



Srednja šola za kemijo, elektrotehniko in računalništvo
Pot na Lavo 22
3000 Celje

SISTEM ZA PREPOZNAVANJE REGISTRSKIH TABLIC

raziskovalna naloga

Avtor:

Liam Einar Matko Marcus, R.4-a

Mentor:

Andraž Pušnik

Mestna občina Celje, Mladi za Celje
Celje, 2023

Mentor Andraž Pušnik 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 Sistem za prepoznavanje registrskih tablic, katere avtor je Liam Einar Matko Marcus:

- 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, 7. 4. 2023



Podpis mentorja

Podpis odgovorne osebe

ZAHVALA

Za strokovno pomoč, nasvete in predloge se zahvaljujem Andražu Pušniku, ki me je s svojimi napotki usmerjal in tako pomagal pri nastajanju raziskovalne naloge.

POVZETEK

Samodejno prepoznavanje registrskih tablic (ANPR) je tehnologija, ki uporablja optično prepoznavanje znakov za branje registrske številke vozila. Cilj raziskovalne naloge je aplikacija, ki bo na avtomobilu, s pomočjo modela umetne inteligence, poiskala mesto registrske tablice. Z OCR (Optical Character Recognition) pa prebere številko registrske tablice. Opisal bom delovanje umetne inteligence in kreiranje modela. S pomočjo ankete pa sem pridobil mnenja dijakov glede uporabe takšnih sistemov na javnih in zasebnih površinah. Sistem za prepoznavanje registrskih tablic je glavni del nedokončane aplikacije.

Ključne besede: Zaznavanje predmetov, optično prepoznavanje znakov (OCR), prepoznavanje registrskih tablic, računalniški vid, globoko učenje, konvolucijska nevronska omrežja (CNN), OpenCV, obdelava slik, strojno učenje, Python

ABSTRACT

Automatic number plate recognition (ANPR) is a technology that uses optical character recognition to read a vehicle's registration number. The goal of the research paper is an application that will, with the help of an artificial intelligence model, find the location of the license plate on the car. It reads the license plate number with OCR (Optical Character Recognition). I will describe the operation of artificial intelligence and the creation of a model. With the help of a survey, I obtained the opinions of students regarding the use of such systems in public and private areas. The license plate recognition system is a major part of the unfinished application.

Key words: Object detection, Optično prepoznavanje znakov (OCR), Computer vision, Deep learning, Konvolucijske nevronske mreže (CNN), OpenCV, Machine learning, Python

KAZALO

1	UVOD.....	7
1.1	Predstavitev problema	7
1.2	Hipoteze	8
1.3	Cilji	8
1.4	Namen naloge.....	8
2	PROGRAMSKA OPREMA.....	9
2.1	OpenCV	9
2.2	Matplotlib.....	10
2.3	TensorFlow	10
2.4	Visual Studio Code.....	11
2.5	Python.....	11
3	TEORETIČNI DEL	12
3.1	Hranjenje osebnih podatkov.....	12
3.2	Dosedanja raziskava.....	13
4	OPIS POSTOPKA.....	15
4.1	Input/vnos	16
4.2	Zajem tablice.....	16
4.2.1	Izočrta	18
4.2.2	Model umetne inteligence	21
4.2.3	Nevronske mreže	23
4.3	Priprava slike.....	24
4.4	Segmentacija znakov	25
4.5	Prepoznavanje znakov.....	26
4.6	Output/izhod.....	26
5	PROBLEM APLIKACIJE	26

5.1 Problemi pri razvijanju aplikacije	27
6 IMPLEMENTACIJA SISTEMA	29
7 ANALIZA ANKETE.....	30
7.1 Rezultati ankete	30
8 ZAKLJUČEK	33
8.1 Hipoteze	33
8.2 Možnost nadaljnje raziskave.....	34
9 VIRI IN LITERATURA	35
10 PRILOGE	36
10.1 Priloga 1: Anketni vprašalnik.....	36

KAZALO SLIK

Slika 1: Logotip OpenCV.....	9
Slika 2: Logotip Matplotlib.	10
Slika 3: TensorFlow.....	10
Slika 4: Logotip Visual Studio Code.	11
Slika 5: Logotip Python.	11
Slika 6: Diagram poteka.	15
Slika 7: Slika pred uporabo modela za iskanje registrskih tablic.	17
Slika 8: Označena registrska tablica	17
Slika 9: Originalna slika.....	18
Slika 10: Črno-bela slika.	18
Slika 11: Vodoravni robovi.....	19
Slika 12: Binarna slika.....	19
Slika 13: Closing Morphological Transformation.	20
Slika 14: Popačena slika.	20
Slika 15: Končna slika.	21

Slika 16: Treniranje modela; originalna slika.	22
Slika 17: Treniranje modela; označena slika.	22
Slika 18: XML datoteka z koordinatami.	23
Slika 19: Približana slika registrske tablice.....	24
Slika 20: Približana slika registrske tablice z posebnim filtrom.....	25
Slika 21: Približana slika registrske tablice z označenimi robovi.	25

KAZALO GRAFOV

Graf 1: Mnenje anketirancev glede zasebne uporabe avtomatskega prepoznavanja registrskih tablic.	30
Graf 2: Mnenje anketirancev glede javne uporabe avtomatskega prepoznavanja registrskih tablic.	31
Graf 3: Mnenje anketirancev glede cene inštalacije sistema.....	32

1 UVOD

Prepoznavanje registrskih tablic in s tem enolično prepoznavanje ter razvrščanje osebnih avtomobilov in drugih prevoznih sredstev je odličen način za zmanjšanje delovne sile, potrebne pri opravljanju monotonih del.

Naloga razvijanja programa za zajemanje in beleženje števil registrskih tablic z uporabo detekcije objektov in OCR predstavlja zanimiv izziv in priložnost za razvijanje umetne inteligence.

Računalniški vid in strojno učenje sta se v zadnjem desetletju močno razvila, kar je prineslo revolucijo v načinu, kako računalniki zaznavajo in analizirajo slike. Detekcija objektov in optično prepoznavanje znakov (OCR) sta dve izmed najbolj uporabljenih metod v tem kontekstu, ki omogočata razvijanje programov za zaznavanje in beleženje različnih podatkov s slik.

Naloga zahteva uporabo številnih konceptov in metod umetne inteligence, kot so nevronske mreže, konvolucijske nevronske mreže, uporaba globokih modelov, transferno učenje in podobno.

Pri razvoju programa za prepoznavanje registrskih tablic sem se srečal s številnimi izzivi, kot je prilagajanje različnim svetlobnim pogojem, premikanju vozil in prepoznavanju različnih tipov tablic. Reševanje teh izzivov zahteva ustrezno razumevanje računalniškega vida in strojnega učenja ter kreativnost in vztrajnost pri razvoju ustreznih algoritmov.

Naloga se mi zdi zanimiva in poučna. Pomaga razviti moje sposobnosti na področju umetne inteligence in mi je omogočila pridobivanje izkušenj za delo na projektih v prihodnosti.

1.1 Predstavitev problema

Večina privatnih parkirišč, ki sem jih našel, je uporabljala eno izmed dveh opcij pri odpiranju zapornice: ali je bil to preprost daljinec ali pa se je zapornica odprla,

ko si poklical neko telefonsko številko. Drugi primer je bistveno boljši, ker zmanjšuje porabo materiala, saj lahko daš telefonsko številko komurkoli brez dodatnih stroškov. Ima pa en problem. Tej osebi ne moreš več zavrniti dostopa, ker bi bila menjava telefonskih številk draga. Če bi se zapornice odpirale glede na registrsko številko, pa bi lahko zavrnili dostop, če je njegov čas potekel. Lahko bi tudi omogočal vstop samo enkrat v primeru parkirišč za trgovine in druge podobne objekte.

1.2 Hipoteze

Pri izvedbi raziskovalne naloge sem postavil naslednje hipoteze:

Hipoteza 1: Uporaba avtomatskega prepoznavanje registrskih tablic je dovolj zanesljiva za uporabo.

Hipoteza 2: Za uporabnika je bolje, če uporablja avtomatsko prepoznavanje registrskih tablic kot, če bi imeli daljinec.

Hipoteza 3: Vgraditev tega sistema na dvorišča in v garažne hiše je finančno izvedljivo.

1.3 Cilji

Moj končni cilj je aplikacija, ki bi brala številke registrske tablice, jih skupaj z datumom in uro zapisovala v podatkovno bazo in uporabniku dala možnost izbiranja dovoljenih in nedovoljenih vozil.

1.4 Namen naloge

Osnovni namen naloge je učenje in pridobivanje novih znanj na področju umetne inteligence in računalniškega vida.

2 PROGRAMSKA OPREMA

Pri izdelavi naloge sem uporabil veliko različnih računalniških orodij z različnih področij, ki jih lahko na spletu lahko najdemo brezplačno. Večina teh orodij so bile spletne knjižnice in ogrodja. Za programski jezik sem hotel uporabiti že znane veščine, in sicer C#. Dokaj hitro sem ugotovil, da ni primeren za takšno nalogo. Po nadaljnji raziskavi sem ugotovil, da je najbolj primeren programski jezik za nalogo Python. Najpomembnejša orodja so opisana v nadaljevanju.

2.1 OpenCV

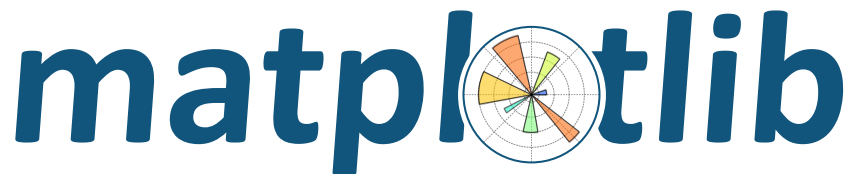
OpenCV je knjižnica programskih funkcij predvsem za računalniški vid v realnem času. Uporabil sem jo, ko sem hotel odpreti in manipulirati s sliko oz. videom. S knjižnico in njenimi metodami lahko obračam sliko, zmanjšam kvaliteto in hitrost sličic, videov, uporabim različne filtre in mnogo drugega. V nadaljevanju bom podrobneje opisal, katere metode sem uporabil.



Slika 1: Logotip OpenCV.

2.2 Matplotlib

Matplotlib je knjižnica za risanje za programski jezik Python in njegovo razširitev numerične matematike. Knjižnica je povezana z računalniškim vidom. Uporabljal sem jo predvsem za računanje v povezavi s sliko. Te knjižnice nisem veliko uporabljal, mislim pa, da je ena izmed pomembnejših.



Slika 2: Logotip Matplotlib

2.3 TensorFlow

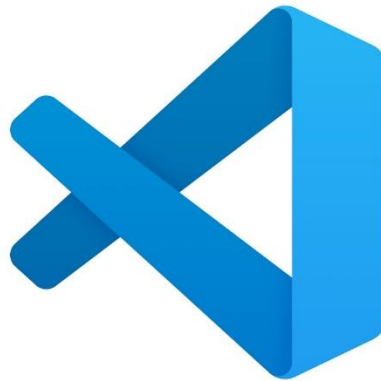
TensorFlow je brezplačna in odprtokodna programska knjižnica za strojno učenje in umetno inteligenco. Je najbolj pomembna knjižnica v mojem programu. Je tudi ogrodje (framework) za umetno inteligenco. Slednje sem uporabil pri izdelovanju modela in aplikaciji modela. To je bila najbolj pomembna knjižnica v moji nalogi.



Slika 3: TensorFlow.

2.4. Visual Studio Code

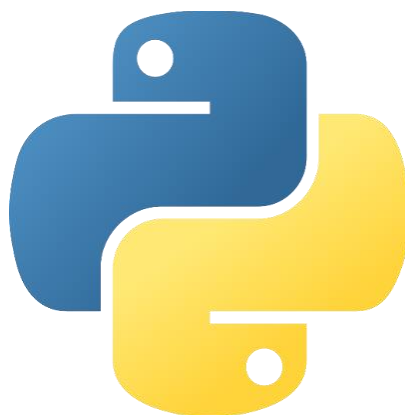
Visual studio code je integrirano razvojno okolje. Je eden izmed mnogih urejevalnikov, ki bi jih lahko izbral. Zanj sem se odločil, ker mi je najbolj všeč in ker ga najbolj poznam. Ima tudi odličen ekosistem z razširitvami, ki so mi pomagale pri nalogi.



Slika 4: Logotip Visual Studio Code.

2.5. Python

Python je interpretirani visoko ravni večnamenski programski jezik. Je eden izmed najbolj popularnih programskih jezikov. V svojem programu sem ga uporabil za celotno programiranje.



Slika 5: Logotip Python.

3 TEORETIČNI DEL

Umetna inteligenca je področje računalništva, ki se osredotoča na razvoj sistemov, ki lahko posnemajo človeško inteligenco in obnašanje. Ena izmed glavnih pod disciplin umetne inteligence je strojno učenje, ki omogoča računalnikom, da se naučijo opravljati naloge na podlagi izkušenj in preteklih primerov. Globoko učenje je trenutno najbolj obetavna in najbolj raziskana poddisciplina strojnega učenja, ki se ukvarja z razvojem umetnih nevronske mreže za obdelavo podatkov.

Računalniški vid je področje računalništva, ki se ukvarja z obdelavo in analizo digitalnih slik in videoposnetkov. Glavni cilj računalniškega vida je razviti algoritme in tehnike, ki lahko prepoznajo, razumejo in analizirajo digitalne slike in videoposnetke na način, podoben človeškemu zaznavanju. To omogoča računalniškim sistemom, da lahko samodejno analizirajo in razumejo digitalne slike in videoposnetke ter s tem pomagajo pri reševanju različnih problemov, kot so avtomatizacija industrijskih procesov, nadzor nad prometom, prepoznavanje obrazov in podobno.

Kombinacija umetne inteligence in računalniškega vida omogoča razvoj izjemno koristnih sistemov, kot so samodejno prepoznavanje in razumevanje slik, računalniško podprta diagnostika in avtonomni sistemi za upravljanje vozil in robotov. V zadnjem času se umetna inteligenca in računalniški vid uporabljata v številnih panogah, od medicinske do avtomobilske industrije, in sta postali ključni tehnologiji, ki pomagata pri izboljšanju našega vsakdana.

3.1 Hranjenje osebnih podatkov

Hranjenje osebnih podatkov je postalo vprašanje, ki je v današnji družbi zelo pomembno. Številni ljudje se bojijo, da njihovi osebni podatki, kot so ime, priimek, naslov, telefonska številka in drugi podatki ne bodo varno shranjeni.

Moja aplikacija iz te naloge ne zbira osebnih podatkov. Namen aplikacije je prepoznavanje registrskih tablic vozil in beleženje teh podatkov. Program ne

zajema nobenih osebnih podatkov o uporabnikih. Aplikacija ne zahteva prijave, v kateri bi morali uporabniki razkriti svoje osebne podatke. Prav tako aplikacija ne prepozna obrazov ali drugih osebnih značilnosti uporabnikov.

Vendar pa bi lahko aplikacija, tako kot vsaka druga, bila zlorabljena, če bi jo uporabljali zlonamerni posamezniki. Na primer, lahko bi uporabili ta program za zbiranje podatkov o gibanju vozil in tako izvedeli, kdaj je določeno vozilo na določenem mestu. To bi lahko bilo uporabljeno za zle namene, na primer za načrtovanje vlomov ali ugrabitev.

Zato je pomembno, da se uporabniki zavedajo, da aplikacije, ki se zdijo nedolžne in nevtralne, lahko postanejo nevarne, če jih uporabljajo zlonamerni posamezniki. Vedno je treba upoštevati varnostne ukrepe in se izogibati neznanim aplikacijam. Varnost podatkov in zasebnost sta v današnji družbi ključni in ju je treba vedno jemati resno.

3.2 Dosedanja raziskava

Sistemi za prepoznavanje registrskih tablic niso novost, na trgu so že nekaj let. Prvič se je ANPR pojavil leta 1993 v Londonu kot del »Ring of Steel«, ki naj bi preprečeval teroristične napade na ulice finančnega okrožja.

Leta 2003 je London ustvaril »The London Congestion Charge«. To območje v središču mesta omejuje vozila, ki se lahko vozijo tam. Vsakdo, ki se želi peljati v središče mesta, mora plačati 15£. Območje zaračunavanja, ki pokriva 20 kvadratnih kilometrov prestolnice, upravlja Transport for London in je sestavljeno iz kroga skoraj 700 kamer ANPR, ki so usposobljene na vsaki cesti v območju zaračunavanja in izven njega. To je prva obsežna uporaba ANPR v civilne namene.

Kasneje, leta 2006, je ANPR International uvedel svoj prvi statični sistem kamer za upravljanje parkiranja – bayGAURDIAN, ki je prvi modul, zgrajen za zaledni pisarniški sistem eyeTRAFFIC. Sistem je omogočal upravljavcem parkirišč, da spremljajo vozila na parkiriščih in izven njih, izračunajo čas, ki ga vozilo porabi

na lokaciji, in samodejno identificirajo vozila, ki so se pojavile brez dovoljenja ali so ostale predolgo.

Razvoj ANPR sistemov je neverjetno hitro narastel, a se je po nekaj letih hitro ustavil. Obstoječe nadgradnje niso nič posebnega.

4 OPIS POSTOPKA

Prepoznavanje registrskih tablic vključuje pridobivanje slik registrskih tablic iz predvidenega prizorišča z uporabo kamere. Fotografije ali videi so zajeti in nadalje obdelani z nizom algoritmov za prepoznavanje, ki temeljijo na obdelavi slik, da se doseže alfa numerična pretvorba zajetih slik v besedilni vnos. Po pridobitvi kakovostne slike prizorišča oziroma vozila je temeljna odvisnost katerega koli sistema ANPR od robustnosti njegovih algoritmov.

Naslednji proces opisuje pridobitev registrske številke s slike. Videe lahko razdelimo na posamezne sličice in pridobimo registrsko številko iz njih.

Postopek prepoznavanja je naslednji:



Slika 6: Diagram poteka.

4.1 Input/vnos

Preden začnem obdelovati sliko oz. video, jo moram najprej »odpreti«. Pri uporabi posnetkov v živo je proces drugačen, pri obdelavi lokalnih posnetkov pa sem uporabimo knjižnico OpenCV.

4.2 Zajem tablice

Zajem tablice s slike je najpomembnejši in hkrati najbolj težaven del naloge. Postopki delovanja so različni.

Da bi program deloval, moram zagotoviti nekaj vhodnih slik, ki jih želim preiskovati za registrske tablice. Te slike moram shraniti v ustreznem formatu na disk. Nato sem moral prebrati te slike z uporabo funkcije `cv2.imread()`, pri čemer moram prenesti pot do slike. Sledi pretvorba slike v ustrezno obliko, tako da se vhodna slika spremeni v 4D NumPy polje z velikostjo (1, 1024, 1024, 3), kar se izvede s pomočjo funkcije `cv2.resize()` in pretvorbe v tip podatkov 'float32'. Nato moram naložiti model, ki se uporablja za prepoznavanje registrskih tablic, in pripraviti interpreter za njegovo izvajanje. Sledi izvedba modela, pri čemer se izvede napovedovanje in izračun koordinat okvirja, ki obdaja registrsko tablico. Na koncu se na izhodni sliki z uporabo funkcije `cv2.rectangle()` nariše okvir okoli prepoznane registrske tablice.



Slika 7: Slika pred uporabo modela za iskanje registrskih tablic.



Slika 8: Označena registrska tablica

Kot sem omenil, je več načinov, s katerimi lahko poiščemo mesto oz. območje registrske tablice. Eden izmed načinov, po katerem bi lahko pridobili območje registrske tablice in se hkrati izognili celotnemu izdelovanju modela, je z izračunavanjem in ugotavljanjem polij. To metodo bom podrobneje opisal v nadaljevanju.

4.2.1 Izočrta

Izočrta ali konturnica funkcije dveh spremenljivk je krivulja, vzdolž katere ima funkcija konstantno vrednost, tako da krivulja povezuje točke enake vrednosti. V praksi se uporablja, da najdemo skupne točke v binarni sliki.



Slika 9: Originalna slika.

1. Za zmanjšanje šuma moramo vhodno sliko zamegliti z Gaussovo zameglitvijo in jo nato pretvoriti v sivine.



Slika 10: Črno-bela slika.

2. Na sliki poiščemo navpične robove.



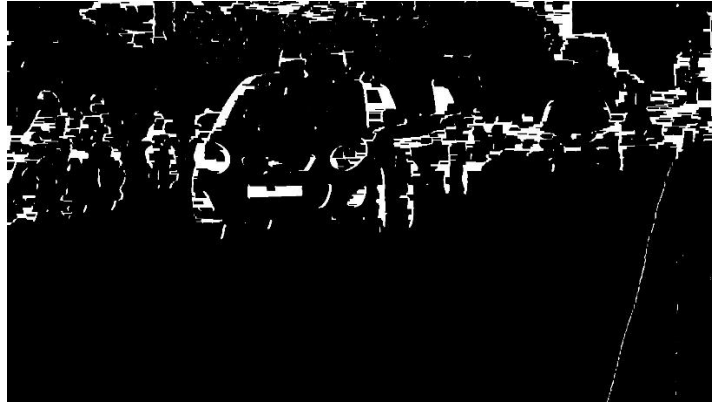
Slika 11: Vodoravni robovi.

3. Da razkrijemo registrsko tablico, moramo sliko zapisati binarno. Za to uporabimo »Otsu's Thresholding«, ki na sliki prikazuje navpične robove. Tukaj tudi nastavimo »threshold value«, tj. konstanta, ki določa, ali bo točka bela ali črna.



Slika 12: Binarna slika.

4. Nato uporabimo »Closing Morphological Transformation« na prejšnji sliki. Ta filter je uporaben za zapolnitev majhnih črnih območij med belimi na prejšnji sliki. Razkriva pravokotno belo polje z registrsko tablico.



Slika 13: Closing Morphological Transformation.

5. Za odkrivanje plošče moramo najti konture oz. izočrte na sliki. Predhodni koraki so obvezni, saj zmanjšajo število izočrt oz. kontur. Če teh postopkov ne bi naredili, bi slika izgledala tako:



Slika 14: Popačena slika.

6. Naslednji korak je, da poiščemo registrsko številko na sliki s pomočjo razmerij, in sicer razmerja dolžine in širine tablice ter razmerje velikosti tablice od celotne slike. Ko so ta razmerja potrjena in preverjena, dobimo koordinate registrske tablice.



Slika 15: Končna slika.

4.2.2 Model umetne inteligence

Modeli umetne inteligence so programski algoritmi, ki strojem omogočajo učenje iz podatkov in samodejno izvajanje nalog. Ti modeli so sestavljeni iz nevronske mreže, ki posnemajo način delovanja človeških možganov in so sposobni reševanja zapletenih nalog.

Obstajajo različni modeli umetne inteligence, od preprostih do kompleksnih, vendar se večina teh modelov osredotoča na nevronske mreže. Nevronske mreže so sestavljene iz velikega števila enot, ki so med seboj povezane. Te enote simulirajo delovanje nevronov v človeških možganih in so sposobne procesiranja in prepoznavanja vzorcev v podatkih.

Modeli umetne inteligence se učijo iz podatkov, ki jim jih predstavimo. Ti podatki vključujejo vhodne podatke in pripadajoče izhodne vrednosti. Med učenjem modeli postopoma izboljšujejo svoje napovedne sposobnosti, pri čemer se prilagajajo različnim vzorcem v podatkih.

Nekateri izmed najbolj znanih modelov umetne inteligence so nevronske mreže, k-NN, SVM, odločitvena drevesa, genetski algoritmi in druge. Vsak model ima svoje prednosti in slabosti ter se uporablja za različne naloge, kot so prepoznavanje objektov, analiza podatkov, napovedovanje trendov in drugo.

4.2.2.1 Kreiranje modela

Proces ustvarjanja modelov za umetno inteligenco je kompleksen postopek. Začne se z analizo problema (v moji nalogi je to pri opisu problema), po analizi pa pride proces zbiranja podatkov.

Pri zbiranju podatkov je pomembno, da so čisti, ustrezni in ustrezno označeni, da lahko model umetne inteligence pravilno in učinkovito deluje. Podatki morajo biti tudi uravnoteženi in pokrivati različne primere, da se izognejo nepravilnim ali pristranskim modelom. Po raziskovanju sem ugotovil, da bi za dober model potreboval 400 slik. Namesto tega sem uporabil že pripravljene slike.

Označevanje slik pomeni, da natančno opišemo, kje se naš objekt nahaja na sliki. Za to sem uporabil ImgLab, s katerim lahko označimo, kje se registracija nahaja. Koordinate registrske tablice se shranijo v .XML datoteko.



Slika 16: Treniranje modela; originalna slika.



Slika 17: Treniranje modela; označena slika.


```

<?xml version='1.0' encoding='ISO-8859-1'?>
<?xml-stylesheet type='text/xsl' href='image_metadata_stylesheet.xsl'?>
<dataset>
<name>dlib face detection dataset generated by ImgLab</name>
<images>
  <image file='porsche-1874925_1280_2.jpg'>
    <box top='344' left='602' width='114' height='43'>
      <label>unlabelled</label>
    </box>
  </image>
</images></dataset>

```

Slika 18: XML datoteka z koordinatami.

Ko imamo izbranih in označenih dovolj slik, lahko nadaljujemo z izbiro algoritmov in »treniranjem« modela. Ugotovil sem, da obstajajo nadzorovani in nenadzorovani načini treniranja modela. Uporabil sem nadzorovanega, ker sem mu podal mesta registrskih tablic. Ta se uči na svojih napakah in se tako izboljšuje. Treniranje modela lahko traja od nekaj minut do nekaj dni, odvisno od velikost podatkovne zbirke, kompleksnosti modela, strojne opreme, ki se uporablja za treniranje, in algoritma, ki se uporabljajo. Po končanem treniranju imamo datoteko oz. model, ki je zmožen prepoznavanja objektov, za katere je bil treniran.

4.2.3 Nevronske mreže

Nevronske mreže so v zadnjih letih postale izjemno priljubljena metoda obdelave podatkov in oblikovanja umetne inteligence. Gre za računalniški model, ki posnema delovanje možganov in omogoča stroju, da se uči iz podatkov, prepoznava vzorce in napoveduje prihodnje dogodke.

Nevronske mreže se sestavljajo iz mreže umetnih nevronov, ki so med seboj povezani s sinapsami. Ti umetni nevroni so oblikovani tako, da sprejemajo vhodne podatke, jih procesirajo in izdajajo izhodne podatke. Vsak umetni nevron ima več parametrov, ki jih je treba programirati, da lahko mreža opravlja svojo funkcijo.

Učenje nevronske mreže se lahko izvaja na več načinov. Najpogosteje se

uporablja nadzorovano učenje, kjer se mreža poskuša naučiti določeno funkcijo na osnovi vhodnih in izhodnih podatkov. Tako se mreža postopoma prilagaja podatkom in izboljšuje svoje sposobnosti.

Prednosti nevronske mreže vključujejo sposobnost prepoznavanja kompleksnih vzorcev, prilagajanja novim podatkom in izboljševanja svojih sposobnosti s časom. Vendar pa so nevronske mreže tudi zahtevne za oblikovanje, učenje in interpretacijo, saj so pogosto črne škatle, kjer ni vedno jasno, kako so prišle do določenih zaključkov.

Nevronske mreže so ključne za razvoj umetne inteligence in napredka na številnih področjih. Z razvojem strojne opreme in algoritmov postaja njihova uporaba vse bolj razširjena in učinkovita.

4.3 Priprava slike

Ko imamo povečano izrezano sliko registrske tablice, jo moramo pripraviti, da jo lahko prepoznamo z bralnikom.



Slika 19: Približna slika registrske tablice.

Sliko pripravimo za prepoznavanje z OCR (Optical Character Recognition), podobno kot pri konturnici. Ugotovil sem da OCR teksta ne prebere natančno. Po nadaljnem raziskovanju sem ugotovil, da OCR lažje in bolj natančno bere slike, ki imajo označene samo robove, zato sem na slike nastavljal poseben filter.



Slika 20: Približana slika registrske tablice z posebnim filtrom.

Potem označimo še robove. Slika je tako pripravljena na branje.



Slika 21: Približana slika registrske tablice z označenimi robovi.

4.4 Segmentacija znakov

Segmentacija znakov je postopek ločevanja znakov na sliki, npr. črke in številke, od drugih elementov, kot so ozadje ali drugi predmeti. Ta korak nam omogoči, da se vsak znak obravnava ločeno in mu lahko dodelimo ustrezen pomen. Postopek segmentacije znakov običajno vključuje uporabo algoritmov računalniškega vida, ki temeljijo na analizi slik. Obstajajo različni pristopi k segmentaciji znakov, vključno z zaznavanjem robov, povezanih komponent in uporabo nevronske mreže. Nekatere tehnike lahko delujejo bolje pri določenih vrstah slik ali znakov, zato se za doseganje najboljših rezultatov pogosto kombinirajo različni pristopi. Ugotovil sem, da je najboljši pristop »threshold« metoda, ki primerja sosednje točke na sliki.

4.5 Prepoznavanje znakov

Prepoznavanje znakov (ang. character recognition) je proces, pri katerem računalnik z uporabo algoritmov umetne inteligence prepoznavlja natisnjene ali pisane znake in jih pretvarja v elektronsko obliko.

Postopek prepoznavanja znakov običajno vključuje zbiranje podatkov, ustvarjanje modela za učenje in prepoznavanje znakov ter končno izvedbo prepoznavanja.

Za to sem uporabil že narejen model.

4.6 Output/izhod

Po končanem prepoznavanju je rezultat programa array z vsemi črkami in številkami registrske tablice.

5 PROBLEM APLIKACIJE

Pri razvijanju programa za zajemanje in beleženje registrskih tablic z uporabo detekcije objektov in OCR sem se srečali s težavo, ki je vezana na varnost sistema. Čeprav se zdi sistem z uporabo prepoznavanja registrskih tablic z uporabo umetne inteligence in računalniškega vida zelo učinkovit, je še vedno možno, da zlonamerna oseba prelisiči sistem s pomočjo ponarejene registrske tablice.

V takem primeru bi lahko prevarant uporabil natisnjeno registrsko tablico, ki bi izgledala enako kot prava. Natisnjena pa bi bila na navaden papir in ne na poseben material, ki se uporablja pri izdelavi registrskih tablic. Ta problem predstavlja resno varnostno tveganje, saj bi lahko nepridiprav na ta način pridobil dostop do zaprtih območij in ogrozil varnost ljudi.

Da bi preprečili to tveganje, bi lahko uporabili več načinov za preverjanje pristnosti registrskih tablic, kot na primer optično branje črtnih kod in drugih posebnih znakov, ki se jih ne da enostavno ponarediti. Prav tako bi lahko uporabili kamere z višjo ločljivostjo, ki bi prepoznale manjše podrobnosti na registrskih tablicah in tako zmanjšale možnost ponarejanja.

Z uporabo teh metod bi lahko zmanjšali možnost zlorabe sistema ter zagotovili boljšo varnost in zaščito ljudi in njihove lastnine.

Poleg tega bi lahko uporabili tudi druge metode preverjanja pristnosti, kot na primer prepoznavanje obraza ali prstnih odtisov, ki bi omogočile bolj natančno preverjanje identitete osebe, ki poskuša vstopiti na zaprto območje. To pa bi seveda bilo v nasprotju z razlogi za razvoj takšne aplikacije.

5.1 Problemi pri razvijanju aplikacije

Pri razvijanju aplikacij sem se pogosto srečeval s težavami. Če bi opisal vse težave, na katere sem naletel, bi lahko napisal nalogo samo o tem. Osredotočil se bom na največje.

1. Problem z nalaganjem knjižnic (dependencies). Večina je bila neproblematičnih, nekaj pa se jih ni hotelo naložiti z ukazom pip. Pri teh sem moral brskati po internetu za pravilno verzijo. V okviru tega problema pa sem naletel še na enega. TensorFlow knjižnica ni podprta z Windows 11 in zato program ni deloval na mojem prenosnem računalniku.
2. Pri risanju kvadrata na sliko avtomobila je bil kvadrat premajhen in je segal samo od kota do kota registrske tablice in v posebnih primerih ni zajel celotne tablice.
3. Pri obračanju registrske tablice se kot nagiba ni zaznal in zato registrske številke ni bilo mogoče prebrati.
4. Pri segmentaciji se posamezni znaki niso ločili ali pa so bili popolnoma preskočeni.

Večino mojih težav sem odpravil z iskanjem rešitev na StackOverflow in podobnih forumih.

Pri pisanju kode, kot so zanke in podobne programske komponente, mi je pomagal GitHub Copilot.

6 IMPLEMENTACIJA SISTEMA

Za zajemanje registrskih tablic potrebujemo kamere, ki so sposobne zajeti kvalitetno sliko vozil, ki se gibljejo s primerno hitrostjo. Za namen zajemanja registrskih tablic se lahko uporabijo različni tipi kamer. Najpogosteje se razlikujejo po ločljivosti slike oz. videa. Najti je potrebno pravo ravnovesje med kvaliteto, ki mora biti dovolj velika, da se bo dalo iz nje razbrati številke, in velikostjo slik, ki bi vplivale na proces obdelovanja in velikost datotek v podatkovni bazi.

Ker bi se sistem izvajal v realnem času, bi se program poganjal na zmogljivem računalniku, ki bi bil sposoben izvajati kompleksne algoritme za detekcijo objektov in optično prepoznavanje znakov (OCR). Za ta namen se lahko uporabi zmogljiv računalniški sistem ali pa se izbere namenski računalnik z vgrajeno grafično kartico, ki bi bila namenjena izključno za obdelavo slikovnega materiala.

Glede na velikost parkirišča ali drugega področja, kjer bi se sistem uporabljal, bi bilo potrebno določiti optimalno število kamer, ki bi zagotovile celovit nadzor nad območjem. Poleg tega bi bilo potrebno namestiti ustrezno infrastrukturo, kot so kabli za napajanje, in povezavo z računalnikom ter morda tudi brezžično povezavo za daljšo razdaljo.

V primeru osnovne inštalacije bi morali zagotoviti vsaj eno kamero, ki bi jo bilo treba pravilno namestiti na primerno mesto za zajemanje registrskih tablic. Potrebno bi bilo zagotoviti tudi primerno strojno opremo za shranjevanje podatkov in uporabo programske opreme za zajemanje in obdelavo podatkov.

Poleg tega bi bilo treba upoštevati tudi delovno silo in čas, ki bi ga bilo treba nameniti za namestitev in konfiguriranje sistema. Končna groba cena stroškov inštalacije bi se gibala med 500 € in 700 €. Dejanska cena bi se spreminjala glede na velikost, ki bi jo moral sistem pokrivati, že vgrajene komponente in kvaliteto komponent.

7 ANALIZA ANKETE

Raziskavo sem izvedel s pomočjo spletnega vprašalnika v Microsoft Forms. Anketo je izpolnilo 67 dijakov. S to raziskavo sem želel ugotoviti kakšno mnenje imajo dijak do sistemov za prepoznavanje registrskih tablic.

7.1 Rezultati ankete

Ali bi pri vstopu na dvorišče oz. ZASEBNO parkirišče raje uporabljal RF daljinec ali avtomatsko prepoznavanje registrske tablice?



Graf 1: Mnenje anketirancev glede zasebne uporabe avtomatskega prepoznavanja registrskih tablic.

Večina anketirancev (66 odstotkov) je izjavila, da bi na zasebnih površinah raje uporabljali sistem za prepoznavanje registrskih tablic. Ostali so se odločili za uporabo RF daljinca.

Ali bi pri vstopu na JAVNO parkirišče oz. garažno hišo uporabljal avtomatsko prepoznavanje registrske tablice, pri čemer, bi upravljalcu moral zaupati registrsko tablico ter svoje ime in priimek?

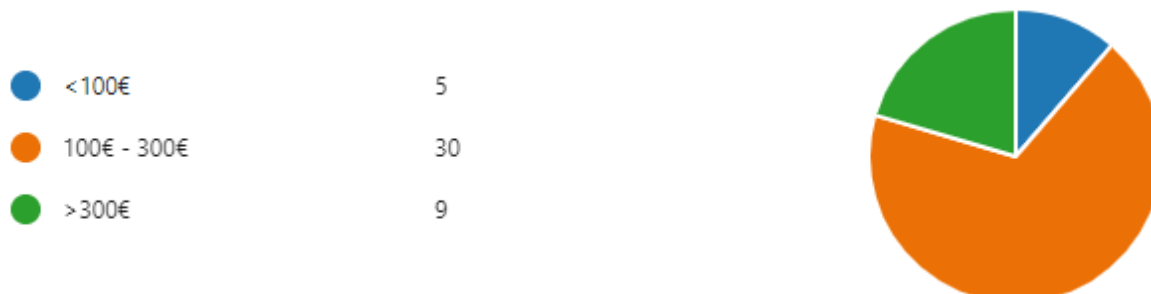


Graf 2: Mnenje anketirancev glede javne uporabe avtomatskega prepoznavanja registrskih tablic.

Čeprav bi večina anketiranih sicer bila pripravljena uporabljati sistem avtomatskega prepoznavanja registrskih tablic, so precej zadržani ob dejstvu, da bi za ta sistem morali upravljavcu zaupati svoje osebne podatke. Takšnih, ki bi brez zadržkov bili pripravljeni predati svoje ime in priimek je zgolj manjšina.

Je pa skupaj več kot polovica anketirancev takšnih, ki bi predvidoma ob primerni zagotovitvi varstva osebnih podatkov bili pripravljeni uporabljati moj sistem.

Koliko bi bil pripravljen plačati za namestitev takšnega sistema. (Cena vključuje kamero, programsko opremo, in vso potrebno delo inštalacije; vrata oz. zapornice niso vključena v ceno.)



Graf 3: Mnenje anketirancev glede cene inštalacije sistema.

Anketirance sem prosil tudi za finančno opredelitev. Tako se je večina opredelila, da bi bili za sistem pripravljeni plačati od 100 do 300 evrov; manjši del bi bil pripravljen investirati zgolj minimalno, nekaj več njih pa celo več kot 300 evrov, kar pa je sicer zgornja meja, ki niti ne bi bila potrebna za vzpostavitev sistema. Analiza je pokazala, da je sistem cenovno privlačen in bi naletel v tem smislu na pozitiven odziv.

8 ZAKLJUČEK

Med raziskovalno nalogo sem prišel do veliko novih spoznanj in izkušenj na področju umetne inteligence. Pri izdelavi programa sem naletel na številne težave, a sem jih na koncu le nekako uspel rešiti. Nisem dosegel svojega cilja, saj sem si zadal, da bom lahko aplikacijo popolnoma dokončal. Naredil sem le njen najbolj pomemben del.

Delal sem sam, ker sem mislil, da bo takšen način dela najbolj učinkovit, saj se mi ni bilo treba usklajevati z drugimi člani ekipe. Kljub temu sem ponosen na svojo raziskovalno nalogo. Če bi jo moral delati še enkrat, bi se je lotil bolj sistematično in morda jo delal z nekom.

8.1 Hipoteze

1. *Uporaba avtomatskega prepoznavanje registrskih tablic je dovolj zanesljivo za uporabo.*

Rezultati raziskave so pokazali, da ima moj model težave prepoznati določene registrske tablice. Pri slovenskih registrskih tablicah ima težavo, saj poleg drugih črk zazna tudi grb mesta. Zaradi tega problema hipoteze ne moram potrditi.

2. *Za uporabnika je bolje, če uporablja avtomatsko prepoznavanje registrskih tablic kot, če bi imeli daljinec.*

S pomočjo ankete sem lahko to hipotezo potrdil, saj je večina udeležencev ankete izbrala prvo opcijo.

3. *Vgraditev tega sistema na dvorišča in v garažne hiše je finančno izvedljivo.*

Večina anketirancev bi bila za vgraditev sistema pripravljeno plačati 100-300 € kar je tudi realna cena. Zato lahko to hipotezo potrdim.

8.2 Možnost nadaljnje raziskave

Prvi korak je dokončati aplikacijo. To bi vključevalo uporabniški vmesnik, podatkovno bazo, morda spletno aplikacijo in podobne stvari. Program bi rad preizkusil tudi na zapornici. Za to pa bi moral program prilagoditi za uporabo kamer v realnem času. Pri tem pa bi moral tudi izboljšati model, za hitrejšo prepoznavanje registrskih tablic.

Pri nadaljnjem razvijanju prepoznavne registrskih tablic pa bi bila možnost zaznave več tablic iz ene slike. Moj sistem namreč tega ni sposoben. Aplikacijo bi lahko integriral v mobilne telefone s katerimi bi lahko pooblašcene osebe upravljale pri nadzoru prometa.

9 VIRI IN LITERATURA

ANPR international. (6. 3 2022). Pridobljeno 6. 3 2023 iz ANPR international:
<http://www.anpr-international.com/history-of-anpr/>

Data Science Central. (7. 9 2021). Pridobljeno 6. 3 2023 iz Data Science Central:
<https://www.datasciencecentral.com/a-step-by-step-guide-to-ai-model-development/>

Geeksforgeeks. (22. 1 2022). Pridobljeno 6. 3 2023 iz Geeksforgeeks:
<https://www.geeksforgeeks.org/detect-and-recognize-car-license-plate-from-a-video-in-real-time/?ref=gcse>

MakeUseOf. (10. 1 2023). Pridobljeno 6. 3 2023 iz
<https://www.makeuseof.com/python-car-license-plates-detect-and-recognize/>

National Library of Medicine. (26. 5 2021). Pridobljeno 6. 3 2023 iz National Library of Medicine:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8123416/>

pyImageSearch. (21. 9 2020). Pridobljeno 6. 3 2023 iz pyImageSearch:
<https://pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-anpr-with-python/>

Renotte, N. (3. 5 2021). *YouTube*. Pridobljeno 6. 3 2023 iz YouTube:
https://www.youtube.com/watch?v=0-4p_QgrdbE&t=343s

Section. (31. 12 2021). Pridobljeno 6. 3 2023 iz Section:
<https://www.section.io/engineering-education/license-plate-detection-and-recognition-using-opencv-and-pytesseract/>

towardsdatascience. (9. 1 2020). Pridobljeno 6. 3 2023 iz towardsdatascience:
<https://towardsdatascience.com/how-to-create-your-own-custom-object-detector-766cb11ccd1c>

Wikipedia. (26. 2 2023). Pridobljeno 6. 3 2023 iz Wikipedia:
https://en.wikipedia.org/wiki/Artificial_intelligence

10 PRILOGE

10.1 Priloga 1: Anketni vprašalnik

SISTEM ZA PREPOZNAVANJE REGISTRSKIH TABLIC

1. Starost *

- ☐ 15
- ☐ 16
- ☐ 17
- ☐ 18+

2. Ali bi pri vstopu na dvorišče oz. ZASEBNO parkirišče raje uporabljal RF daljinec ali avtomatsko prepoznavanje registrske tablice? *

- ☐ RF daljinec
- ☐ Avtomatsko prepoznavanje registrske tablice

3. Koliko bi bil pripravljen plačati za namestitev takšnega sistema. (Cena vključuje kamero, programsko opremo, in vso potrebno delo inštalacije; vrata oz. zapornice niso vključena v ceno.) *

- ☐ <100€
- ☐ 100€ - 300€
- ☐ >300€

4. Ali bi pri vstopu na JAVNO parkirišče oz. garažno hišo uporabljal avtomatsko prepoznavanje registrske tablice, pri čemer, bi upravljalcu moral zaupati registrsko tablico ter svoje ime in priimek? *

- ☐ DA
- ☐ NE
- ☐ Odvisno od upravljalca površin

Priloga 1: Anketni vprašalnik.