gdje se objekti nalaze na danoj slici te kojoj klasi svaki objekt pripada, tj. Klasifikacija objekta [9]. Jednostavnim riječima, detekcija objekata je vrsta tehnike klasifikacije slika, a osim klasificiranja, ova tehnika dodatno još identificira lokaciju instanci objekta iz velikog broja unaprijed definiranih kategorija u slikama. Ova tehnika ima mogućnost traženja određene klase objekata, kao što su automobili, ljudi, životinje, ptice itd. i uspješno se koristi u sljedećoj generaciji sustava za obradu slika kao i video. Nedavni napredak u ovoj tehnici postao je moguć tek pojavom metodologija dubokog učenja.

Tehnike detekcije objekata mogu se koristiti u projektima iz stvarnog svijeta kao što su detekcija lica, detekcija pješaka, detekcija vozila, detekcija prometnih znakova, videonadzor i ostale.

Tradicionalni modeli detekcije objekata mogu se uglavnom podijeliti u tri faze, a to su izbor informativne regije, ekstrakcija značajki i klasifikacija [9]. Postoji nekoliko popularnih modela temeljenih na dubokom učenju za detekciju objekata kao što su MobileNet, You Only Live Once (YOLO), Mark-RCNN, RetinaNet i ostale [13].

Tijekom proteklih nekoliko godina postignut je veliki uspjeh u kontroliranom okruženju za problem otkrivanja objekata. Međutim, problem ostaje neriješen na nekontroliranim mjestima, posebice kada se predmeti postavljaju u proizvoljnim položajima u pretrpanom i zatvorenom okruženju.

5. Razvoj metode za detekciju objekata u Pythonu

U ovom dijelu rada bit će prikazan praktični primjer detekcije objekata na slikama, točnije fokus će biti na detekciju brodova na slikama. Cijela metoda programirana je u Pythonu (programsko okruženje PyCharm), a sama detekcija izvršena je pomoću biblioteke OpenCV i prethodno treniranih modela skinutih s GitHuba.

5.1. OpenCV

OpenCV (eng. *Open Source Computer Vision Library*) biblioteka je softvera za računalni vid i strojno učenje otvorenog koda. OpenCV je napravljen kako bi osigurao zajedničku infrastrukturu za aplikacije računalnog vida i ubrzao korištenje strojne percepcije u komercijalnim proizvodima. Budući da je proizvod s licencom za Apache 2, OpenCV tvrtkama olakšava korištenje i izmjenu koda [10].

Biblioteka ima više od 2500 optimiziranih algoritama, što uključuje opsežan skup klasičnih i najsuvremenijih algoritama računalnog vida i strojnog učenja. Ovi se algoritmi mogu koristiti za otkrivanje i prepoznavanje lica, identificiranje objekata, klasificiranje ljudskih radnji u videozapisima, praćenje pokreta kamere, praćenje pokretnih objekata, izdvajanje 3D modela objekata, spajanje slika kako bi se proizvela visoka rezolucija cijelog prizora, pronalazak sličnih slika iz baze podataka, uklanjanje crvenih očiju sa slika snimljenih bljeskalicom, praćenje pokreta očiju, prepoznavanje krajolika, itd. OpenCV ima više od 47 tisuća korisnika i procijenjeni broj preuzimanja veći je od 18 milijuna. Biblioteka se intenzivno koristi u tvrtkama, istraživačkim grupama i vladinim tijelima [10].

Uz dobro poznate tvrtke kao što su Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota koje koriste biblioteku, postoji i mnogo startupova kao što su Applied Minds, VideoSurf i Zeiter, koji u velikoj mjeri koriste OpenCV. Upotrebe OpenCV-a su svestrane, a neki od primjera su: spajanje slika uličnog prikaza, otkrivanje upada u videonadzoru u Izraelu, praćenje rudničke opreme u Kini, pomaganje robotima u navigaciji i skupljanju objekata u Willow Garageu, otkrivanje nesreća utapanja u bazenima u Europi, pokretanje interaktivne umjetnosti u Španjolskoj

i New Yorku, provjera pista za krhotine u Turskoj, pregled naljepnica na proizvodima u tvornicama diljem svijeta, brzo prepoznavanje lica u Japanu itd. [10]

Biblioteka također ima C++, Python, Java i MATLAB sučelja i podržava Windows, Linux, Android i Mac OS. OpenCV se uglavnom oslanja na vizualne aplikacije u stvarnom vremenu i koristi MMX i SSE upute kada su dostupne. Trenutno se aktivno razvijaju potpuno opremljena CUDA i OpenCL sučelja. Postoji preko 500 algoritama i oko 10 puta više funkcija koje sastavljaju ili podržavaju te algoritme. OpenCV je izvorno napisan u C++ i ima predložak sučelja koji besprijekorno radi sa STL spremnicima [10].

5.2. Priprema radnog okruženja i instalacija softvera

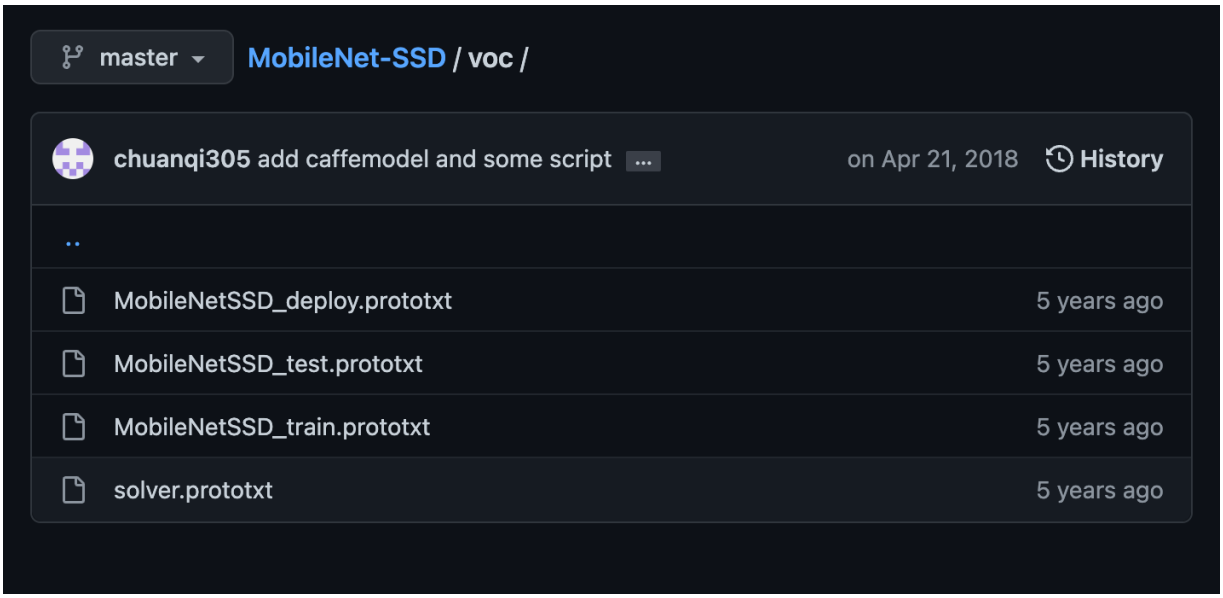
Za početak moramo pripremiti strukturu samog projekta budući da će nam za razvoj metode detekcije biti potrebne vanjska datoteke. Sama detekcija objekata, odnosno, brodova na slici, bit će izvedena s prethodno treniranim modelima koji su dostupni na GitHubu za slobodno korištenje.

Takvih modela ima više, no onaj koji je korišten u ovom projektu nalazi se u GitHub repozitoriju zvanom *MobileNet-SSD* [11]. Iz tog repozitorija potrebne su nam dvije datoteke, jedna je caffemodel (slika 8) i jedna je .prototxt datoteka.

Repozitorij sadrži i README.md datoteku koja sadrži sve upute o korištenju resursa koji su tamo dostupni, između ostalog i kako se koristi i na koji način je razvijen ovaj model. Potrebno je preuzeti te dvije datoteke prije nego krenemo s radom PyCharmu.

Slika 8: Caffemodel potreban za detekciju

Prototxt datoteka nalazi se u folderu voc/ te nam je potrebna ona koja u imenu sadrži riječ *deploy (slika 9)*.



Slika 9: Prototxt datoteka potrebna za detekciju objekata

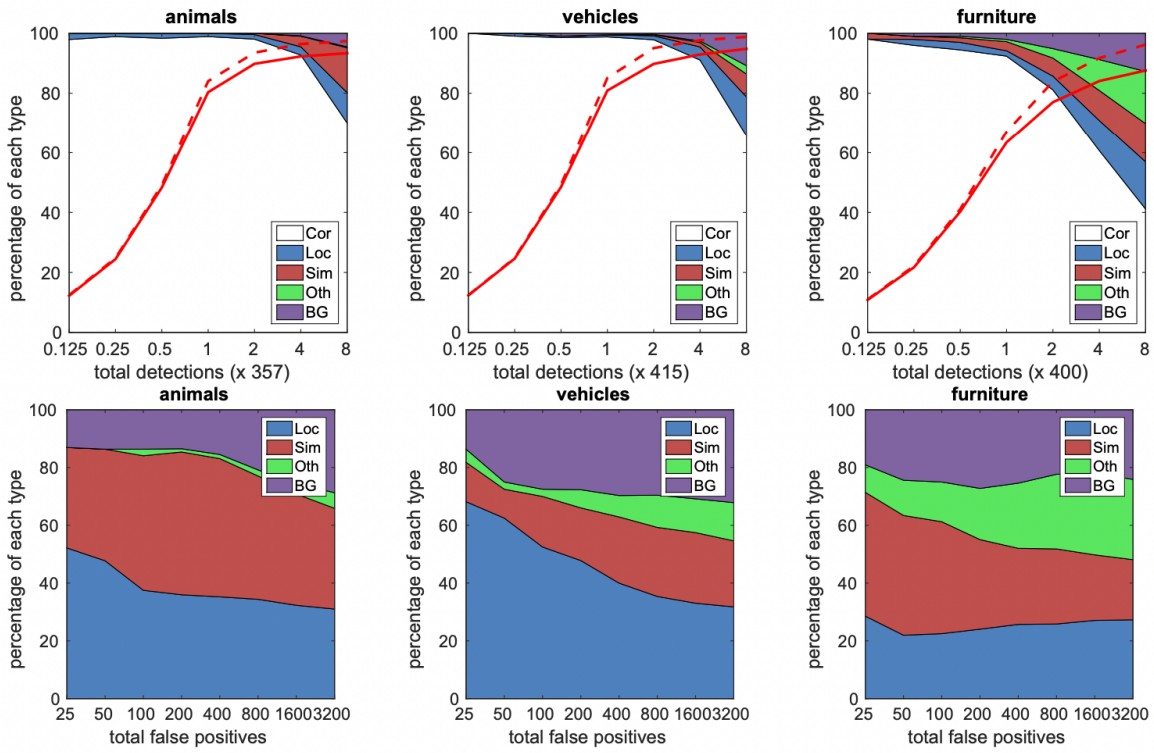
Što se tiče samog modela, radi se o metodi za detekciju objekata pomoću duboke neuronske mreže SSD (eng. *Single Shot MultiBox Detector*) [15]. Mreža generira rezultate za prisutnost svake kategorije objekata u svakom zadanom okviru (boxu) i prilagođava okvir kako bi bolje odgovarao obliku predmeta. To čini SSD lakim za učenje i jednostavnim za integraciju u sustave koji zahtijevaju komponentu detekcije.

Ekperimentalni rezultati potvrđuju da SSD ima konkurentnu točnost u odnosu na metode koje koriste dodatni korak prijedloga objekta i puno je brži od njih te uz to pruža jedinstveni radni okvir (eng. *framework*) za treniranje i zaključivanje. Za ulaznu sliku veličine 300 × 300 piksela, SSD postiže 74,3% mAP1 (postotak točnosti) na testu VOC2007 pri 59 FPS na Nvidia Titanu X i za ulaznu sliku veličine 512 × 512, postiže 76,9% mAP, nadmašujući najsuvremeniji Faster R-CNN model. U usporedbi s drugim jednofaznim metodama, SSD ima mnogo veću točnost čak i s manjom veličinom ulazne slike [15]. Na slici je prikazana usporedba točnosti ostalih modela s SSD-om ovisno o klasificiranim objektima te je jasno vidljivo kako SSD postiže veću točnost.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	**76.2**	57.6	87.3	88.2	88.6	60.5	85.4	**76.7**	**87.5**	**89.2**	84.5	81.4	55.0	81.9	**81.5**	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	**81.6**	**86.6**	**88.3**	**82.4**	76.0	**66.3**	**88.6**	**88.9**	**89.1**	**65.1**	**88.4**	73.6	86.5	88.9	**85.3**	**84.6**	**59.1**	**85.0**	80.4	**87.4**	**81.2**

Slika 10: Usporedba točnosti SSD modela s ostalim modelima (izvor – zadnje posjećeno 12. prosinca 2022.: https://arxiv.org/pdf/1512.02325.pdf)

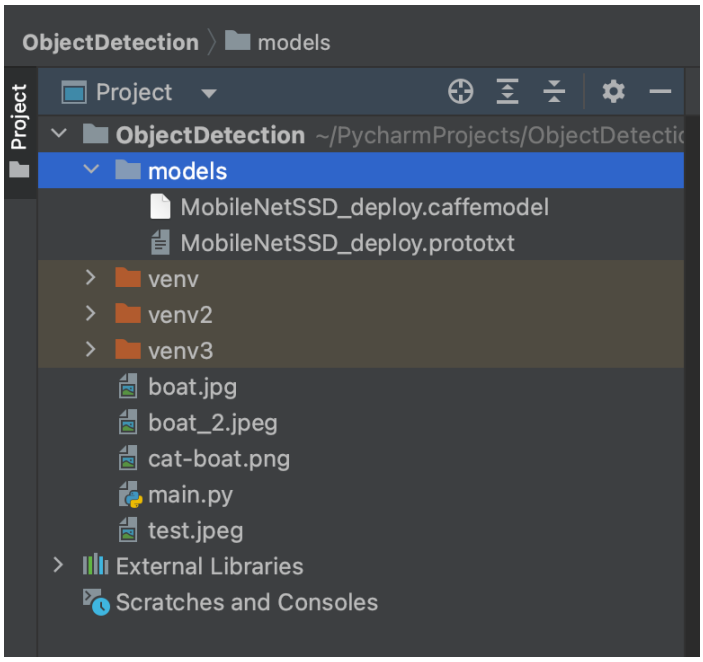
Brojevi 300 i 512 govore o dva SSD modela koji su trenirani unosom različitih dimenzija slika. SSD ima veću točnost na velikim objektima koji se nalaze na slikama. Povećanje veličine unosa (npr. od 300×300 do 512×512) može poboljšati otkrivanje malih objekata, ali ima još puno prostora za poboljšanje [15]. Vizualni prikaz evaluacije modela SSD516 prikazan je na slici.



Slika 11: Vizualizacija točnosti detekcije modela SSD516 (izvor – zadnje posjećeno 12. prosinca 2022.: https://arxiv.org/pdf/1512.02325.pdf)

Model SSD512 značajno nadmašuje najmoderniji Faster R-CNN u smislu točnosti na više različitih izvora podataka (eng. *datasetova*), dok je ujedno i tri puta brži. Model SSD300 u stvarnom vremenu radi pri 59 FPS, što je brže od trenutnog YOLO modela koji je alternativa, a istovremeno proizvodi puno veću točnost detekcije [15].

Kada smo preuzeli model s GitHuba, potrebno je u PyCharmu pripremiti strukturu datoteka za daljnji rad. U zaseban folder stavit ćemo oba modela koja smo preuzeli te ćemo još pronaći i nekoliko slika brodova i ostalih objekata kako bi mogli testirati radi li naš model detekcije dobro (slika 10).



Slika 12: Struktura foldera i datoteka unutar projekta u PyCharmu

U sljedećem koraku potrebno je instalirati biblioteke koje ćemo koristiti, a to su NumPy i OpenCV, a naredbe u Command Promptu (cmd) za to su:

pip install numpy
pip install opencv-python

Nakon što su biblioteke instalirane, okruženje je spremno za kodiranje.

5.3. Priprema podataka i definicija varijabli

Verzija Pythona korištena za ovaj projekt je 3.10. i cijeli kod je izveden u PyCharm okruženju. U prvom dijelu potrebno je uvesti biblioteke koje su preuzete u ranijim pripremama. Također je potrebno specificirati sve putanje do slika koje će biti korištene za proces detekcije objekata, kao i putanje do modela koji se koriste da bi proces detekcije radio (slika 13).

```python
import numpy as np
import cv2

image_path = 'cat-boat.png'
prototxt_path = 'models/MobileNetSSD_deploy.prototxt'
model_path = 'models/MobileNetSSD_deploy.caffemodel'
```

Slika 13: Uvoz biblioteka i definicija putanji objekata

U sljedećem koraku potrebno je definirati varijablu koja se zove *min_confidence*. Ta varijabla zapravo određuje s kolikim postotkom sigurnosti ova metoda prepoznaje objekte na slikama. Na primjer, ukoliko je na slici stolica, model može reći da je 99% siguran da je na slici stolica te će ju tako klasificirati i označiti/locirati na slici. Potrebno je postaviti minimalnu vrijednosti tog postotka sigurnosti modela nakon koje model više neće detektirati taj objekt na slici, točnije, ako je taj postotak premalen (npr. 12%), taj objekt neće biti detektiran i neće se prikazati njegova lokacija. Kao minimalan postotak sigurnosti za ovaj model odabrano je 20%, odnosno 0.2 pa tako i postavljamo varijablu (slika 14).

```python
# if this minimal confidence is not reached, the model will not recognize that object in the image (20% in this example)
min_confidence = 0.2
```

Slika 14: Postavljanje minimalnog postotka sigurnosti modela (izvor: izradila studentica)

Sljedeće je potrebno definirati klase za model. Budući da se koristi prethodno trenirani model, potrebno je unutar GitHub repozitorija pronaći koje klase su korištene pri izradi modela te ih sve spremiti u varijablu u obliku liste (slika 15).

```python
# using the classes from the file and putting them in a specific array to be used as detection objects later
classNames = ["background", "aeroplane", "bicycle", "bird", "boat",
              "bottle", "bus", "car", "cat", "chair", "cow",
              "diningtable", "dog", "horse", "motorbike", "person", "pottedplant",
              "sheep", "sofa", "train", "tvmonitor"]
```

Slika 15: Specificiranje klasa i spremanje u listu

Kao posljednju fazu pripreme, potrebno je definirati na koji način će model označiti detektirani objekt na slici. Za ovu metodu koristit će se pravokutnici u različitim bojama, točnije, svaka klasa detektiranih objekata imati će svoju boju (npr. ljudi u plavoj boji, stolice u crvenoj i slično). Također će iznad svakog pravokutnika biti definiran postotak sigurnosti modela za taj detektirani objekt. Generiramo listu boja u RGB rasponu od 0 do 255 gdje će duljina te liste biti ista kao i broj klasa unutar *classNames* liste koju smo ranije definirali. Na taj način ćemo dobiti željeni ispis različite boje za svaku klasu detektiranih objekata. Također je cilj da te nasumično generirane boje nisu jako slične pa je kroz par testiranja definiran i niz koji najbolje funkcionira za ovakav zadatak (54321) kao što je i vidljivo na slici 16.

```
# every class has its own color coordinated box
# random.seed() makes a determined random choice to make sure the colors it randomly chooses are not to similar
np.random.seed(543210)
colors = np.random.uniform(0, 255, size=(len(classNames), 3))
```

Slika 16: Definiranje načina označavanja detektiranih objekata na slici

5.4. Detekcija objekata na slici

Kada su sve biblioteke i varijable spremne, potrebno je izvršiti detekciju. Iako je ovaj rad fokusiran na detekciju brodova na slikama, uvezene su i druge klase kako bi se provjerila točnost modela i povećala mu se sveukupna funkcionalnost.

Sljedeće što je potrebno napraviti je uvesti duboku neuronsku mrežu i prethodno trenirani model u našu metodu te i im proslijediti slike koje je potrebno obraditi. Sve te funkcionalnosti dostupne su unutar OpenCV biblioteke, točnije, koristit će se funkcija *readNetFromCaffe()* s parametrima koji su varijable u koje su spremljene putanje do dvije datoteke s prethodno treniranim modelom kojeg smo skinuli s GitHuba u prijašnjim koracima (slika 17).

```
# loading the pretrained model
net = cv2.dnn.readNetFromCaffe(prototxt_path, model_path)
```

Slika 17: Uvoz prethodno treniranog modela u metodu

Kada smo uveli neuronsku mrežu, potrebno je još željenu sliku uvesti u model da mogli vidjeti rezultat detekcije. Slike je potrebno pripremiti po ovim koracima:

- Uvoz slika
- Promijeniti veličinu slike da ju neuronska mreža može koristiti (300x300 piksela)
- Detekcija objekata na slici
- Ponovna promjena veličine
- Prikaz rezultata (pravokutnika) na slici

Dio koda vezan za pripremu slika prikazan je na slici 18.

```
# resizing the image to fit the pretrained model standards
# comment this first line for live detection
image = cv2.imread(image_path)
height, width = image.shape[0], image.shape[1]
blob = cv2.dnn.blobFromImage(cv2.resize(image, (300, 300)), 0.007, (300, 300), 130)

net.setInput(blob)
detected_objects = net.forward()
```

Slika 18: Priprema slika za detekciju

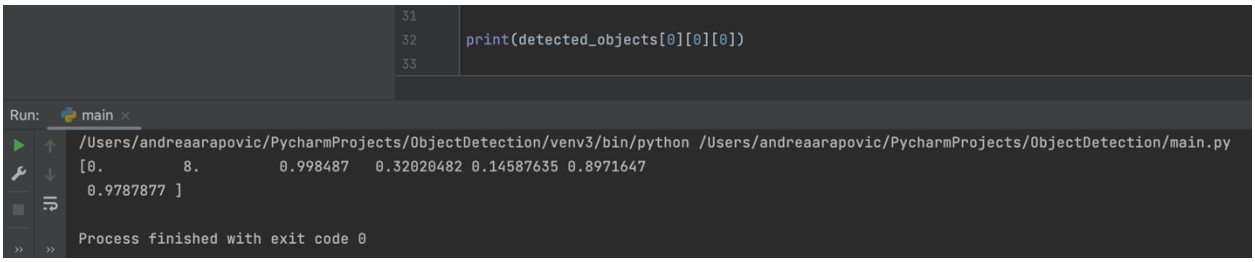
Sliku uvodimo pomoću *imread()* funkcije kojoj dodajemo varijablu u kojoj je spremljena putanja do željene slike. Zatim u varijable *height* i *width* spremamo trenutni oblik i veličinu slike koju smo odabrali. Sljedeće je potrebno definirati blob kojim ćemo odabranu sliku formatirati u veličini koja nam je potrebna za detekciju, točnije, 300x300 piksela za neuronsku mrežu.

Blob (eng. *Binary Large Object*) ili BLOB je zbirka binarnih podataka pohranjenih kao jedna cjelina. Blobovi su obično slike, audio ili drugi multimedijski objekti, iako se ponekad binarni izvršni kod pohranjuje kao blob. Mogu postojati kao postojane vrijednosti unutar nekih baza podataka ili kao programske varijable u nekim programskim jezicima. Ne smije se brkati s binarnom datotekom pohranjenom u datotečnom sustavu.

Sliku kojoj smo promijenili veličinu spremamo u *blob* varijablu pomoću funkcije *blobFromImage()*. Kao sliku koja će se koristiti za detekciju uzimamo onu spremljenu u *blob* varijabli i prenosimo ju modelu spremljenom u *net* varijabli da bi se detekcija mogla izvršiti i to

radimo pomoću funkcija *setInput()* i *forward()* i spremamo rezultate detekcije u *detected_objects* varijablu.

Budući da je proces detekcije gotov, možemo isprintati detektirane objekte u obliku polja i dobiti ćemo nekoliko brojeva koji na prvu nemaju puno smisla, no svaki taj broj ima značenje (slika 19).



Slika 19: Detekcija prikazana u obliku polja

Na slici vidimo informacije za prvi objekt koji je detektiran. Broj 8 označava da je to osmi objekt u classes listi koju smo definirali na početku. Sljedeći broj označava postotak sigurnosti modela da je taj objekt detektiran na slici te svi ostali brojevi označavaju koordinate objekta na slici (prvi broj – gornja lijeva x koordinata, drugi broj – gornja lijeva y koordinata, treći broj – donja desna x koordinata, četvrti broj – donja desna y koordinata). Sve te vrijednosti su normalizirane što znači da ih možemo pomnožiti s veličinom i širinom slike i dobit ćemo prave koordinate piksela na slici. Sve ove informacije zapravo nam pomažu da nacrtamo pravokutnike oko detektiranih objekata (slika 20).

```
for i in range(detected_objects.shape[2]):
    confidence = detected_objects[0][0][i][2]

    if confidence > min_confidence:
        class_index = int(detected_objects[0][0][i][1])
        upper_left_x = int(detected_objects[0][0][i][3] * width)
        upper_left_y = int(detected_objects[0][0][i][4] * height)
        lower_right_x = int(detected_objects[0][0][i][5] * width)
        lower_right_y = int(detected_objects[0][0][i][6] * height)
        prediction_text = f"{classNames[class_index]}: {confidence: .2f}%"
        cv2.rectangle(image, (upper_left_x, upper_left_y), (lower_right_x, lower_right_y), colors[class_index], 3)
        cv2.putText(image, prediction_text, (upper_left_x,
                    upper_left_y - 15 if upper_left_y > 30 else upper_left_y + 15),
                    cv2.FONT_HERSHEY_COMPLEX, 0.6, colors[class_index], 2)
cv2.imshow("Detected objects", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Slika 20: Izrada pravokutnika oko detektiranih objekata

Koristimo postotak sigurnosti kao uvjet unutar for petlje. Za početak moramo postaviti uvjet da je postotak sigurnosti veći od minimalnog kojeg smo postavili na početku. U sljedećim recima koda specificiramo koordinate kutova detektiranog objekta po pravilima koje smo prethodno uočili u prikazanom polju.

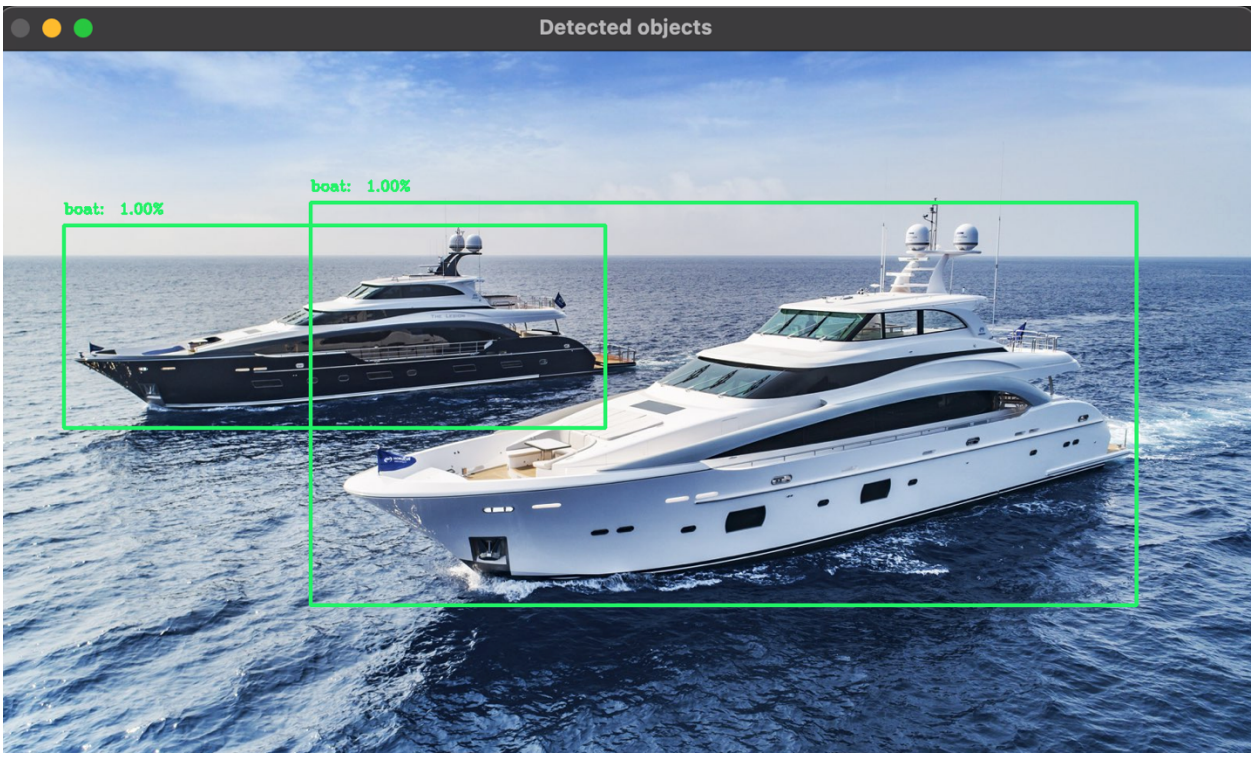
Formatirat ćemo i predikcijski tekst pomoću imena klase zajedno s postotkom sigurnosti modela. Pomoću OpenCV biblioteke prikazat ćemo pravokutnik u boji koju smo ranije pripremili i pozicionirati predikcijski tekst na mjesto koje želimo i u fontu kojeg želimo.

Posljednja stvar koja je preostala je prikazati detektiranu sliku i to radimo pomoću *imshow()* funkcije.

5.5. Prikaz rezultata

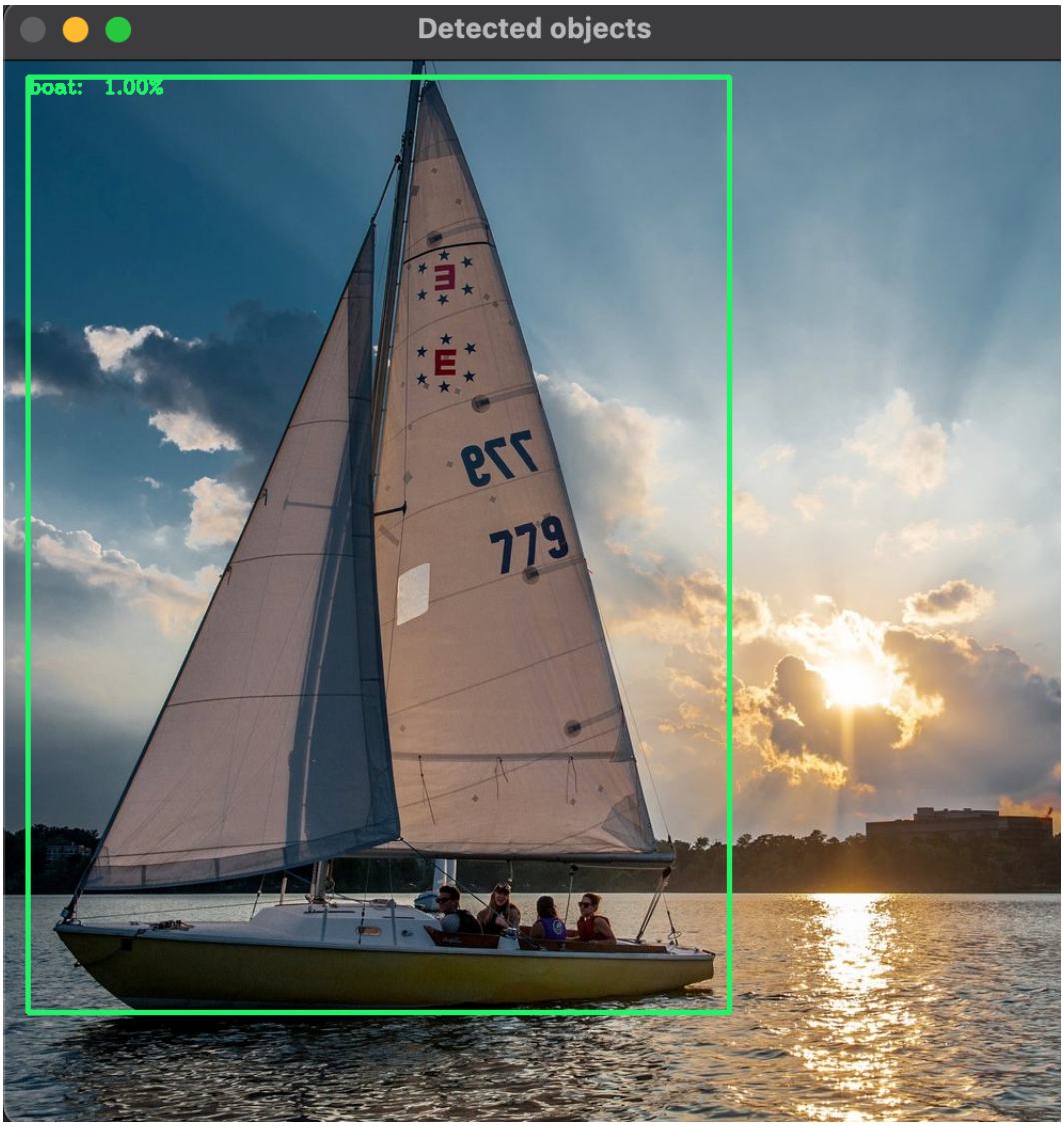
Kao testne primjerke pripremljene su 4 slike. Na dvije slike nalaze se samo brodovi, dok su na druge dvije i ostali objekti kako bi se bolje testirao model i njegova funkcionalnost i točnost.

Na slikama 21, 22, 23 i 24 prikazani su rezultati detekcije objekata.



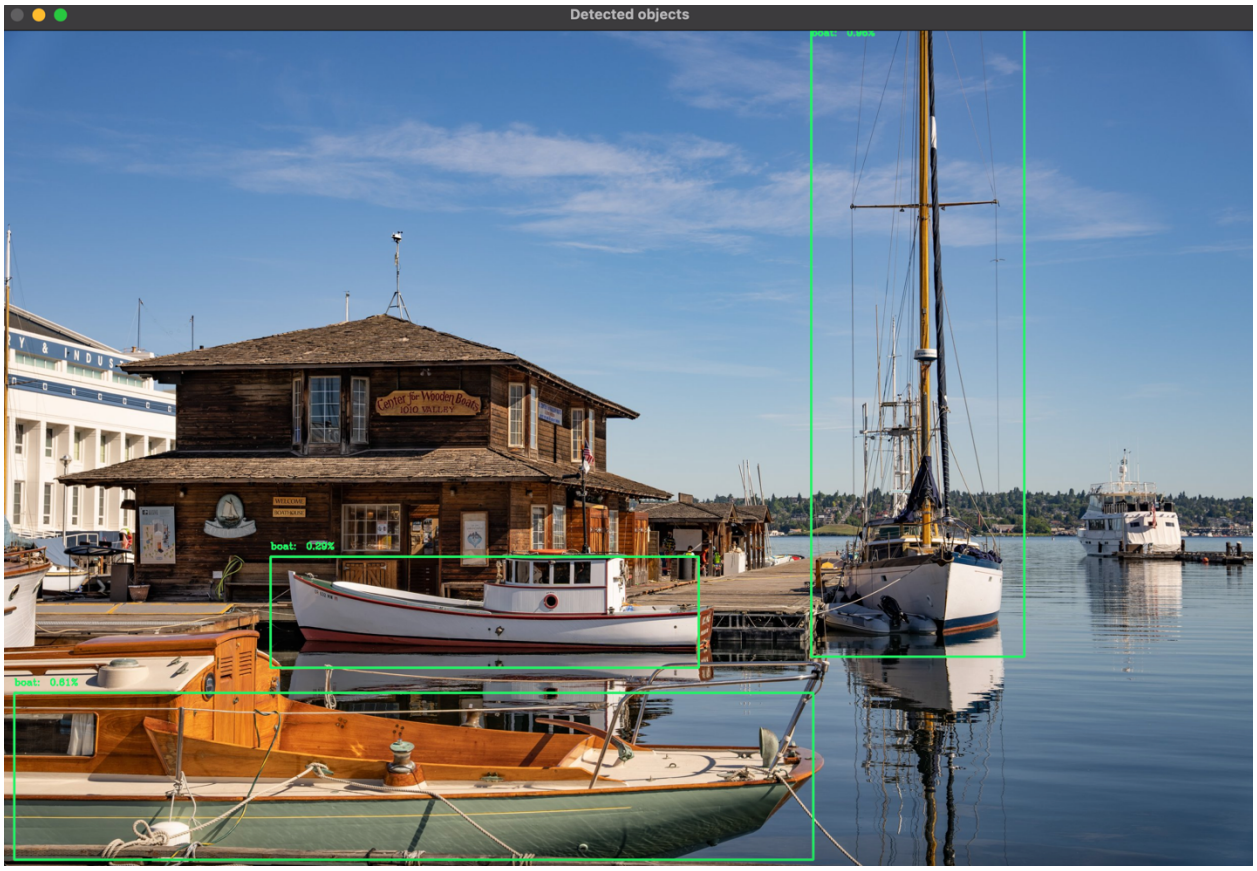
Slika 21: Rezultat detekcije objekata na prvoj slici

Na slici je vidljivo kako je točnost modela visoka i kako sa sto postotnom sigurnošću može detektirati brod na slici. Isti slučaj je i na sljedećoj testnoj slici.



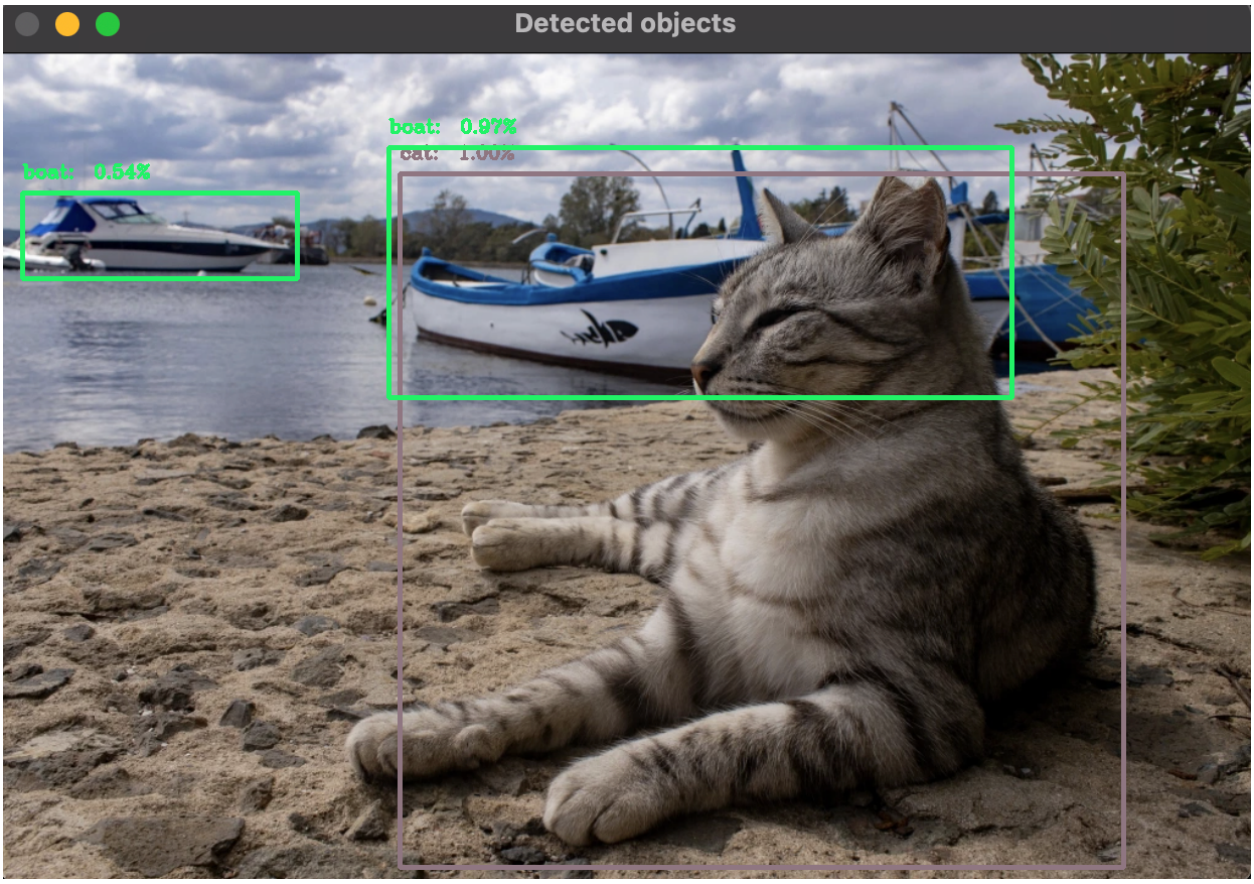
Slika 22: Rezultat detekcije objekata na drugoj slici

Sljedeće dvije testne slike primjer su onih koje imaju više objekata, odnosno, na slikama nisu samo brodovi.



Slika 23: Rezultat detekcije objekata na trećoj slici

Model uspješno detektira one objekte slične onima koje je imao u skupu za učenje, neovisno o tome koliko je objekata na slici. No, kao što je i pretpostavljeno, problem mu stvaraju objekti koji se međusobno prekrivaju, nisu u prvom planu tj. manjih su dimenzija na slici, nalaze se u sjeni ili nisu na slici u cijelosti (slika 23). Takav slučaj može se također primijetiti i na slici 24 gdje je u prvom planu objekt čiji je postotak sigurnosti modela 100%, dok su ostali objekti više udaljeni i postotak se smanjuje.



Slika 24: Rezultat detekcije objekata na četvrtoj slici

6. Modificiranje metode za detekciju pomoću web kamere

Ova metoda ima sposobnost izvršavati detekciju i u real-time okruženju, odnosno, preko web kamere. Potrebno je samo napraviti nekoliko promjena u samom kodu da bi takav model proradio.

Za početak treba komentirati unos statičke slike u model i umjesto toga koristiti *VideoCapture()* funkciju da bi se koristila kamera računala (slika 25). Također dio koda koji je vezan za pripremu slike treba staviti u *while* petlju s uvjetom da se petlja ponavlja neprestano dok se očitavaju podaci iz kamere.

```
# comment this first line for live detection
# image = cv2.imread(image_path)

cap = cv2.VideoCapture(0)

while True:

    _, image = cap.read()

    height, width = image.shape[0], image.shape[1]
    blob = cv2.dnn.blobFromImage(cv2.resize(image, (300, 300)), 0.007, (300, 300), 130)

    net.setInput(blob)
    detected_objects = net.forward()
    # print(detected_objects[0][0][0])

    for i in range(detected_objects.shape[2]):
        confidence = detected_objects[0][0][i][2]

        if confidence > min_confidence:
```

Slika 25: Promjene u kodu za live detekciju

Na kraju je još potrebno nadodati gašenje kamere pri prestanku rada, a ostatak koda ostaje isti kao i na detekciji na statičkim slikama (slika 24).

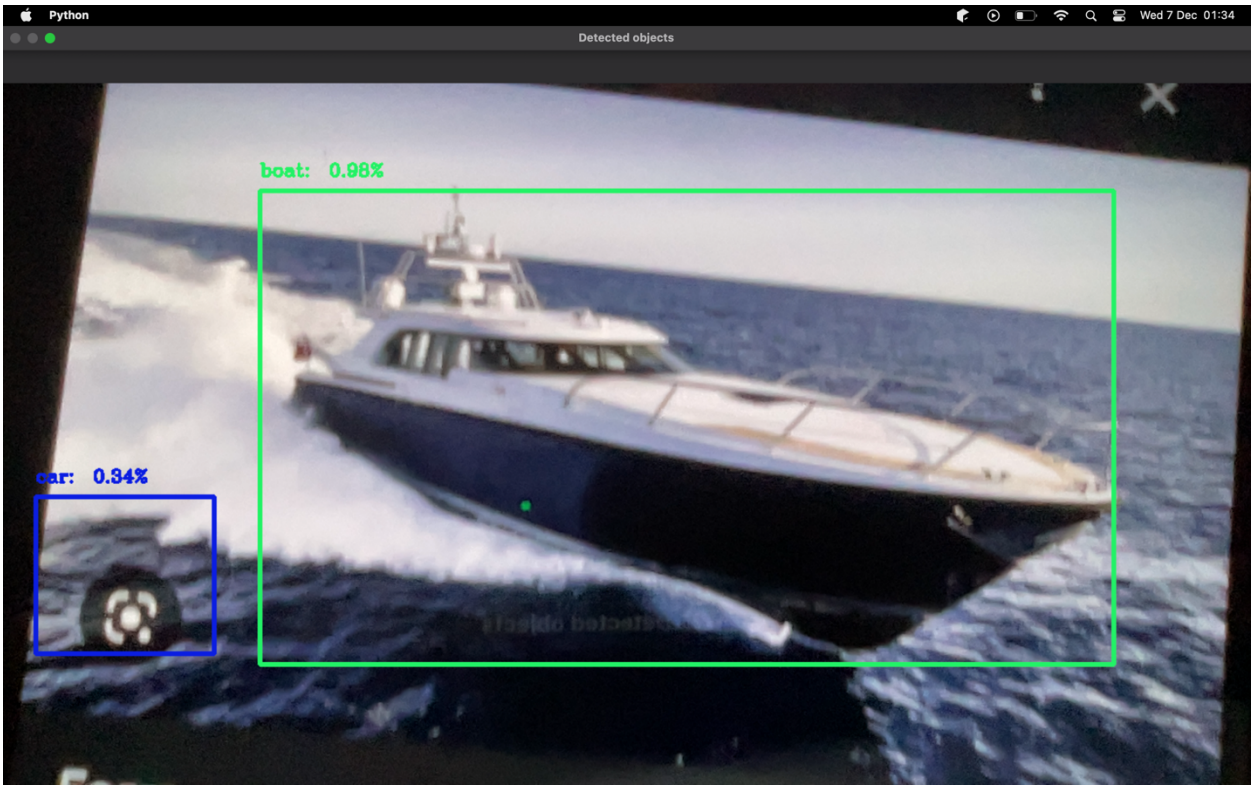
```
            cv2.putText(image, prediction_text, (upper_left_x,
                                                 upper_left_y - 15 if upper_left_y >
                        cv2.FONT_HERSHEY_COMPLEX, 0.6, colors[class_index], 2)
    cv2.imshow("Detected objects", image)
    cv2.waitKey(8)

cv2.destroyAllWindows()
cap.release()
```

Slika 26: Dodavanje gašenja kamere na kraju koda

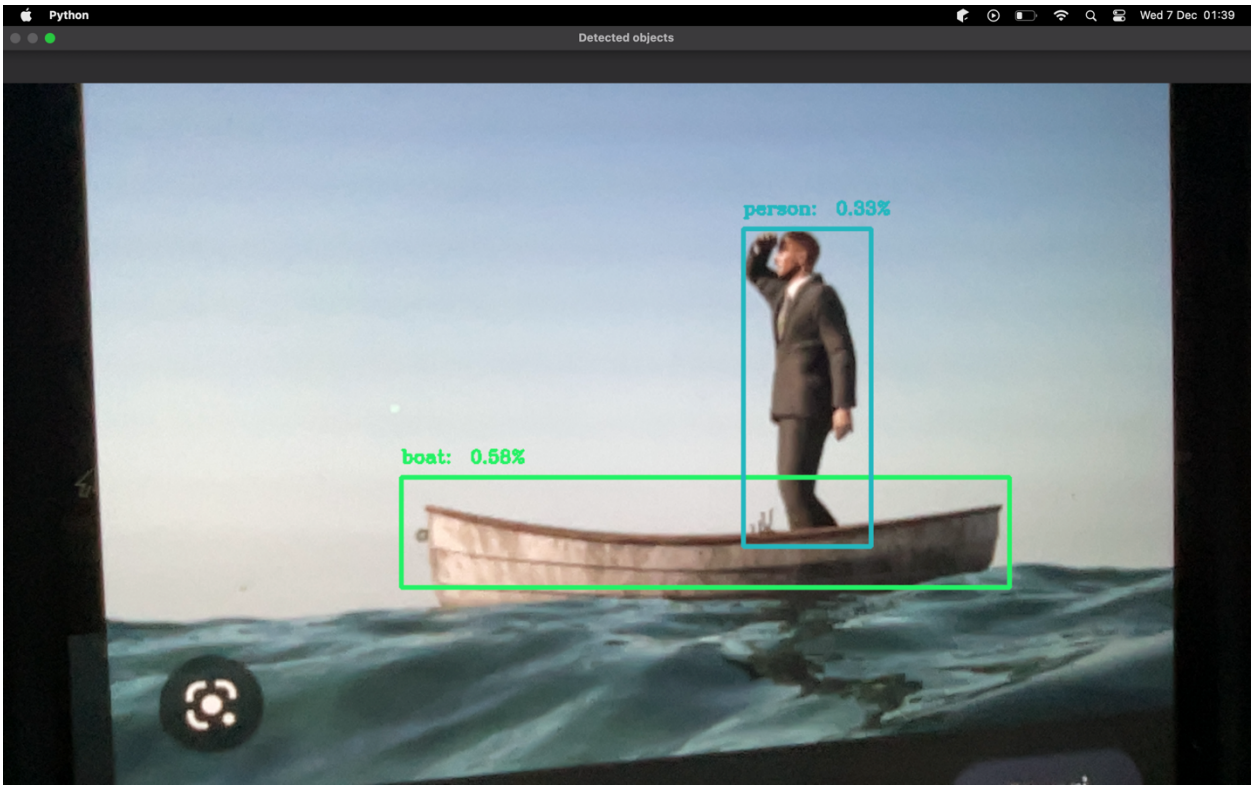
Na slikama 27, 28 i 29 nalaze se primjeri dobiveni detekcijom preko web kamere.

Slika 27: Rezultati detekcije preko web kamere na prvoj slici



Slika 28: Rezultati detekcije preko web kamere na drugoj slici

Na drugom primjeru vidimo smanjenu točnost modela i grešku koju je napravio gdje je znak na slici interpretirao kao objekt auto.



Slika 29: Rezultati detekcije preko web kamere na trećoj slici

Na posljednjem primjeru vidimo onu smanjenu sigurnost koju smo primijetili i na statičkim slikama, ali sve u svemu model dobro radi i preko kamere.

6. Zaključak

Kao ljudi, uvijek smo bili fascinirani tehnološkim promjenama i fikcijom, a upravo sada živimo usred najvećeg napretka u našoj povijesti. Umjetna inteligencija se pokazala kao sljedeća velika stvar na polju tehnologije. Organizacije diljem svijeta smišljaju revolucionarne inovacije u umjetnoj inteligenciji i strojnom učenju. Umjetna inteligencija ne samo da utječe na budućnost svake industrije i svakog čovjeka, već je također djelovala kao glavni pokretač novih tehnologija poput robotike i iOT-a. S obzirom na stopu rasta, nastavit će djelovati kao tehnološki inovator u doglednoj budućnosti. Dakle, postoje goleme mogućnosti za obučene i certificirane stručnjake da započnu karijeru koja će biti isplativa. Kako ove tehnologije budu nastavile rasti, one će imati sve veći utjecaj na društveno okruženje i kvalitetu života.

Kroz rad vidljivo je prethodno trenirani model ima još mogućnosti i prostora za poboljšanje. Postoji i mogućnost razvoja vlastitog modela koji može biti točniji ukoliko se pravilno trenira s velikom količinom podataka. Unatoč tim sitnijim greškama, model ima dosta dobru točnost koja je veća kada na slici ima manje prisutnih objekata.

Korištenje umjetne inteligencije dolazi s mnogo benefita i vidljiv je pomak prema takvoj vrsti tehnologije u velikom broju sektora ljudske djelatnosti, što pokazuje veliki trend modernizacije koji je trenutno u svijetu. Konkretno, računalni vid i detekcija objekata na slici imaju puno pozitivnih aspekata, posebice u medicini gdje može pomoći u ranom pronalaženju zloćudnih bolesti i prestanka širenja takvog stanja na vrijeme i u konačnici, spašavanja života pacijenata.

Literatura

1. Hrga, Milan: "*Računalni vid",* Zbornik radova Veleučilišta u Šibeniku, Vol. No. 1-2/2018, 2018.

Web izvori:

2. https://www.analyticsinsight.net/whats-the-difference-between-ai-and-computer-vision/ - What's the difference between AI and Computer Vision? – zadnje posjećeno 05. prosinca 2022.

3. https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence - What is artificial intelligence (AI)? – zadnje posjećeno 05. prosinca 2022.

4. https://www.expert.ai/blog/machine-learning-definition/ - What Is Machine Learning? A Definition. – zadnje posjećeno 05. prosinca 2022.

5. https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/ - What is Computer Vision and how does it work? An Introduction. – zadnje posjećeno 05. prosinca 2022.

6. https://kili-technology.com/blog/computer-vision-and-machine-learning-differences - Computer Vision vs. Machine Learning: What are the Differences? – zadnje posjećeno 05. prosinca 2022.

7. https://nexocode.com/blog/posts/ai-in-maritime-artificial-intelligence-solutions-in-the-shipping-sector/ - AI in Maritime Industry: How Artificial Intelligence Solutions Benefit the Shipping Sector – zadnje posjećeno 05. prosinca 2022.

8. https://www.fritz.ai/object-detection/ - Object Detection Guide – zadnje posjećeno 05. prosinca 2022.

9. https://medium.com/analytics-vidhya/image-classification-vs-object-detection-vs-image-segmentation-f36db85fe81 - Image Classification vs. Object Detection vs. Image Segmentation – zadnje posjećeno 05. prosinca 2022.

10. https://opencv.org/about/ - OpenCV (About) – zadnje posjećeno 05. prosinca 2022.

11. https://pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/ - Object Detection with Deep Learning and OpenCV – zadnje posjećeno 05. prosinca 2022.

12. https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e - Everything you ever wanted to know about Computer Vision – zadnje posjećeno 12. prosinca 2022.

13. https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852 - Review of Deep Learning Algorithms for Object Detection – zadnje posjećeno 12. prosinca 2022.

14. https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html - Artificial Intelligence History – zadnje posjećeno 12. prosinca 2022.

15. https://arxiv.org/pdf/1512.02325.pdf - Single Shot MultiBox Detector – zadnje posjećeno 12. prosinca 2022.

Popis slika: