ŠOLSKI CENTER VELENJE

ELEKTRO IN RAČUNALNIŠKA ŠOLA

Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

UPORABA UMETNE INTELIGENCE ZA PREPOZNAVANJE NOŠENJA MASK

Tematsko področje: aplikativni inovacijski predlogi in projekti

Avtorja:

Marcel Andrej Beliš, 4. letnik

Anže Plazl. 4. letnik

Mentor:

Rok Urbanc, dipl. inž. rač. in inf. tehnol. (UN)

Velenje, 2021

Raziskovalna naloga je bila opravljena na ŠC Velenje, Elektro in računalniška šola, 2021.

Mentor: Rok Urbanc, dipl. inž. rač. in inf. tehnol. (UN)

Datum predstavitve: april 2021

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD ŠC Velenje, šolsko leto 2020/21

KG Aplikativni inovacijski predlogi in projekti

AV BELIŠ, Marcel/ PLAZL, Anže

SA URBANC, Rok

LE Vlasta Leban

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola

LI 2021

IN UPORABA UMETNE NTELIGENCE ZA PREPOZNAVANJE NOŠENJA MASK

TD RAZISKOVALNA NALOGA

OP IX, 18 str., 0 pregl., 0 graf., 2 sl., 0 pril., 10 vir.

IJ SL

JI sl/en

AI V današnjem času pandemije se soočamo z različnimi problemi, kot so nošenje mask v zaprtih prostorih, česar se veliko ljudi ne drži. Do te ideje sva prišla, ko sva na šoli videla veliko dijakov, ki niso nosili mask in so s tem ogrožali svoje zdravje in zdravje dijakov in profesorjev okoli sebe. Najprej sva se dogovorila, katere programske jezike bi uporabila ter kakšno tehnologijo. Sama sva napisala kodo za učenje in delovanje nevronske mreže tako, da sva prepoznala masko na obrazu osebe. Za optimalno delovanje sva morala optimizirati že obstoječo kodo tako, da sva prepoznala samo najbližjo osebo. Učinkovitost programa sva testirala z iteracijami učenja nevronske mreže ter popravki programa. Najin cilj je omogočiti varno okolje in spomniti ljudi, da nosijo maske.

KEYWORDS DOCUMENTATION

ND ŠC Velenje, 2020/21

CX Applied innovation proposals and projects

AU BELIŠ, Marcel/ PLAZL, Anže

AA URBANC, Rok

PR Vlasta Leban

PP 3320 Velenje, SLO, Trg mladosti 3

PB ŠC Velenje, Elektro in računalniška šola

PY 2021

TI USING ARTIFICIAL INTELLIGENCE TO RECOGNIZE WEARING MASKS

DT RESEARCH WORK

NO IX, 18 p., 0 tab., 0 graf, 2 fig., 0 ann., 10 ref.

LA SL

AL sl/en

AB In today’s pandemic era, we face various problems such as wearing masks indoors, which many people do not adhere to. We came up with this idea when we saw a lot of students at school who weren’t wearing masks and thereby endangering their health and the health of the students and professors around them. We first agreed on which programming languages to use and what technology. We wrote the code for learning and operating the neural network ourselves by recognizing the mask on a person’s face. For optimal performance, we had to optimize the existing code so that we recognized only the closest person. We tested the effectiveness of the program with iterations of neural network learning and program corrections. Our goal is to provide a safe environment and remind people to wear masks.

KAZALO KRATIC

ŠCV – Šolski center Velenje

ŠD – številka dokumenta

KG – klasifikacijska gesla

AV – avtor

SA – sekundarni avtorji

KZ – kraj založbe

ZA – založnik

LI – leto izdaje

IN – izvirni naslov

TD – tip dokumenta

OP – opombe

IJ – izvirni jezik

JI – jezik izvlečka

AI – avtorski izvleček

AR - razširjena resničnost

API - aplikacijski programski vmesnik

iOS – iPhone operacijski sistem

KAZALO VSEBINE

KAZALO SLIK

1 UVOD

Idejo za raziskovalno nalogo sva se spomnila pri pouku, ko sva opažala, da veliko ljudi pozabi nadeti masko pri vhodu v šolo. Takšno početje je lahko nevarno tako za dijake kot tudi uslužbence šole. Tako sva si zadala cilj, da bova napisala program, ki bo uporabljal umetno inteligenco in računalniški vid, s katerim bova prepoznala ali dijak nosi masko ali ne in se mu bodo, na podlagi tega, odprla vrata. Za raziskovalno sva se odločila, ker naju je zanimalo to področje kako računalniki prepoznajo obraz, ter kako delujejo nevronske mreže. Za pomoč glede računalniškega vida sva pa prosila svojega učitelja Roka Urbanca, in je z veseljem priskočil na pomoč in nama postal mentor, ki pa nama je kot uvod v računalniški vid, predlagal programski jezik Python.

1.1 HIPOTEZE

- Programa nebo motilo kakšne barve je maska
- Program bo prepoznal razliko med medicinsko masko in navadno ruto
- Program bo prepoznal ali je maska pravilno nameščena

2 RAČUNALNIŠKI VID

Definicija računalniškega vida.

Področje računalništva, ki se ukvarja z računalniškimi sistemi, zmožnimi interpretacije in analize slikovnih podatkov. Podobno kot ljudje s pomočjo vida razpoznavamo osebe, predmete in svojo neposredno okolico, da bi se lahko v njej gibali in delovali, naj bi računalniški vid to omogočil tudi računalnikom oziroma robotom.

Začel se je razvijati v 70-tih letih 20. stoletja, ko je procesorska moč računalnikov postala dovolj zmogljiva. Osnovni motiv za nastanek računalniškega vida je povezan z idejo umetne inteligence, ki s pomočjo računalnikov rešuje zahtevnejše naloge. Cilj računalniškega vida je potem postal, da bi s pomočjo vizualnih informacij reševal naloge.

2.1 Uporabnost računalniškega vida

Nekateri mislijo, da je računalniški vid nekaj iz daljne prihodnosti oblikovanja. Ni res. Računalniški vid je že vključen v številna področja našega življenja. Spodaj je le nekaj pomembnih primerov, kako danes uporabljamo to tehnologijo:

Organizacija vsebine

Sistemi računalniškega vida nam že pomagajo organizirati vsebino. Apple Photos je odličen primer. Aplikacija ima dostop do naših zbirk fotografij, fotografijam pa samodejno doda oznake in nam omogoča brskanje po bolj strukturirani zbirki fotografij. Apple Photos je zanimiva, ker aplikacija ustvari kuriran pogled na vaše najboljše trenutke za vas. V razdelku Za vas za Fotografije za iOS si lahko ogledate predstavljeno vsebino, ki jo je ustvarila aplikacija, tako da si lahko ogledate najljubše trenutke. V razdelku Za vas za Fotografije za iOS lahko vidite predstavljeno vsebino, ki jo je ustvarila aplikacija, tako da si lahko ogledate najljubši trenutki.

Prepoznavanje obraza

Prepoznavanje obraza je ključna tehnologija za biometrično preverjanje pristnosti. Številne mobilne naprave, ki so danes na voljo na trgu, uporabnikom omogočajo, da naprave odklenejo tako, da pokažejo obraze. Kamera na sprednji strani se uporablja za prepoznavanje obraza; mobilne naprave obdelajo to sliko in na podlagi analize lahko ugotovijo, ali je oseba, ki drži napravo, pooblaščena za to napravo. Lepota te tehnologije je, da deluje zelo hitro.

Razširjena resničnost

Računalniški vid je osrednji element aplikacij razširjene resničnosti. Ta tehnologija aplikacijam AR omogoča, da v realnem času zaznajo fizične predmete (tako površine kot posamezne predmete v določenem fizičnem prostoru) in te informacije uporabijo za umestitev navideznih predmetov v fizično okolje. Aplikacija Ikea Place uporablja AR, da uporabnikom pomaga razumeti, ali se pohištvo, ki ga želijo kupiti, prilega njihovi notranjosti. Aplikacija Ikea Place uporablja AR, da uporabnikom pomaga razumeti, ali pohištvo, ki ga želijo kupiti, ustreza njihovi notranjosti.

Samo vozeči avtomobili

Računalniški vid omogoča avtomobilom, da razumejo svojo okolico. Pametno vozilo ima nekaj kamer, ki zajemajo videoposnetke z različnih zornih kotov in jih pošiljajo kot vhodni signal v

programsko opremo za računalniški vid. Sistem v realnem času obdela video in zazna predmete, kot so cestne oznake, predmeti v bližini avtomobila (na primer pešci ali drugi avtomobili), semaforji itd. Eden najpomembnejših primerov uporabe te tehnologije je avtopilot v avtomobilih Tesla.

Zdravje/Medicina

Podatki o sliki so ključni element diagnoze v medicini, saj predstavljajo 90 odstotkov vseh zdravstvenih podatkov. Številne zdravstvene diagnoze temeljijo na obdelavi slik - rentgenski žarki, magnetna resonanca in mamografija, če naštejemo le nekatere. In segmentacija slik je med analizo medicinskih preiskav dokazala svojo učinkovitost. Na primer, algoritmi računalniškega vida lahko zaznajo diabetično retinopatijo, najhitreje rastoči vzrok slepote. Računalniški vid lahko obdela slike očesa (glejte spodaj) in jih oceni glede na prisotnost in resnost bolezni. Algoritmi računalniškega vida se lahko uporabljajo za obdelavo fotografij očesnega očesa mrežnice za prikaz diabetične retinopatije. Še en pomemben primer je odkrivanje raka. Natančnost diagnosticiranja različnih oblik raka je ključnega pomena. Po Googlovih besedah orodja za računalniški vid pomagajo pri odkrivanju metastaz raka z veliko večjo natančnostjo kot pri človeških zdravnikih. Algoritem računalniškega vida uspešno prepozna tumorsko regijo (živo zeleno) in ga običajna področja, ki so videti kot tumorji, ne zmedejo.

Kmetijstvo

Številne kmetijske organizacije uporabljajo računalniški vid za spremljanje letine in reševanje skupnih kmetijskih problemov, kot so pojav plevela ali pomanjkanje hranil. Sistemi računalniškega vida obdelujejo slike s satelitov, brezpilotnih letal ali letal in poskušajo težave odkriti v zgodnji fazi, kar pomaga preprečiti nepotrebne finančne izgube.

2.2 Delovanje

Tehnologija računalniškega vida posnema način delovanja človeških možganov. Naši možgani prepoznajo elemente, ki jih vidimo tako, da se zanašajo na vzorce za dekodiranje posameznih predmetov. Ta koncept se uporablja za ustvarjanje sistemov za računalniški vid.

Algoritmi računalniškega vida, ki jih danes uporabljamo, temeljijo na prepoznavanju vzorcev. Računalnike treniramo na ogromni količini vizualnih podatkov - računalniki obdelujejo slike, na njih označujejo predmete in v njih najdejo vzorce. Če na primer pošljemo milijon podob cvetov, jih računalnik analizira, prepozna vzorce, ki so podobni vsem cvetjem, in na koncu tega postopka ustvari model "cvet". Posledično bo računalnik lahko natančno zaznal, ali je določena slika roža vsakič, ko jim pošljemo slike.

Programska oprema ne vidi slike kot celota ampak jo vidi z eno samo 8-bitno številko, ki se giblje od 0 (črna) do 255 (bela). Te številke so tisto, kar programska oprema vidi, ko vnesete sliko. Ti podatki so na voljo kot vhod v algoritem računalniškega vida, ki bo odgovoren za nadaljnjo analizo in odločanje.

2.3 Revolucija računalniškega vida

Da bi razumeli nedavni proces tehnologije računalniškega vida, se moramo poglobiti v algoritme, na katere se ta tehnika opira. Sodobni računalniški vid temelji na globokem učenju, specifični podskupini strojnega učenja, ki z algoritmi pridobiva vpogled v podatke. Strojno učenje pa se naslanja na umetno inteligenco, ki deluje kot temelj za obe tehnologiji. Poglobljeno učenje se prilega strojnemu učenju, podmnožici umetne inteligence, globoko učenje pa spada v strojno učenje, podmnožici umetne inteligence.

Poglobljeno učenje predstavlja učinkovitejši način za računalniški vid - uporablja poseben algoritem, imenovan nevronska mreža. Nevronske mreže se uporabljajo za pridobivanje vzorcev iz predloženih vzorcev podatkov. Algoritme navdihuje človeško razumevanje delovanja možganov, zlasti medsebojnih povezav med nevroni v možganski skorji.

Na osnovni ravni nevronske mreže je perceptron, matematični prikaz biološkega nevrona. Podobno kot biološki nevroni v možganski skorji je mogoče imeti več plasti medsebojno povezanih perceptronov. Vhodne vrednosti (neobdelani podatki) se prenašajo po omrežju, ki ga ustvarijo perceptroni, in končajo v izhodni plasti, kar je napoved ali visoko izobraženo ugibanje o določenem objektu.

2.4 Prednosti računalniškega vida

Prednosti računalniškega vida spadajo pod fascinantno veliko naslovov. Skoraj vsak sektor, tako zasebni kot javni, lahko izkoristi uporabo računalnikov za sledenje, analizo in razlago sveta okoli njih. Ko močnejše organizacije spoznajo, kaj lahko računalniški vid in strojno učenje prineseta na mizo, bomo videli, da ta tehnologija umetne inteligence res vpliva na naše življenje.

3 NEVRONSKE MREŽE

3.1 Kaj so?

Nevronska mreža je algoritem za obdelavo informacij, ki deluje po zgledu človeških oz. živalskih možganov. Podobno kot naši možgani se morajo tudi te nevronske mreže sprva naučiti s pomočjo primerov. Model sestavlja veliko število nevronov, ki jih je potrebno za vsako aplikacijo (prepoznavanje obraza) znova naučiti z učnim procesom. Učenje v bioloških sistemih pomeni prilagajanje sinaps, ki povezujejo nevrone. Enako velja tudi za umetne nevronske mreže. Bistvo nevronskih mrež je, da skozi proces učenja same ugotovijo, kaj je prav in kaj narobe. Tako se lahko naučijo več in tudi bolje kot človek.

3.2 Katere poznamo?

Nevronske mreže delimo na topologije ter na namen in proces učenja.

3.2.1 Topologija nevronskih mrež

3.2.1.1 Preprosti nevron

Umetni nevron je nevronska mreža z velikim številom vhodov in enim izhodom. Ima dva načina delovanja. Učni način in uporabniški način. V učnem načinu naučimo mrežo kako se mora odzvati na določen vzorec na vhodu. Ko mreža na vhodni strani zazna naučeni vzorec, na izhodni strani poda ustrezen izhodni signal. Če na vhodni strani ni pravilnega vzorca, se za izhod uporabljajo pravila proženja.

3.2.1.1 Kompleksnejši nevron

Kompleksnejši nevron se od preprostejšega razlikuje v tem, da so vhodni podatki obteženi. Učinek vsakega izmed vhodnih podatkov na vsakega izmed izhodnih je odvisen od uteži na vsakem izmed vhodov. Obtežene vhodne signale nato seštejemo, in če vsota presega vnaprej določen prag, potem se vozlišče sproži. V vseh ostalih primerih se vozlišče ne sproži. Takšen nevron se z lahkoto prilagaja drugačnim situacijam, ker se uteži na vhodih lahko spreminjajo.

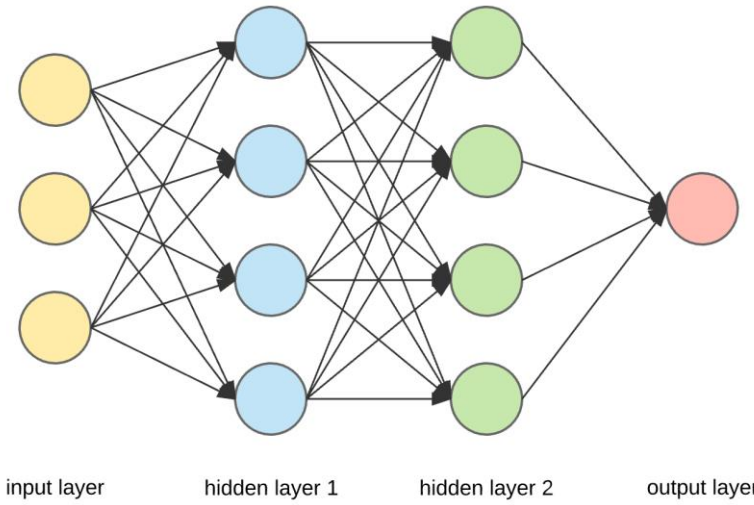
3.3 Delovanje

Tipična nevronska mreža ima lahko par ducat, sto, tisoč, ali pa več milijonov umetnih nevronov imenovanih enote razporejenih po serijah v plasteh, katere se povezujejo na obe strani.

Nekatere enote so narejene, da sprejemajo vhode. Te so znane kot vhodne enote. Lahko jih je le nekaj za preproste nevronske mreže, lahko jih je pa tudi veliko več. Koliko jih je, je odvisno kako veliko mrežo imamo in koliko različnih podatkov ji želimo dati.

Na drugu strani so izhodne enote. Te dajejo različne signale odvisne od vhodnih podatkov in kaj se je nevronska mreža že naučila.

Med vhodnimi in izhodnimi enotami je ena ali več plasti skritih enot, ki skupaj tvorijo večino nevronske mreže. Večina nevronskih mrež je popolnoma povezanih, kar pomeni, da je vsaka skrita enota in vsaka izhodna enota povezana z vsako enoto v plasteh na kateri koli strani. Povezave med eno in drugo enoto so predstavljene s številko, tako imenovano utež, ki je lahko bodisi pozitivna (če ena enota vzbudi drugo) bodisi negativna (če ena enota drugo). Večja kot je utež, večji vpliv ima ena enota na drugo.



Slika 1 Nevrosnka mreža (vir: https://francescolelli.info/tutorial/neural-networks-a-collection-of-youtube-videos-for-learning-the-basics/)

4 DELO

4.1 Okolje

4.1.1 Python

Je programski jezik, ki se uporablja za splošno uporabo. Oblika sintakse temelji na berljivosti kode z uporabo pomembnega presledka. Njegova preprosta sintaksa in prilagodljivost naj bi programerjem pomagalo pri pisanju jasne in z lahkoto berljive kode za male in velike projekte.

Python je naredil Guido van Rossum leta 1991, kot naslednik programskega jezika ABC. Python 2.0, izdan leta 2000, je predstavil nove funkcije, na primer razumevanje seznamov in druge funkcije. Nato je leta 2008 bil izdan Python 3.0, ki pa ni popolnoma podpiral kode iz verzije 2.0, če je nismo vsaj malo spremenili. Sedaj najnovejša različica Pythona je 3.9.0.

Za najin projekt sva se odločila, da bova uporabila Python 3.8.5.

4.1.2 Knjižnice

Python podpira veliko knjižnic za delo z različnimi projekti. Zato tudi govorijo, da je Python programski jezik, ki ima »baterije že priložene«. Knjižnice v Pythonu ohlapno opisuje zbirko osnovnih modulov.

Za raziskovalno nalogo sva uporabila različne knjižnice od knjižnic za delanje z večjimi zbirkami podatkov ter za zajemanje slike iz kamere.

4.1.2.1 OpenCV

Je knjižnica, ki je namenjena reševanju težav z računalniškim vidom.

4.1.2.2 NumPy

Se uporablja za delo z nizi. Ima tudi funkcije za delo na področju linearne algebre, Fourierjeve transformacije in matric. NumPy je leta 2005 ustvaril Travis Oliphant. Gre za odprtokodni projekt, ki ga lahko uporabljate prosto.

4.1.2.3 Keras

Keras je API, zasnovan za ljudi, ne za stroje. Keras sledi najboljšim praksam za zmanjšanje kognitivne obremenitve: ponuja dosledne in preproste API-je, zmanjšuje število uporabniških dejanj, potrebnih za običajne primere uporabe, in zagotavlja jasna in uporabna sporočila o napakah. Ima tudi obsežno dokumentacijo in vodnike za razvijalce.

4.1.2.4 Scikit-learn

Je brezplačna knjižnica strojnega učenja za Python. Ima različne algoritme, kot so podporni vektorski stroj, naključni gozdovi in k-sosedje, podpira pa tudi numerično in znanstveno knjižnico Pythona, NumPy.

4.1.2.5 Imutils

Serija priročnih funkcij za poenostavitev osnovnih funkcij obdelave slik, kot so prevajanje, vrtenje, spreminjanje velikosti, skeletonizacija, prikaz slik knjižnice Matplotlib, razvrščanje kontur, zaznavanje robov in še veliko več z uporabo OpenCV-ja in Python-a 3.

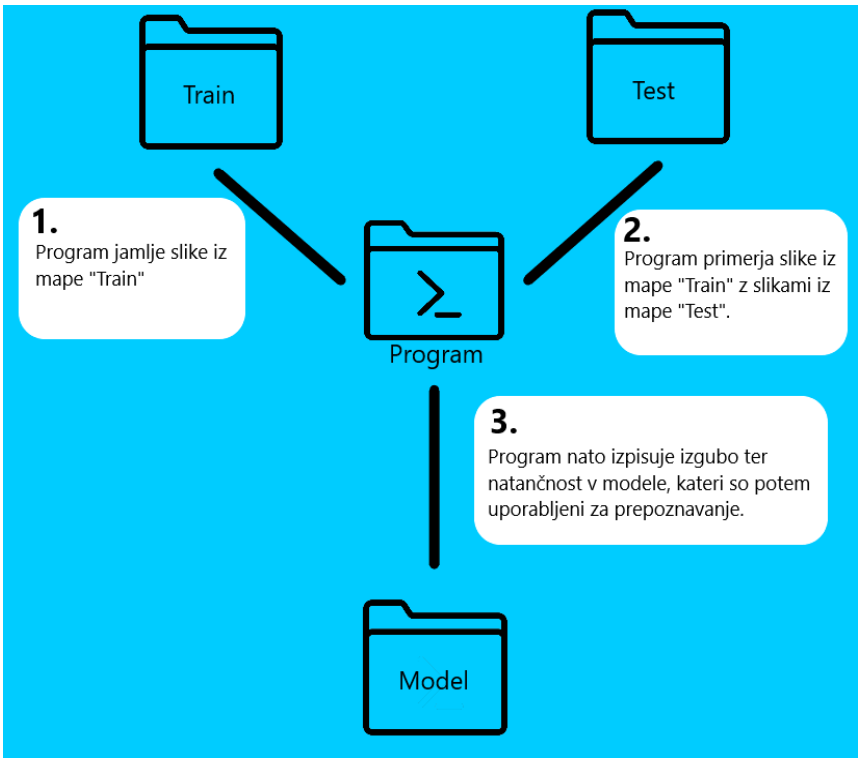
4.2 Opis potek dela

4.2.1 Treniranje programa

Sprva sva napisala program, za treniranje svoje umetne inteligence. To sva naredila v Pythonu in poimenovala datoteko »train.py«. Koda je delovala tako, da sta v mapi »train« ter »test« bili dve mapi. V eni mapi so bile slike ljudi z maskami in v drugi slike ljudi brez mask. Vse slike so bile naložene s spleta.

Mapa »train« je program uporabljaj kot temelje, da se je naučila kaj je pravilno

Program je potem sam primerjal ljudi z maskami in brez mask in zapisoval ugotovitve v modele. Med učenjem je program izpisoval, kolikšna je izguba in kolikšna je natančnost programa v dvajset različnih modelov. Paziti sva morala, da novi model ni imel večje izgube in slabše natančnosti kot prejšnji model. Če je prišlo do tega, sva morala takoj ustaviti



Slika 2 Delovanje programa (Vir: Plazl Anže)

program in naložiti nove slike ljudi z in brez mask. Ti modeli nato predstavljajo nevronsko mrežo, s katerimi bomo v prihodnjem programu lahko ugotovili ali ima človek masko gor ali ne.

Ko sva dobila najboljši model, to je model, ki ima najmanjšo izgubo in največjo natančnost, sva lahko začela pisati program, s katerim bi uporabila kamero, da prepoznava, ali ima človek masko gor ali ne.

Program sva poimenovala »test.py«, da sva testirala, ali program deluje in ali modul deluje in pravilno zazna človeka. Sprva je koda uporabila kamero, da je lahko videla človeka. Nato je koda uporabljala modul, da je prepoznala, ali ima človek gor masko ali ne in je okrog njega narisala pravokotnik in besedilo glede na to, ali je bila maska prisotna ali ne. Če je oseba imela masko gor, je napisalo, da ima masko gor in zelen pravokotnik, drugače pa je pravokotnik bil rdeč in pisalo, da nima maske gor.

4.2.2 Težave in rešitve

Vse skupaj sva hotela narediti tako, da bi se lahko uporabljal na preprostem računalniku na eni plošči kot je Raspberry PI ampak je prišlo do težave pri glavnemu programu »train.py«, ker je uporabljal preveliko vsoto virov na navadnem računalniku, ki je imel močnejšo procesno moč, kaj šele na Raspberry Pi-ju. Program je iz 60 sličic na sekundo padel na 12-15 sličic na sekundo kar ni bilo lepo za gledati.

Med pregledovanju kode sva ugotovila, da veliko virov porabi »for« zanka, katera vsakemu uporabniku nariše posebej pravokotnik. To sva naredila zato, ker je lahko več ljudi na enkrat na kameri. Rešitev je bila, da nariševa pravokotnik samo na najbližjega človeka. Tako sva optimizirala kodo in jo pospešila, da sva konstantno dobivala 60 sličic na sekundo.

5 DISKUSIJA

Dandanes, zaradi bolezni, mora veliko ljudi po svetu nositi masko. Vendar veliko ljudi ne nosi maske in tako nastanejo večji problemi, ki vodijo do večjih problemov in težav. Zato so krivi ljudje, ki ne nosijo mask.

Pri večjih podjetjih lahko pride do velike okužbe večje skupine ljudi, če že en sam človek ne nosi maske. Večje komplikacije nastanejo, če bližnji od delavcev v podjetju delajo v zdravstvu. Tako imamo verižno reakcijo, ki lahko okuži večje število ljudi, kot bi pa bilo treba.

Zato bi lahko vsa podjetja, katera imajo večje skupine ljudi na enem mestu, implementirale najin projekt na vse vhode. Tako bi zavarovali tudi redarje pri vhodih kot tudi delavce, ki vstopajo v podjetje.

Zato sva s tem namenom ustvarila program, ki bo prepoznal ali ima človek gor masko in si zadala 3 hipoteze.

Hipoteza 1: Programa nebo motilo kakšne barve je maska

Ko sva začela delati program, sva imela samo eno masko. Svetlo modro medicinsko masko. Ta maska je bila podlaga za vse ostale slike, ki sva jih potem podala. Nikoli pa nisva pomislila na različne barve mask, ki jih imajo ljudje. Tako sva najprej pomislila, da bi začela dodajati različne barve mask programu in ga učiti, da obstajajo tudi druge barve mask. To sva kar hitro ovrgla in poiskala drugo rešitev, ki je pa vse slike spremenila iz barvne in pisane oblike v dvobarvno črno-belo sliko. Tako sva iz mnogo barv, ki so na barvni lestvici, zožila opcijo na samo barve, ki so med belo in črno. Takšna koda bo porabila manj virov pri treniranju in bo posledično bolj natančna.

Hipoteza je na podlagi ugotovitev potrjena.

Hipoteza 2: Program bo prepoznal razliko med medicinsko masko in navadno ruto

Med pisanjem kode sva se tudi vprašala, ali bi bilo dovolj, če ima človek samo majico čez del, kjer bi morala biti maska. Tako sva iskala slike, kjer ima človek ruto, jopico, majico, kakršno koli drugo oblačilo skozi usta, ki ni maska in jo podala kot napačen odgovor, oziroma da človek nima maske gor. Sprva se je program malo izgubil in je izguba dosegla najvišje število, kar sva jo videla. Vendar je posledično po treh poizkusih vendarle se naučil in ločil razliko med masko in ostalimi kosi oblačila.

Hipoteza je na podlagi ugotovitev potrjena.

Hipoteza 3: Program bo prepoznal ali je maska pravilno nameščena

Po končanem programu sva nato veliko ljudi videla, da maske nosijo pod nosom, nad brado ali kakorkoli drugače, kar je pa seveda ne pravilno. Tako sva se odločila, da bova v svoj program implementirala ali človek pravilno nosi masko. Za to sva morala znova na trenirati model in pri tem upoštevati vse nepravilne načine nošenja maske. Po kar nekaj fotografijah nepravilno nameščenih mask in ponovnem treniranju modela sva dobila dobre rezultate. Program in model sva nato testirala, in bila kar zadovoljna z rezultatom in začela uporabljati ta model kot primarni model najinega programa.

Hipoteza je na podlagi ugotovitev potrjena

6 ZAKLJUČEK

Izdelava raziskovalne naloge je bila vsekakor izziv. Prav tako pa sva se skozi raziskovalno nalogo veliko naučila in pridobila novega znanja na podlagi nevronskih mrež in računalniškega vida.

Ugotovila sva, da bi bil ta izdelek lahko uporabljen v podjetjih kot tudi v javnih ustanovah.

Po najinem mnenju je cilj raziskovalne naloge v veliki meri dosežen, saj nama je uspelo uresničiti cilj in željo, da narediva lastni program za zaznavanje mask. Ker naju to področje zanima, nama je bila izdelava te naloge v veselje, ampak tako kot vsepovsod sva imela tudi težave, nekatere manjše, kot je iskanje pravilnih različic programov, ki bodo skupaj delovali. Imela sva pa tudi veliko večjo težavo, programu zagotoviti dovolj veliko slik pri treniranju AI za čim večjo uspešnost.

Imava pa tudi željo, da bi se lahko ta program naložil na Raspberry Pi in dal signal, da se vrata odprejo samo, če dijak nosi masko. Vendar kadar je na sliki več kot en dijak, pride do težave, da se delovanje sistema zelo upočasni. Program sva poskušala optimizirati ampak je trenutno najboljša rešitev ta, da je na kameri viden samo en dijak. Vendar si še vedno želiva program optimizirati tako daleč, da ni važno, koliko ljudi je na kameri ter da program dela brez problema.

Ampak kot je že omenjeno, sva nad zaključenim izdelkom zelo zadovoljena, še vedno je prostor za izboljšavo, vendar deluje izdelek mnogo bolje, kot sva si na začetku zamislila.

7 POVZETEK

Namen najine raziskovalne naloge je bil ustvariti AI program, ki bi prepoznal, ali dijaki nosijo pravilno maske, kadar prihajajo v šolo, ter bi ob pravilnem nošenju mask dal signal, da se lahko vrata odklenejo. V nasprotnem primeru pa vrata ostanejo zaklenjena toliko dolgo, dokler si dijak ne popravi maske. Ta program bi deloval na osebnem računalniku ali na Raspberry Pi, ki ima kamero, da dogajanje vidi ter monitor, da dijaku prikaže, kaj kamera vidi in če so se mu vrata odklenila.

8 SUMMARY

The purpose of our research project was to create an AI program that would recognize whether students wear masks correctly when they come to school. If they wear the mask properly, it would give a signal to the door that it can be unlocked. Otherwise, the door remains locked until the student fixes the mask. This program would run on a PC or a Raspberry Pi, which has a camera to see what is happening and a monitor to show the student what the camera sees and if the door has unlocked.

9 ZAHVALA

Raziskovalna naloga ne bi mogla nastati, če nama pri nastajanju ne bi pomagale naslednje osebe, katerim bi se rada zahvalila:

- Mentorju Roku Urbancu, za pomoč, vztrajnost, njegov prosti čas ter spodbudo;
- Recenzentu raziskovalne naloge
- komisiji Mladih raziskovalcev in koordinatorici gibanja Mladi raziskovalci Karmen Hudournik;
- staršem za spodbudo;
- vsem neomenjenim, ki so kakorkoli pomagali pri izdelavi naloge.

10 VIRI

10.1 Spletni viri

About Keras. (brez datuma). Pridobljeno iz Keras: https://keras.io/about/

Neural Network. (brez datuma). Pridobljeno iz pathmind: https://wiki.pathmind.com/neural-network

Nevronska mreža. (4. april 2019). Pridobljeno iz Wikipedija: https://sl.wikipedia.org/wiki/Nevronska_mre%C5%BEa

ogrisel. (17. marec 2021). *scikit-learn*. Pridobljeno iz GitHub: https://github.com/scikit-learn/scikit-learn/blob/main/README.rst

Oliphant, T. (31. januar 2021). *Documentation*. Pridobljeno iz NumPy: https://numpy.org/doc/stable/

Python (programming language). (3. april 2021). Pridobljeno iz Wikipedija: https://en.wikipedia.org/wiki/Python_(programming_language)

Računalniški vid. (23. marec 2021). Pridobljeno iz Wikipedija: https://sl.wikipedia.org/wiki/Ra%C4%8Dunalni%C5%A1ki_vid

Rosebrock, A. (30. april 2017). *imutils*. Pridobljeno iz GitHub: https://github.com/jrosebr1/imutils/blob/master/README.md

skvark. (2. januar 2021). *opencv-python*. Pridobljeno iz pypi.org: https://pypi.org/project/opencv-python/

Solina, F. (6. september 2012). *Računalniški vid*. Pridobljeno iz videolectures.net: http://videolectures.net/promo_franc_solina_slo/

10.2 Slikovni viri