Camventory

Raziskovalna naloga

Avtorji: Luka Jurhar, Ažbe Dolinšek, Emil Ibrić

Mentor: Boštjan Lubej, dipl. inž.

Mestna občina Celje, Mladi za Celje

Celje, 2023

IZJAVA*

Mentor/-ica Boštjan Lubej v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Camventory, katere avtorji so Luka Jurhar, Ažbe Dolinšek in Emil Ibrić:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 09.04.2024

ŠOLSKI CENTER CELJE Žig šole CELJE, Pot na Lavo 22

Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

Zahvala

Zahvaljujemo se vsem, ki so kakorkoli pomagali pri izdelavi naše raziskovalne naloge. Najprej bi se zahvalili našemu mentorju, prof. Boštjanu Lubeju, ki nam je pomagal skozi celotno raziskovalno nalogo, nas spodbudil in nas spravil k temu, da smo sploh začeli in nam vedno bil pripravljen pomagati ne glede na trud in čas, ki bi ga bil moral vložiti za raziskovanje naše raziskovalne naloge. Prav tako bi se zahvalili tudi ravnateljici prof. Dr. Aniti Laznik, ki spodbuja nastajanje raziskovalnih nalog na šoli

Povzetek

V raziskovalni nalogi smo raziskali področje zaznavanja objektov. Razložili smo pojem zaznavanje objektov in predstavili, kje se uporablja. Glavni namen raziskave je razvoj programa Camventory. Ta program uporabniku pomaga pri lažji in hitrejši izvedbi inventure. Cilj naloge je razviti program, ki s pomočjo zaznavanja objektov uspešno prepoznava objekte, ki jih želimo šteti pri inventuri. V našem primeru je to računalniška periferija (tipkovnice, miške, monitorji). Verjamemo, da bi lahko takšen program pripomogel k izboljšanju procesa inventure v šolah in podjetjih. Naši rezultati so pokazali prav to, da je možno zaznati in prešteti objekte v zelo kratkem času. Upamo, da bo ta raziskava pripomogla k nadaljnjemu raziskovanju optimizacije procesa inventure.

Ključne besede: Zaznavanje objektov, inventura, objektno programiranje, YOLO, Python

Abstract

In this research paper we have explored the field of object detection. We have explained the concept of object detection and presented where it is used. The main aim of the research is to develop a program called Camventory. This program helps the user to carry out inventory easier and faster. The aim of the thesis is to develop a program that successfully identifies the objects to be counted in an inventory by means of object detection. In our case, this is computer peripherals (keyboards, mice, monitors). We believe that such a program could help to improve the inventory process in schools and companies. Our results have shown just that, as it is possible to detect and count objects in a very short time. We hope that this research will contribute to further research on optimising the inventory process.

Keywords: Object detection, inventory, object-oriented programming, YOLO, Python

Kazalo vsebine

Kazalo slik

Kazalo grafov

1 UVOD

Inventura je opravilo, s katerim se bo marsikdo srečal tekom svoje poslovne poti. Inventura je proces štetja in beleženja materiala, sredstev in pripomočkov v podjetju ali organizaciji. Je ključen proces pri uspešni organizaciji in posledično uspešnem poslovanju podjetij. V nadaljevanju bomo predstavili problem, ki ga raziskovalna naloga rešuje. Opisali bomo tudi naše cilje in hipoteze, ki smo jih nato potrdili ali zavrgli na podlagi rezultatov raziskovalne naloge.

1.1 Opredelitev problema

Mnogim se inventura zdi težavno in dolgotrajno, ampak vselej pomembno opravilo. Inventura je kljub naporom ključni proces v poslovanju podjetij, ki zahteva natančno in točno zapisovanje sredstev in izdelkov. Odločili smo se raziskati, kako bi lahko s pomočjo prepoznavanja objektov (object detection) olajšali izvajanje inventure in s tem prihranili čas in trud. Naš cilj je bil ustvariti program, ki zazna objekt, v katerega merimo s kamero in ga prišteje. Na ta način bi bilo opravljanje inventure veliko lažje.

Cilji

Cilj raziskovalne naloge je bil ustvariti program, ki bo učinkovito zaznaval računalniško periferijo (monitor, tipkovnica, miška) in jo prišteval ter s tem zanesljivo opravil inventuro. Podatke o številu predmetov bo zapisal v podatkovno bazo, ki bo enostavno dostopna.

1.2 Hipoteze

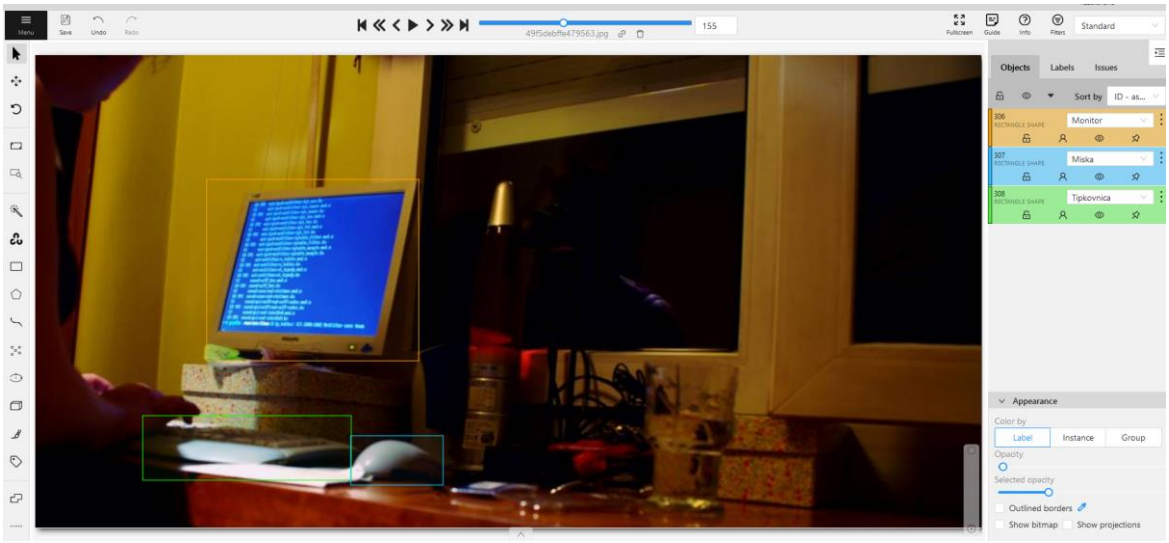
1. HIPOTEZA: Izdelali bomo program, ki bo uspešno zaznal predmete in jih zabeležil v podatkovno bazo.
2. HIPOTEZA: Za zaznavanje predmetov bomo lahko uporabili spletno kamero.
3. HIPOTEZA: Uspešno bomo zaznavali 10 predmetov.

2 Uporabljena programska oprema

V tem poglavju bomo opisali programska orodja in opremo, ki smo jih uporabljali tekom izvajanja te naloge. Vsako orodje posebej predstavlja ključen del pri uspešni izvedbi te naloge.

2.1 CVAT

Computer vision annotation tool (CVAT) je brezplačno odprtokodno spletno orodje za anotacijo slik in videoposnetkov. Podpira glavne naloge strojnega učenja, kot so zaznavanje predmetov, razvrščanje in segmentacijo slik. S tem orodjem smo si pomagali k točni anotaciji slik, kjer smo morali označiti objekte, ki smo jih želeli zaznavati. Orodje nam je močno pomagalo pri zaznavanju objektov, v našem primeru tipkovnic, mišk in monitorjev.



Slika 1: CVAT anotacija (Vir: Lasten iz spletne strani cvat.ai, 2024)

2.2 Pycharm

PyCharm je integrirano razvojno okolje (IDE), ki se uporablja za programiranje v Pythonu. Razvila ga je češka organizacija JetBrains, na voljo pa je za operacijske sisteme Windows, macOS in Linux. PyCharm je na voljo v dveh različicah: Community Edition in Professional Edition. Community Edition je prosto dostopna in odprta, medtem ko je Professional Edition plačljiva naročnina.

Glavne funkcije PyCharm programskega okolja:

Pametno dokončanje kode: PyCharm predstavlja pametno kodo celote, ki vam omogoča hitrejši in pravilnejši zapis kode. Na podlagi konteksta vaše kode lahko zagovarja ključne besede, spremenljivke, funkcije in druge elemente kode.

Pregledi kode: PyCharm lahko vašo kodo pregleda za napake in težave. Zazna lahko sintaktične napake zmogljivosti, tipske napake in druge logične napake.

Odpravljanje napak: PyCharm ponuja učinkovit razhroščevalnik, ki vam omogoča postopno pregledovanje kode po vrsticah, pregledovanje vrednosti spremenljivk in nastavljanje prekinitvenih točk.

Preizkušanje: PyCharm se povezuje z znanimi ogrodji za testiranje Pythona, vključno z unittestom in pytestom. To vam omogoča, da teste zaženete naenkrat iz IDE.

Nadzor različic: PyCharm se povezuje s priljubljenimi strukturami za upravljanje modelov, vključno z Gitom in Mercurialom. Tako lahko spremljate spremembe svoje kode in sodelujete z drugimi razvijalci.

Slika 2: PyCharm logotip (Vir: medium.com, 2024)

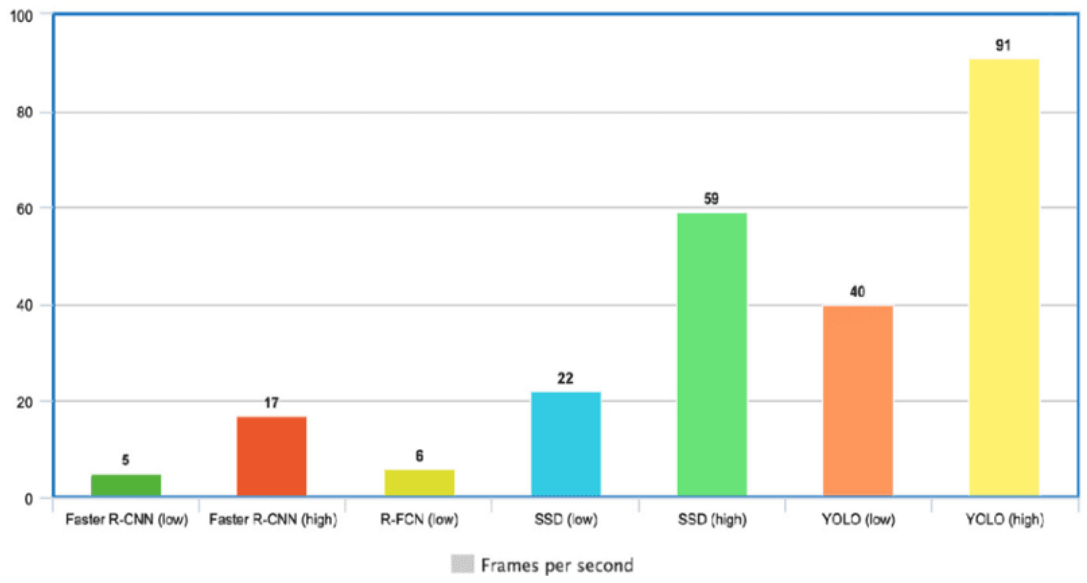
2.3 Pytorch

PyTorch je programsko odprtokodno ogrodje za globoko učenje, ki se uporablja za gradnjo nevronskih omrežij in združuje knjižnico strojnega učenja Torch z visokonivojskim vmesnikom API, ki temelji na jeziku Python. Zaradi svoje prilagodljivosti in enostavne uporabe ter drugih prednosti, je postalo vodilno ogrodje ML za akademske in raziskovalne skupnosti.

Slika 3: PyTorch logotip (Vir: Wikipedija, 2024

2.4 Ultralytics YOLOv8

YOLOv8 je najnovejša različica programa YOLO (You Only Look Once), podjetja ultralytics. Je model za prepoznavo objektov in segmentacijo slik. Zaradi svoje hitrosti in natančnosti je med razvijalci zelo priljubljen. Slike obdeluje s hitrostjo 45 fps, poleg tega pa dosega več kot dvakrat večjo povprečno natančnost (mAP) v primerjavi z ostalimi sistemi za obdelavo objektov.



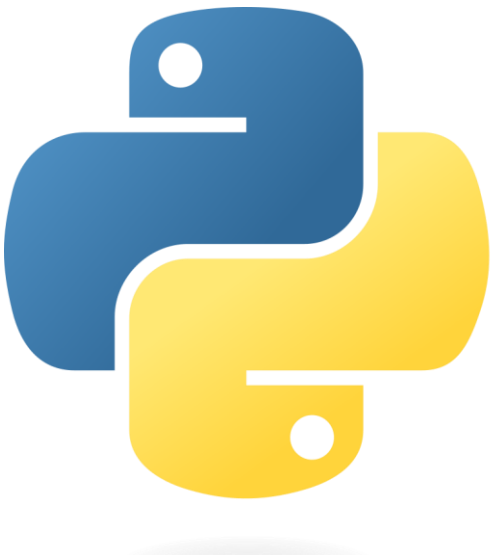
Graf 1: Graf slik na sekundo različnih načinov zaznavanja objektov

3 Uporabljeni programski jeziki

Programski jezik je jezik namenjen komuniciranju človeka z računalnikom. Programski jeziki se uporabljajo predvsem za nadzor delovanja stroja ali za izvajanje algoritmov. Trenutno je implementiranih več tisoč programskih jezikov. Program lahko razdelimo na dve obliki, in sicer sintakso in semantiko. V nadaljevanju bomo opisali temeljni programski jezik in njegove programske knjižnice, ki smo jih uporabili za izdelavo programa.

3.1 Python

Python je programski jezik, ki je interpretiran, objektno usmerjen in velja tudi za jezik visoke ravni. Python je eden najlažjih, a hkrati najbolj uporabnih programskih jezikov, ki se pogosto uporablja v industriji programske opreme. Ljudje uporabljajo Python za konkurenčno programiranje, razvoj spletnih strani in ustvarjanje programske opreme. Zaradi najpreprostejše sintakse je priporočljiv za začetnike, ki šele vstopajo na področje programskega inženirstva. Njegovo povpraševanje zelo hitro narašča zaradi obsežnih primerov uporabe na sodobnih tehnoloških področjih, kot so podatkovna znanost, strojno učenje in naloge avtomatizacije. Že vrsto let se uvršča med najboljše programske jezike.



Slika 4:Python logotip (Vir: Wikipedija, 2024)

3.1.1 SQLAlchemy

SQLAlchemy je odprtokodna zbirka orodij SQL in knjižnica za objektno-relacijsko preslikavo (ORM) za Python. Razvijalcem zagotavlja zmogljiv, a prilagodljiv nabor orodij za delo s podatkovnimi bazami v aplikacijah Python.

SQLAlchemy razvijalcem omogoča preslikavo Pythonovih objektov v tabele podatkovne baze. To omogoča razvijalcem, da s podatkovnimi zbirkami komunicirajo z objekti Pythona, namesto da bi pisali surove poizvedbe SQL. SQLAlchemy skrbi za prevajanje kode Python-a v poizvedbe SQL in obratno, zaradi česar so interakcije s podatkovnimi bazami enostavnejše in hitrejše.

Ena od prednosti programa SQLAlchemy je podpora različnim sistemom podatkovnih baz, vključno s PostgreSQL, MySQL, SQLite, Oracle, Microsoft SQL Server in drugimi. Zagotavlja vmesnik za različne podatkovne baze, kar razvijalcem omogoča pisanje kode, ki ni odvisna od baze. SQLAlchemy prav tako ponuja zmogljiv vmesnik za poizvedbe, ki razvijalcem omogoča izdelavo zapletenih poizvedb SQL z uporabo kode Python. Podpira filtriranje, razvrščanje, grupiranje in združevanje podatkov, kar olajša pridobivanje in manipulacijo podatkov iz podatkovne zbirke.

3.1.2 Flask

Flask je prilagodljivo spletno ogrodje (framework) za Python programski jezik. Zasnovano je tako, da je začetek spletnega razvoja v Pythonu enostaven in preprost, hkrati pa je dovolj zmogljivo za gradnjo kompleksnih spletnih aplikacij.

Flask se pogosto imenuje "mikro" ogrodje, saj je njegovo jedro preprosto. V svoji programski knjižnici nima posebnih knjižnic ali orodij, zato lahko razvijalci sami izberejo komponente, ki jih potrebujejo za svoje projekte. Ena od ključnih značilnosti ogrodja Flask je njegov API za enostavno uporabo za določanje poti, obdelavo zahtevkov in generiranje odgovorov. Za povezovanje poti URL s funkcijami Pythona uporablja dekoratorje, kar olajša ustvarjanje spletnih končnih točk. Modularna zasnova Flaska omogoča enostavno razširjanje in prilagajanje. Na voljo je bogato področje razširitev Flask za naloge, kot so integracija podatkovne zbirke, upravljanje obrazcev, avtentikacija in druge.

4 Zaznavanje objektov

Zaznavanje objektov je ključna naloga računalniškega vida, katere cilj je prepoznati in locirati objekte na slikah ali video zaporedjih. Vključuje prepoznavanje predmetov in njihovo natančno lociranje s pomočjo omejitvenih okvirov. Na začetku se slike predobdelajo, kar lahko vključuje spremembo velikosti ali normalizacijo, da se pripravijo za analizo. Modeli globokega učenja, kot so konvolucijske nevronske mreže (CNN), iz slik izločijo ustrezne lastnosti in zajamejo bistvene vzorce za razlikovanje predmetov. Model nato predvidi omejitvene okvirje, ki obkrožajo predmete in označujejo njihove prostorske lokacije, hkrati pa jih razvrsti v kategorije, kot sta "oseba" ali "avtomobil". Tehnike naknadne obdelave izboljšajo rezultate s filtriranjem lažno pozitivnih rezultatov in tako povečajo natančnost. Zaznavanje predmetov se pogosto uporablja na področjih kot so avtonomna vozila, nadzor, robotika in medicinsko slikanje, saj omogoča strojem učinkovito razumevanje in interakcijo z vizualnim svetom.

4.1 Zaznavanje predmetov / Razvrščanje predmetov

Zaznavanje predmetov in razvrščanje predmetov predstavljata dva ključna pristopa k zaznavanju predmetov v računalniškem svetu, ki imata različne cilje in izhodne rezultate.

Zaznavanje objektov je v osnovi kot detektiv, ki ne le prepozna objekt na sliki, temveč tudi natančno označi, kje se nahaja. V rezultatih zaznavanja objektov tako dobimo razredno oznako, ki pove, kateri predmet je zaznan, in hkrati pravokotnik ali kvadrat, ki precizno opisuje njegovo lokacijo na sliki. To omogoča vpogled v prostorsko razporeditev predmetov na sliki ali v videu in tudi stvarno-časovno zapisovanje zaznanih predmetov.

Po drugi strani pa razvrščanje objektov opravlja bolj osnovno nalogo prepoznavanja. Če bi primerjali z branjem naslovov knjig, bi razvrščanje predmetov samo povedalo, katera vrsta knjige je prisotna na sliki, ne pa tudi, kje se nahaja. Zato je izhod razvrščanja objektov sestavljen iz seznama razrednih oznak, ki identificirajo vrste predmetov, vendar ne vsebujejo informacij o njihovi prostorski postavitvi.

V praksi se zaznavanje objektov pogosto uporablja tam, kjer je ključno vedeti ne le, kateri predmeti so prisotni, temveč tudi, kje točno se nahajajo. Nasprotno pa razvrščanje najdemo v scenarijih, kjer je dovolj le prepoznati vrsto predmetov, ne da bi natančno vedeli, kje se nahajajo na sliki ali v videu. Oba pristopa dopolnjujeta drug drugega, odvisno od specifičnih potreb in izzivov analize vizualnih podatkov.

Zaznavanje objektov je bolj zapleteno, saj zahteva, da računalnik razume različne predmete, njihove velikosti in oblike ter se spopada s situacijami, ko so predmeti delno skriti ali prekriti, razvrščanje pa je bolj osnovno, saj se osredotoča le na prepoznavanje vrste predmetov.

4.2 Kako deluje

Zaznavanje predmetov deluje tako, da natančno pregleda slike ali videoposnetke ter ugotovi in opiše prisotnost predmetov. Postopek poteka v vrsti strukturiranih korakov. Na začetku se vhodna slika ali kader predobdela, da se standardizira njena oblika in izboljša razlagalnost modela. Tehnike, kot so spreminjanje velikosti, normalizacija in pretvorba barvnega prostora, optimizirajo podatke za poznejšo analizo.

Nato algoritmi za zaznavanje predmetov uporabljajo arhitekture globokega učenja, predvsem konvolucijske nevronske mreže (CNN), da iz vhodnih podatkov izluščijo ustrezne lastnosti. Te značilnosti zajemajo bistvene vizualne značilnosti, vključno z vzorci, teksturami in oblikami, ki kažejo na različne predmete v prizoru.

Po ekstrakciji značilnosti se odkrivanje predmetov nadaljuje s predlaganjem kandidatnih območij na sliki, kjer bi se lahko nahajali predmeti. Te regije, ustvarjene z metodami, kot so selektivno iskanje ali mreže za predlaganje regij, služijo kot osrednje točke za nadaljnjo analizo.

Za vsako predlagano regijo CNN izlušči značilnosti, ki se nato prenesejo v klasifikacijsko omrežje. To omrežje vsaki regiji dodeli oceno verjetnosti, ki označuje verjetnost, da vsebuje določen razred predmetov. Hkrati regresija mejnih polj prilagodi predlagana mejna polja, da natančneje zajamejo meje zaznanih predmetov.

Po klasifikaciji in regresiji se izvede ključni korak naknadne obdelave, imenovan nemaksimalno zatiranje. V tem koraku se izločijo podvojena zaznavanja in ohranijo le najbolj zanesljive napovedi, s čimer se zagotovi, da je vsak objekt identificiran natančno enkrat.

Končni rezultat zaznavanja predmetov so koordinate zaznanih mejnih polj, pripadajoče oznake razredov in ocene zaupanja, ki kažejo na gotovost zaznavanja. Modeli za zaznavanje predmetov so usposobljeni na obsežnih zbirkah podatkov, ki vsebujejo slike z opombami, na katerih so predmeti natančno označeni. Med usposabljanjem se model nauči razločevati razrede predmetov in jih natančno locirati na slikah. Nato se lahko usposobljeni model uporabi za analizo novih, še nevidenih slik ali videoposnetkov in učinkovito prepozna predmete v različnih scenarijih iz resničnega sveta.

Algoritmi za zaznavanje predmetov v bistvu omogočajo strojem, da razumejo in dešifrirajo vizualno vsebino, kar omogoča številne aplikacije na področjih, kot so avtonomna vozila, nadzor, robotika in medicinsko slikanje.

4.3 Uporaba zaznave predmetov

Odkrivanje objektov se uporablja na številnih področjih, pri čemer se izkorišča sposobnost prepoznavanja in lociranja objektov v slikah ali video tokovih. Navajamo nekaj pomembnih primerov uporabe:

- Avtonomna vozila: Za avtonomna vozila je zaznavanje in razumevanje okolice ključnega pomena. Pomaga pri prepoznavanju pešcev, vozil, prometnih znakov in ovir, kar vozilu omogoča, da v realnem času sprejema utemeljene odločitve za zagotavljanje varne navigacije.
- Nadzor in varnost: V nadzornih sistemih zaznavanje predmetov pomaga pri spremljanju in analiziranju videoposnetkov za odkrivanje sumljivih dejavnosti, vsiljivcev ali zanimivih predmetov. Omogoča avtomatiziran nadzor, kar zmanjšuje potrebo po človeškem posredovanju in izboljšuje odzivni čas na morebitne grožnje.
- Analitika maloprodaje: Trgovci na drobno uporabljajo zaznavanje predmetov za sledenje gibanja strank, analizo postavitve trgovin in spremljanje postavitve izdelkov. Pomaga pri optimizaciji postavitev trgovin, analiziranju vedenja strank in izboljšanju postavitve izdelkov za boljšo prodajo in zadovoljstvo strank.
- Medicinsko slikanje: Pri medicinskem slikanju zaznavanje predmetov pomaga pri prepoznavanju in lociranju anatomskih struktur, tumorjev, nepravilnosti in drugih zdravstvenih stanj z rentgenskimi žarki, magnetno resonanco,

računalniško tomografijo in drugimi načini slikanja. Radiologom in zdravstvenim delavcem pomaga pri diagnosticiranju in načrtovanju zdravljenja.

- Industrijska avtomatizacija: Zaznavanje objektov se v industrijski avtomatizaciji uporablja za nadzor kakovosti, odkrivanje napak in upravljanje zalog. Pomaga pri prepoznavanju izdelkov z napako na proizvodnih linijah, sledenju ravni zalog in optimizaciji logističnih procesov.
- Razširjena resničnost in igre: Zaznavanje predmetov omogoča aplikacije razširjene resničnosti in igralne izkušnje s prepoznavanjem predmetov, gest in gibov v resničnem svetu. Omogoča interaktivne izkušnje in potopljivo igranje iger ter briše meje med fizičnim in virtualnim svetom.
- Spremljanje okolja: Zaznavanje predmetov pomaga pri spremljanju okolja z identifikacijo in sledenjem divjih živali, spremljanjem krčenja gozdov in preučevanjem biotske raznovrstnosti. Raziskovalcem in naravovarstvenikom pomaga pri prizadevanjih za ohranjanje divjih živali in pri upravljanju okolja.

To je le nekaj primerov široke uporabe zaznavanja predmetov, ki poudarjajo njegov pomen v različnih panogah in področjih. Ker tehnologija še naprej napreduje, lahko pričakujemo, da se bo v prihodnosti pojavilo še več inovativnih aplikacij.

4.4 Metode zaznavanja predmetov

4.4.1 Viola-Jones detektor

Leta 2001 je bil predlagan Viola-Jonesov detektor za reševanje problema zaznavanja obrazov v računalniškem vidu, pri čemer je bila težava v tem, da računalnik za izvedbo naloge potrebuje natančna navodila in omejitve. Njihov detektor je temeljil na metodi ekstrakcije in klasifikacije značilk, ki jim je omogočila vestno analizo slike, da bi ugotovili ali je na kateri od njih človeški obraz. Da bi to dosegli, je moral algoritem iti skozi štiri stopnje:

Izbiranje značilnosti, podobnih Haarjevim - črna in bela pravokotna polja, združena v takšnem razmerju, da predstavljajo posebne značilnosti človeškega obraza, na primer svetlejši zgornji del ličnic in temnejše predele oči.

Ustvarjanje celostne slike, da se oceni, kateri deli slike imajo največjo navzkrižno korelacijo z značilnostjo.

Usposabljanje AdaBoost: izbira najboljših lastnosti in usposobljenih klasifikatorjev, ki jih uporabljajo.

Kaskadni klasifikatorji: odgovorni za filtriranje podoken, ki vsebujejo predmete.

4.4.2 Histogram usmerjenih gradientov

Drug deskriptor značilnosti, imenovan histogram usmerjenih gradientov (HOG), je postal priljubljen po konferenci o računalniškem vidu in prepoznavanju vzorcev, ki je potekala leta 2005.

Ta metoda, ki se večinoma uporablja v samovozečih avtomobilih in drugih aplikacijah, ki zahtevajo zaznavanje ljudi, poskuša izluščiti kontrast v različnih območjih slike na podlagi zamisli, da je obliko predmeta mogoče opredeliti z dolžino in gostoto vektorjev gradientov.

Po neobvezni normalizaciji gama se vhodna slika razdeli na gosto mrežo, ki jo sestavljajo povezane celice, in za vsako piko se izračunata velikost in kot gradienta. Nato se za vsako celico ustvari histogram usmerjenosti, ki se stehta z velikostjo gradienta ali njegovo obrezano različico.

Lokalna normalizacija slike omogoča normalizacijo kontrasta, kar posledično izboljša natančnost. Vse celice, združene v bloke, se lahko izločijo kot tako imenovani elementi HOG in uporabijo v stroju s podpornimi vektorji (SVM) ali katerem koli drugem algoritmu strojnega učenja za učenje na dveh podatkovnih nizih - slikah, ki vsebujejo predmet zanimanja, in tistih, ki ga ne vsebujejo. Deskriptor lastnosti HOG se lahko uporablja za odkrivanje različnih predmetov, čeprav je najprimernejši za odkrivanje ljudi.

4.4.3 Deep learning

4.4.3.1 AlexNet

Po tehnološkem napredku so se začele pojavljati velike zbirke podatkov za boljše usposabljanje inovativnih algoritmov in modelov, ki se osredotočajo na umetno inteligenco. Da bi izboljšali njihovo uspešnost, so skrbniki enega od naborov podatkov,

imenovanega ImageNet, začeli izvajati letno tekmovanje programske opreme ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

Po pričakovanjih je tekmovanje preseglo pričakovanja ustvarjalcev in njegov zmagovalec leta 2012, klasifikacijski model konvolucijske nevronske mreže (CNN) AlexNet je zaznamoval začetek sodobne zgodovine prepoznavanja predmetov. To je bila implementacija hitre grafične procesne enote CNN, ki temelji na stari strukturi LeNet, v kombinaciji s povečanjem podatkov, ki je dosegla najnižjo stopnjo napak. Uspeh je povzročil, da je AlexNet pomladil zanimanje za nevronske mreže in postal najvplivnejša inovacija na področju računalniškega vida v tistem času, kar je pomenilo začetek revolucije globokega učenja.

4.4.3.2 Regionalni CNN

Prelomnica za tehnologije zaznavanja objektov je bila, ko je bila v to področje vključena regionalna konvolucijska nevronska mreža (R-CNN). Predlagani model je temeljil na algoritmu selektivnega iskanja, ki je izločil okoli 2000 predlogov regij (omejevalnih okvirjev, ki lahko vsebujejo predmet), ki so bili nato preoblikovani in posredovani CNN v analizo. Funkcije, ekstrahirane v tem procesu, so bile uporabljene za razvrščanje predlogov regij, omejevalni okvirji pa so bili izboljšani z uporabo regresije omejevalnega okvirja. Ta pristop je bil natančen, a počasen in ga ni bilo mogoče implementirati v realnem času, zato so njegovi nasledniki hitro prevzeli njegovo mesto.

Različici R-CNN Fast R-CNN in Faster R-CNN sta poskušali rešiti težave svojih predhodnikov. Z izvedbo konvolucijske operacije samo enkrat na celotno vhodno sliko so iz nje izluščili zemljevid funkcij, kar je opazno povečalo zmogljivost modela. Poleg tega je Faster R-CNN popolnoma odpravil zamuden algoritem selektivnega iskanja, tako da ga je nadomestil z ločenim omrežjem za napovedovanje regionalnih predlogov. Ta odločitev je znatno izboljšala preskusni čas modela in omogočila njegovo implementacijo v nalogah odkrivanja objektov v realnem času.

4.4.3.3 You Only Look Once

Približno v istem času se je pojavil algoritem You Only Look Once (YOLO). Kot že ime pove, se v tem algoritmu za napovedovanje razredov in omejevalnih okvirjev v eni oceni uporablja eno samo konvolucijsko omrežje. Vhodna slika je razdeljena na kvadratne celice in vsaka od njih je odgovorna za predvidevanje določenega števila mejnih okvirjev. Škatle, za katere obstaja majhna verjetnost, da so predmeti, se odstranijo, medtem ko se tiste, za katere obstaja velika verjetnost, da vsebujejo predmet, izboljšajo, tako da se tesno prilegajo predmetom, ki nas zanimajo.

Odvisno od modela YOLO obdeluje slike s hitrostjo 45 sličic na sekundo in naredi manj lažno pozitivnih napovedi v primerjavi z drugimi algoritmi, kar ga uvršča v aplikacije v realnem času. Vendar pa lahko zaradi močnih prostorskih omejitev pri predvidevanjih omejevalnih okvirjev zazna omejeno število bližnjih predmetov, majhni predmeti, ki se pojavljajo v skupinah, pa prav tako povzročajo odstopanja v napovedih. Ni treba posebej poudarjati, da so njegove posodobljene različice, kot so YOLOv2 ter YOLOv3 in YOLOv4, izboljšale splošno natančnost zaznavanja in bile boljše pri zaznavanju predmetov majhnega obsega.

YOLOv1 (You Only Look Once v1): YOLOv1 je bil prvi model v družini YOLO algoritmov, predstavljen leta 2016. Ta model je postal prepoznaven zaradi svoje revolucionarne ideje, da se celotna slika obravnava kot celota pri detekciji objektov. To pomeni, da YOLOv1 hkrati napove verjetnosti različnih razredov objektov in njihove omejitvene okvirje na sliki. Kljub hitrosti in preprostosti je bil manj natančen v primerjavi z nekaterimi kasnejšimi različicami. Verjetnost pri YOLO sistemu predstavlja stopnjo prepričanja algoritma o prisotnosti določenega razreda na sliki.

YOLOv2 (YOLO9000): YOLOv2, znan tudi kot YOLO9000, je bil izdan leta 2017. Glavna izboljšava je bila integracija zmožnosti detekcije več tisoč različnih razredov objektov, kar je omogočalo prepoznavo širokega spektra predmetov (You Only Look Once for 9000 Object Categories, zato tudi ime YOLO9000). YOLO9000 je prav tako prinesel izboljšave v hitrosti in natančnosti ter bil korak naprej v razvoju YOLO algoritmov.

YOLOv3: YOLOv3, izdan leta 2018, je prinesel dodatne izboljšave. Arhitektura je postala bolj zahtevna s povečanjem števila plasti. Uporaba "anchor boxov" je dodatno izboljšala prilagajanje različnim oblikam objektov. YOLOv3 je dosegel boljšo

natančnost kot prejšnje različice, hkrati pa je ohranil hitrost, zaradi česa je bil v praksi zelo uporaben za detekcijo objektov v realnem času.

YOLOv4: YOLOv4, ki je bil predstavljen leta 2020, prinaša nadaljnje izboljšave v hitrosti in natančnosti. Uporablja napredne tehnologije, kot sta CSPDarknet53 (Cross-Stage Partial Networks) in PANet (Path Aggregation Network), kar pripomore k boljši stabilnosti pri delu z velikimi podatkovnimi seti. YOLOv4 ostaja ena najnatančnejših in najhitrejših različic algoritma.

YOLOv5: YOLOv5, je sicer bil izdan od drugega proizvajalca in je nadgradil YOLOv4 z razliko v hitrosti. Največja sprememba pa je bila uporaba PyCharm okolja namesto Dark Net Okolja.

YOLOv7: YOLOv7, je bil izdan leta 2022, v juliju. Ta nov sistem je presegel prejšnje limite hitrosti in natančnosti za objekte, od 5 FPSov vse do 160 FPSov. Njegova arhitektura temelji na treh novih funkcijah. E-ELAN, za učinkovito učenje, skaliranje modelov različnih velikosti in “bag-of-freebies" pristop za natančnost in učinkovitost.

YOLOv8: YOLOv8, ki smo ga uporabili tudi mi, je bil izdan leta 2023. Trenutno ne vemo točne arhitekture YOLOv8, tako da se moramo zanašati na prejšnje verzije in približno ugibati. YOLOv8 je Anchor-Free, predvidi manj kvadratov in ima hitrejši NMS proces. Med učenjem YOLOv8 uporablja mosaic augmentacijo, ampak ne do konca. Na zadnjih desetih epochih je ta onemogočen, saj je lahko škodljiv, če se uporabi skozi celoten proces učenja.

5 Potek dela

5.1 Prenos slik za anotacijo

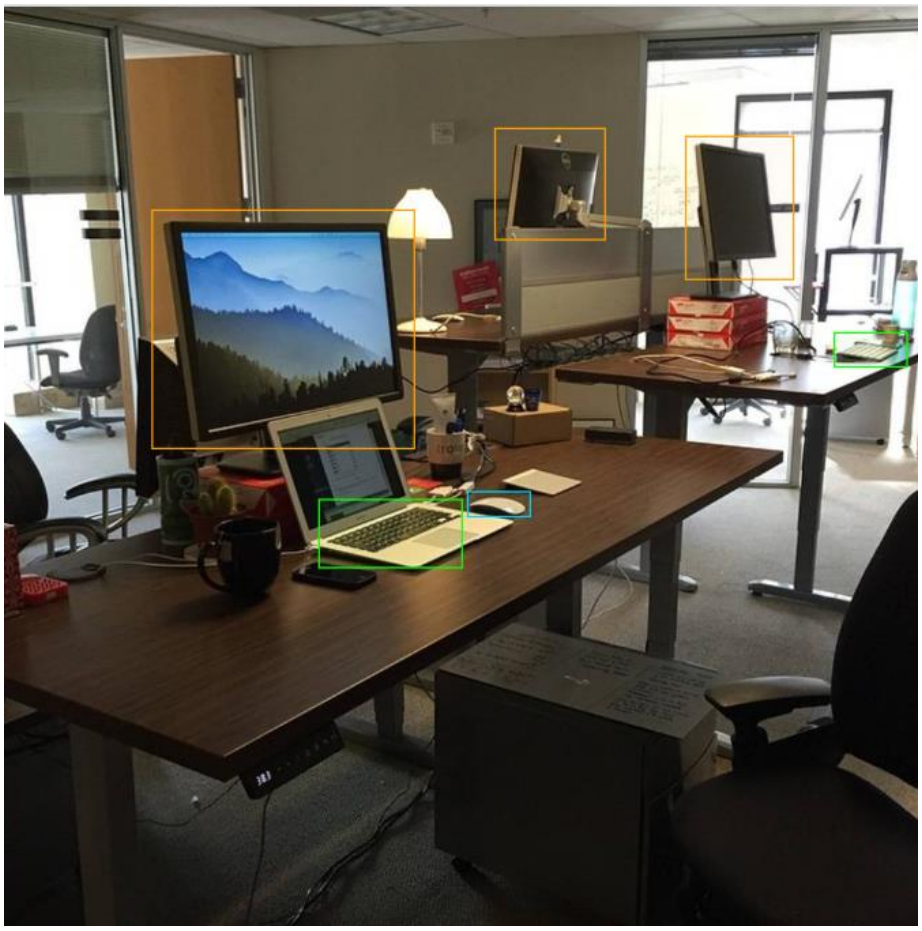
Za pravilno delovanje programa smo anotirali čim večje število slik in s tem zagotovili čim višjo natančnost in točnost zaznavanja objektov. Slike smo naložili iz seta podatkov Open images v7. To je set podatkov, ki ga podpira Google in vsebuje približno 9 milijonov slik, anotiranih na različne načine, da ustrezajo različnim nalogam povezanih z računalniškim vidom. Prenašanje slik iz seta podatkov ni najbolj preprosto, saj smo za prenos slik morali uporabiti program Anaconda prompt. Anaconda prompt je vmesnik ukazne vrstice in omogoča interakcijo z aplikacijo Anaconda in uporabo z njo povezanimi orodji in knjižnicami. Preko te aplikacije smo morali pognati ukaz, kjer smo določili katere slike želimo naložiti, v našem primeru »computer_keyboard«, »computer_mouse« in »computer_monitor«. V ukazu smo določili tudi, da želimo naložiti “train dataset.” Določili smo še število slik, ki jih želimo naložiti.

Ukaz:

```
python main.py downloader --classes computer_keyboard -type_csv train --multiclass 1 --limit 200
```

5.2 Anotacija slik

Ko smo slike uspešno prenesli, smo bili pripravljeni, da začnemo z anotacijo. Za anotacijo slik smo uporabili CVAT spletno aplikacijo. Velika prednost tega programa je, da smo lahko ustvarili skupen projekt, do katerega smo lahko dostopali vsi, vsak iz svojega računa. Ustvarili smo nalogo(task), kamor smo vnesli slike, ki smo jih prej naložili. Za CVAT smo se odločili tudi zaradi preproste uporabe. Način anotacije se nam je zdel najlažji, saj na sliki želene objekte označimo tako, da jih obkrožimo s štirikotnikom, ki ima v naprej določeno vrednost (tipkovnica, miška, monitor).



Slika 5: Anotacija slik (Vir: lasten, 2024)

Na tej sliki je predstavljen primer anotacije slike. Želene predmete smo označili s primernimi štirikotniki. Slaba stran anotacije s to aplikacijo je, da je časovno zamudna, saj je potrebno vsako sliko pregledati in označiti vsak predmet posebej.

Pri anotaciji slik je zelo pomembno, da sliko natančno pregledamo in ne spregledamo želenih objektov. Pomembno je, da označimo čisto vsak ustrezen objekt, saj le tako zagotovimo natančnost pri treniranju.

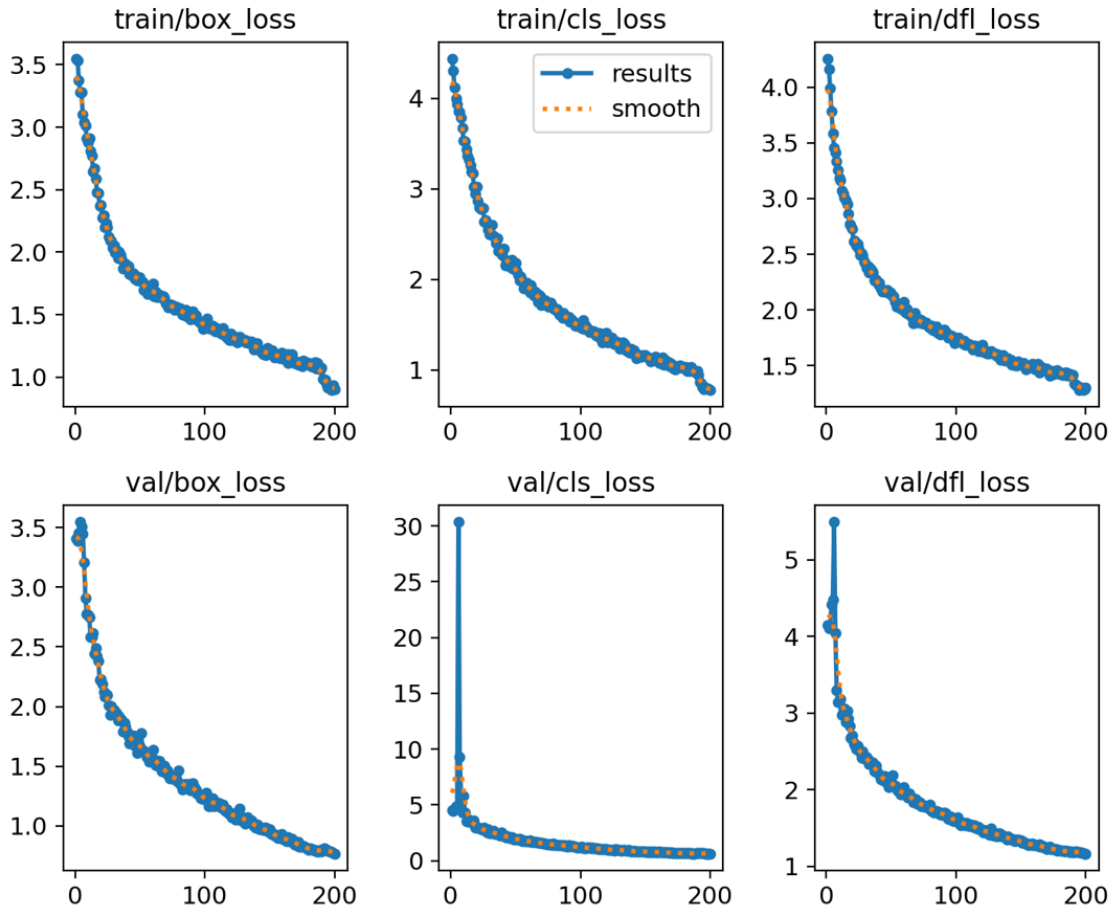
5.3 Učenje modela

Po anotaciji je bil set podatkov pripravljen za globoko učenje(deep learning). Globoko učenje smo izvedli s pomočjo programske knjižnice ultralytics, in sicer z uporabo razreda »YOLO«. Za začetek smo morali ustvariti model, ki se bo učil na podlagi naših podatkov. Za uspešen nastanek takšnega modela je potrebno podati vse potrebne

podatke o mestu seta podatkov na računalniku in kako do njega dostopati. To smo storili s pomočjo datoteke formata YAML.

Naslednji korak je bilo globoko učenje, za katerega smo uporabili model, ki smo ga ustvarili. Učenju lahko nastavimo število, ki bo določalo kolikokrat bo postopek izveden. Testirali smo z različnimi števili, splošno znano pa je, da večkrat kot se postopek ponovi, bolj točni bodo rezultati. Odločili smo se, da bomo za končno učenje proces ponovili 200-krat. Celotni proces je tako trajal več kot 10 ur. V primeru, da bi bil naš računalnik na katerem smo izvajali globoko učenje močnejši, bi se čas močno zmanjšal.

Po končanem postopku nam je program vrnil mapo z grafi poteka učenja in rezultati, ki ji potrebujemo za nadaljnje delo. Pomembno je, da grafi kažejo izboljšave pri učenju, ki se kažejo s padanjem krivulje na grafu.



Graf 2: Grafi, pokazatelji smeri učenja (Vir: lasten, 2024)

Globoko učenje generira tudi mapo imenovano "weights", v kateri so podatki globokega učenja, s pomočjo katerih lahko nato zaznavamo predmete v realnem času. Pomembno je, da uporabimo datoteko "best.pt", saj je to model z najboljšimi rezultati.

5.4 Program za zaznavanje v realnem času

Za program za zaznavanje v realnem času potrebujemo kamero in izučen model. Z uporabo ultralytics programske knjižnice omogočimo, da na podlagi videa, ki ga program prejema s kamero in na podlagi učenega modela označi predmete in nato video vrne kot okno na računalniku. V primeru, da program ne najde kamere ali kamera ne vrača primernih podatkov, se program zapre in vrne podatke, ki jih je do takrat pridobil. Program lahko zapremo tudi s tipko "q". Ko se program zanka za zaznavanje zapre, program sestavi seznam, ki vsebuje vsako sliko (frame) videa in razrede predmetov, ki jih je zaznal. Program nato izlušči sliko z največjim številom zaznanih predmetov. Predmeti se nato zapišejo v podatkovno bazo.

5.5 Izdelava podatkovne baze

Po končanem snemanju okolice in prejemu podatkov je potrebno podatke zapisati v podatkovno bazo. Podatkovno bazo smo ustvarili s pomočjo spletne aplikacije, ki deluje na "Flask framework". Z uporabo programske knjižnice SQLAlchemy smo ustvarili podatkovno bazo z atributi, ki označujejo vsak različen predmet, ki ga želimo zaznati.

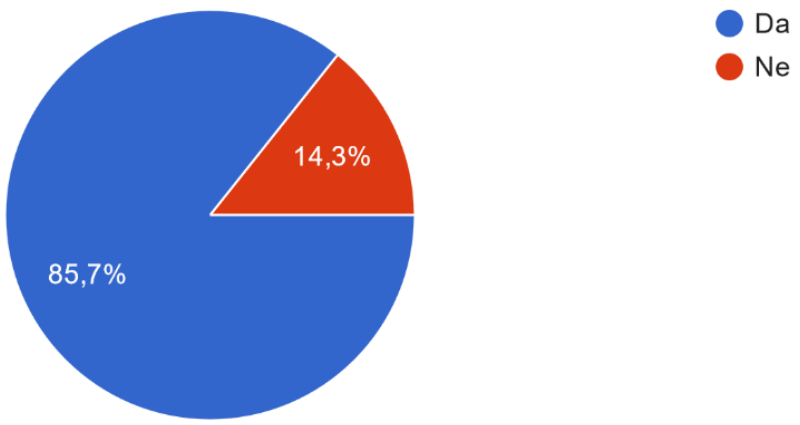
Za vpis podatkov, dobljenih z zaznavanje predmetov, smo ustvarili funkcijo, ki prešteje zaznane predmete in jih vpiše v pravilen stolpec v podatkovni bazi. Torej, če najdaljša slika (frame) vsebuje 3 tipkovnice, se bo število 3 dodalo k vrednosti v podatkovni bazi pod stolpec namenjen tipkovnicam. V primeru napake program o tem obvesti uporabnika.

6 Analiza ankete

Opravili smo anketo s pomočjo Google Forms. Anketo je izpolnilo 21 oseb. Z anketo smo želeli izvedeti kako je pojem zaznavanje objektov poznan med splošno javnostjo. Prav tako smo želeli izvedeti, če so ljudje zainteresirani za uporabo takšne aplikacije.

1. Vprašanje

Ali ste že kdaj slišali pojem "zaznavanje objektov" ali "object detection"?

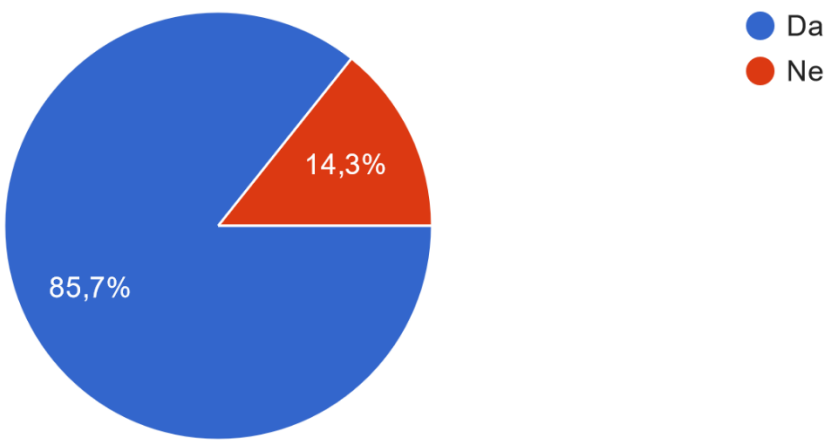


Graf 3: Poznavanje pojma "zaznavanje objektov"

Večina anketirancev (85,7%) je izjavila, da so že slišali za pojem »zaznavanje objektov« ali »object detection«, kar pomeni, da večini ta tehnologija ni tuja. Ta podatek nas je presenetil, saj smo mislili, da je to področje znano le tistim, ki se s tem ukvarjajo

2. Vprašanje

Ali bi izvajali inventuro z aplikacijo, s katero usmerite kamero v objekt in ta vam avtomatsko prišteje predmet, ki ga želite šteti?

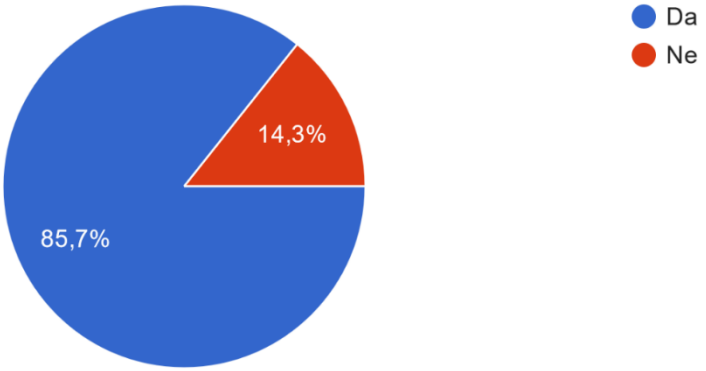


Graf 4: Zanimanje po izvajanju inventure s pomočjo zaznavanja objektov

85,7% anketirancev bi inventuro opravljalo s pomočjo aplikacije. Zaradi tega se nam zdi smiselno, da to aplikacijo razvijemo, saj se nam zdi uporabna in koristna in uporabna ljudem. To pomeni, da bi to zanimalo tudi širšo javnost in ne le peščice.

3. Vprašanje

Se vam zdi, da bi takšno aplikacijo uporabljalo veliko ljudi?

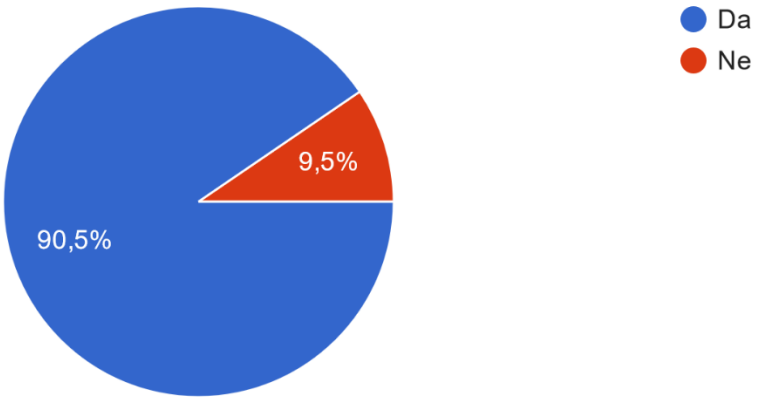


Graf 5: Uporabnost aplikacije

Tudi anketirancem se zdi, da bi takšno aplikacijo uporabljalo veliko ljudi. To je dodatna spodbuda, da aplikacijo do konca razvijemo in dodelamo.

4. Vprašanje

Se vam zdi, da bi takšna aplikacija bistveno pospešila in olajšala inventuro?

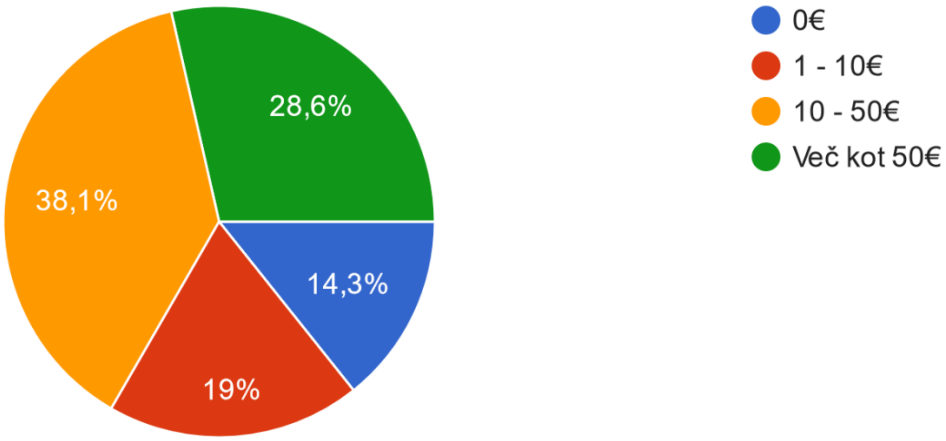


Graf 6: Olajšanje inventure s pomočjo aplikacije

Tako kot tudi nam, se veliki večini zdi, da bi takšna aplikacija inventuro lahko bistveno pospešila in olajšala. To bi podjetjem močno olajšalo delo in prihranilo čas, prav tako pa povečalo zadovoljstvo delavcev, ki jim je inventura v velik napor.

5. Vprašanje

Koliko ste pripravljeni plačati za takšno aplikacijo?



Graf 7: Koliko bi plačali za aplikacijo?

Anketirance smo prosili še za finančno opredelitev. Tukaj so bili odgovori bolj razdeljeni. Večina je odgovorila, da bi za aplikacijo plačala 10 – 50€, malo manj pa jih je odgovorilo z več kot 50€. To je za nas koristna informacija, za nadaljnji razvoj in delo.

7 Analiza hipotez

V nadaljevanju bomo ovrgli ali potrdili hipoteze, ki smo jih postavili na začetku raziskovanja. Opisali bomo, kaj smo ugotovili na podlagi rezultatov, ki smo jih dobili z izdelavo programa za zaznavanje objektov.

1.HIPOTEZA: Izdelali bomo program, ki bo uspešno zaznal predmete in jih zabeležil v podatkovno bazo.

Prvo hipotezo lahko potrdimo, saj nam je uspelo napisati kodo za program, ki uspešno zaznava predmete. Zaznava tipkovnice, monitorje in računalniške miške. Predmete zaznava z natančnostjo med 60% in 80%, kar je glede na število ponovitev pri treniranju primerna natančnost. Z večjim številom ponovitev, bi lahko izboljšali natančnost tudi na 90% ali več.

2.HIPOTEZA: Za zaznavanje predmetov bomo lahko uporabili spletno kamero

Za zaznavanje predmetov smo v postopku testiranja uporabili različne kamere, tako spletna kamere povezane z USB kablom kot tudi kamero na prenosnem računalniku. K natančnosti zaznavanja pripomore tudi boljša kvaliteta posnetka, kar pomeni, da kamera z višjo resolucijo pripomore k natančnejšemu zaznavanju predmetov. Ugotovili smo, da kamere na prenosniku in druge cenejše kamere niso primerne za zaznavanje predmetov, saj predmeti niso dovolj vidni. Kljub slabši resoluciji kamere program deluje, le da je natančnost nižja.

3.HIPOTEZA: Uspešno bomo zaznavali 10 predmetov

Tretjo hipotezo lahko ovržemo. V času testiranja smo uspešno zaznavali 6 predmetov. To so tipkovnica, monitor, miška, pisarniška miza, pisarniški stol int tiskalnik, ki smo jih zaznali vsakič, ko so bili posneti s kamero. V nadaljevanju bi ta spekter morali razširiti in s tem omogočili boljšo uporabo aplikacije.

8 Zaključek

Z raziskovanjem smo pridobili znanje o zaznavanju predmetov in metodah, ki so uporabljene za to nalogo. Spoznali smo različne programske knjižnice, ki so nam olajšale pisanje kode in izvajanje nalog. Tekom raziskovanj smo se srečali z mnogimi problemi, ki smo jih rešili s podrobnejšimi raziskavami in pregledovanjem kode, odgovore pa smo našli tudi na spletu, kjer kar mrgoli spletnih strani z odgovori. Pridobili smo tudi znanje o poteku inventure, kar nam je omogočilo, da smo program zasnovali tako, da rešuje problem. S pomočjo ultralytisc programske knjižnice je zaznavanje predmetov mogoče razširiti na različna področja, kar nam omogoča, da lahko inventuro izvajamo tudi za različne druge predmete in ne samo tiste v učilnicah. Kot lahko razberemo iz hipotez, smo si zastavili realne cilje in jih speljali do konca. Verjamemo, da lahko ta program v prihodnosti pripomore k izboljšanju postopka inventure in predvsem zmanjša čas tega postopka.

8.1 Pogled v prihodnost

To še zagotovo ni zaključek tega projekta. Prvi korak nadaljnje raziskave je, da ga izpilimo in nadgradimo. Prestaviti ga želimo v mobilno aplikacijo, kar bo omogočalo uporabniku bolj prijazno izkušnjo in lažje delo. Mobilna aplikacija bi prav tako izboljšala kvaliteto kamere, saj se kamere na mobilnih telefonih zdaj že lahko primerjajo s kamerami za profesionalno rabo, zato bi se učinkovitost zaznavanja povečala.

Razširili bi spekter predmetov, ki jih program zaznava v okolju. To bi omogočilo večje zanimanje podjetij in organizacij. V primeru zanimanja podjetij, bi lahko aplikacijo prilagodili njihovim potrebam in željam in s tem omogočili bolj specifično zaznavanje.

V prihodnosti bi prav tako v našo aplikacijo želeli vključiti umetno inteligenco, ki bi izboljšala učenje modela za zaznavanje. Umetna inteligenca bi pripomogla k učenju med uporabo aplikacije, kar pomeni da bi se točnost zaznavanja izboljševala sproti.

9 Viri in literatura

Detect an object with OpenCV-Python:. (10. Oktober 2023). Pridobljeno iz https://www.geeksforgeeks.org/detect-an-object-with-opencv-python/

Gallagher, J. (6. December 2023). *How to Detect Objects with YOLOv8*. Pridobljeno iz https://blog.roboflow.com/how-to-detect-objects-with-yolov8/

geeksforgeeks. (brez datuma). Pridobljeno 28. 2 2024 iz SQLAlchemy – Introduction: https://www.geeksforgeeks.org/sqlalchemy-introduction/

github. (13. 4 2020). Pridobljeno 10. 12 2023 iz opencv: https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalcatface_extended.xml

Material Design. (brez datuma). *Object detection: live camera*. Pridobljeno iz https://m2.material.io/design/machine-learning/object-detection-live-camera.html#usage

Rosebrock, A. (11. September 2017). *Object detection with deep learning and OpenCV*. Pridobljeno iz https://pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/

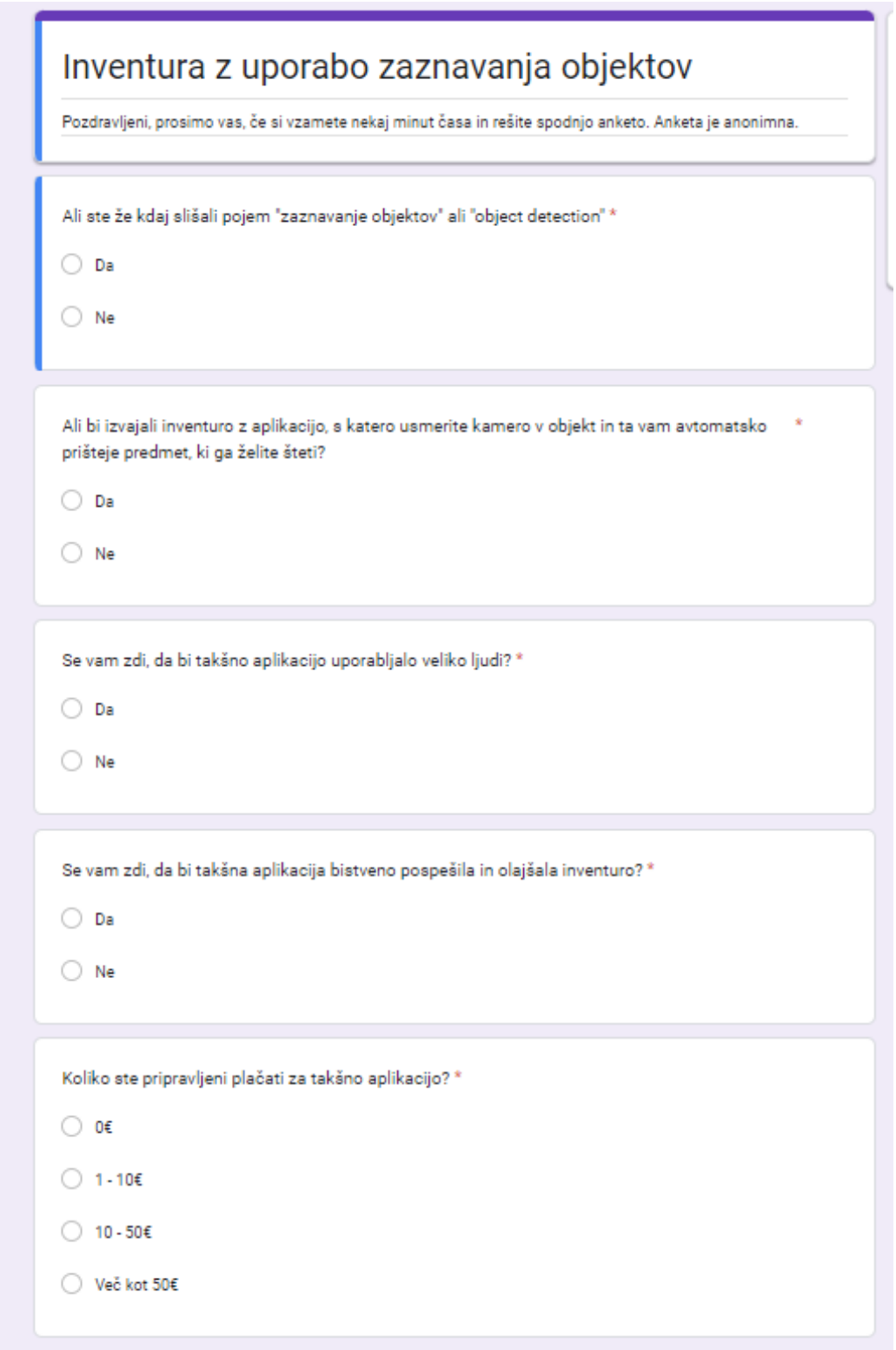
TensorFlow. (15. December 2021). *Youtube*. Pridobljeno iz Train a custom object detection model using your data: https://www.youtube.com/watch?v=-ZyFYniGUsw

Wikipedia. (19. 11 2002). Pridobljeno 28. 2 2024 iz Python: https://en.wikipedia.org//wiki/Python_(programming_language)

Wikipedia. (24. 2 2015). Pridobljeno 28. 2 2024 iz PyCharm: https://en.wikipedia.org/wiki/PyCharm

10 Priloga

10.1 Anketni vprašalnik



Inventura z uporabo zaznavanja objektov

Pozdravljeni, prosimo vas, če si vzamete nekaj minut časa in rešite spodnjo anketo. Anketa je anonimna.

Ali ste že kdaj slišali pojem "zaznavanje objektov" ali "object detection" *

- Da
- Ne

Ali bi izvajali inventuro z aplikacijo, s katero usmerite kamero v objekt in ta vam avtomatsko prišteje predmet, ki ga želite šteti? *

- Da
- Ne

Se vam zdi, da bi takšno aplikacijo uporabljalo veliko ljudi? *

- Da
- Ne

Se vam zdi, da bi takšna aplikacija bistveno pospešila in olajšala inventuro? *

- Da
- Ne

Koliko ste pripravljeni plačati za takšno aplikacijo? *

- 0€
- 1 - 10€
- 10 - 50€
- Več kot 50€