Prototip sustava za ulaz u garažu korištenjem računalnog vida

Calvi, Karlo

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:195:708840

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-18**

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

zir.nsk.hr

DIGITALNA KNJIŽNICA

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Sveučilišni diplomski studij informatika – modul Poslovna informatika

Karlo Calvi

Prototip sustava za ulaz u garažu korištenjem računalnog vida

Diplomski rad

Mentor: doc. dr. sc, Miran Pobar

Rijeka, prosinac 2023.

Sažetak

U radu je opisana izrada prototipa sustava za ulaz u garažu korištenjem računalnog vida. Prikupljene su i označene slike za učenje modela za detekciju registarskih pločica i testiranje sustava. Za detekciju pločica na prikupljenom skupu od 1379 slika naučen je model SSD ResNet 50. Nakon detekcije, na slike je nakon dodatne obrade je primijenjen OCR u svrhu očitanja registarskih oznaka. Očitani tekst registracije je proslijeđen funkciji koja metodom Levenshteinove udaljenosti traži najbolje podudaranje s tekstom iz popisa registracija s dozvolom ulaska u garažu. Iako na OCR utječu razna obilježja registracija, sustav je postigao preciznost od 98.11% i točnost od 96.36%. Korištenjem Python biblioteke Gradio izrađen je prototip aplikacije sustava za ulaz u garažu koja uz ostalo omogućava uređivanje liste dozvoljenih tablica i pregled liste ulaska u garažu.

Ključne riječi:

OCR, registarske oznake, registracije, Levenshteinova udaljenost, pretprocesiranje slika

Sadržaj

1. Uvod ... 1
2. Računalni vid ... 2
   2.1. Detekcija objekata i optičko prepoznavanje znakova ... 2
3. Pregled literature ... 4
   3.1. Prepoznavanje registarskih oznaka ... 4
   3.2. Pregled postojećih sustava za prepoznavanje registarskih oznaka ... 5
4. Izrada prototipa sustava ... 8
   4.1. Registarske pločice ... 8
   4.2. Programi i alati ... 10
      4.2.1. Alati i biblioteke ... 11
   4.3. Opis metoda ... 12
   4.4. Priprema podataka i učenje modela za detekciju pločica ... 14
      4.4.1. Priprema podataka ... 14
      4.4.2. Označavanje slika ... 16
      4.4.3. TFRecords i Labelmap datoteke ... 17
      4.4.4. Konfiguriranje modela ... 18
      4.4.5. Treniranje modela ... 18
   4.5. Algoritam za očitavanje registarskih pločica ... 20
      4.5.1. Pretprocesiranje slika ... 20
      4.5.2. Funkcija *rotate_image* ... 24
      4.5.3. Prepoznavanje registarskih oznaka ... 24
      4.6. Prototip aplikacije ... 26
      4.6.1. Testiranje i evaluacija ... 28
5. Rezultati ... 30
6. Diskusija ... 33
7. Zaključak ... 35
8. Reference ... 36
9. Popis priloga ... 39

1. Uvod

Tema koju obrađuje ovaj diplomski rad odnosi se na izradu prototipa sustava za ulaz u garažu upotrebom tehnologije računalnog vida. Cilj projekta je automatizacija otvaranja garažne rampe korištenjem računalnog vida za detekciju i očitavanje registarskih oznaka vozila.

S obzirom na suvremeni način života, svi imaju potrebu za prijevoznim sredstvom te tako danas malo tko ne posjeduje vlastito vozilo. U toj gomili vozila nužno je osigurati što više parkiranih prostora, a isto tako tim prostorima potrebno je kvalitetno, efikasno i organizirano upravljati. Problem koji ovaj projekt rješava odnosi se na upravljanje parkirnim garažama sveučilišnog kampusa u Rijeci. Implementacijom sustava zasnovanog na detekciju objekata i optičkom prepoznavanju teksta, sustav će omogućiti lako i precizno identificiranje vozila na ulasku i izlasku iz garaže. Na taj način bi se olakšao ulazak u garažu profesorima, zaposlenicima i ostalim korisnicima garaže koji imaju ovlaštenje ulaska. Svaki ulazak i izlazak bi se tako bilježio u bazi podataka što bi ujedno služilo i za evidenciju dolazaka i odlazaka s posla.

Ovim projektom omogućilo bi se lakše upravljanje pristupom vozila, poboljšanje popunjenosti parkirnih mjesta uz istovremeno učinkovito i brzo evidentiranje dolazaka zaposlenika na posao.

2. Računalni vid

Računalni vid je znanstveno područje čiji se početci vežu za rane 1970-e, za svrhu ima oponašanje ljudskog vida uporabom računala. Ono podrazumijeva učenje, poduzimanje radnji na osnovu vizualnih unosa te sposobnost donošenja odluka. Područje spada u granu umjetne inteligencije (AI) koja za cilj ima oponašati ljudsku inteligenciju [1]. Računalnim vidom se kroz jednu ili više slika pokušava opisati svijet, na način kako ga ljudi vide i rekonstruirati mu svojstva poput oblika, svjetlosti, boja itd.

Područje sadrži razne tehnike i ima široku primjenu u stvarnom svijetu među kojima su: optičko prepoznavanje znakova (OCR), medicinska dijagnostika, autonomna vozila, biometrijsko prepoznavanje i sl. [2]. Tako Hull u svom radu govori o upotrebi računalnog vida za prepoznavanje ručno ispisanih poštanskih brojeva, naziva gradova i država s pošiljki.[3]

2.1. Detekcija objekata i optičko prepoznavanje znakova

Za svrhu ovog rada najvažnije su potkategorije računalnog vida: optičko prepoznavanje znakova i detekcija objekata.

Cilj detekcije objekata je računalnim vidom lokalizirati i identificirati objekte na slikama ili videozapisima. Koristi se u raznim primjenama među kojima su klasifikacija slika, prepoznavanje lica, prepoznavanje prometnih znakova i sl. [4]. Primjerice u medicini se detekcija objekata široko koristi od brojanja zubi s rendgenskih (RTG) snimki, prebrojavanja i klasificiranja krvnih stanica pa sve do detekcije i klasifikacije tumora s RTG ili CT slika. [5]

Optičko prepoznavanje znakova (OCR) je postupak automatske klasifikacije optičkih uzoraka s digitalne slike, koji odgovaraju znakovima poput slova, brojki, specijalnim znakovima i sl., radi pretvaranja u strojno čitljiv tekst [6]. OCR sustavi se prema načinu prikupljanja slika, povezivosti znakova, ograničenjima fonta i sličnom mogu kategorizirati na prepoznavanje rukopisa i prepoznavanje strojno ispisanih znakova [7]. Ovaj projekt se oslanja na detekciju registarskih pločica i OCR strojno ispisanih znakova s detektirane pločice.

Glavne faze OCR sustava odnose se na: prikupljanje slika, pretprocesiranje slika, izdvajanje znakova, izdvajanje tekstualnih značajki, klasificiranje znakova te naknadnu obradu teksta [7]. Takve faze OCR-a prikazane su i u sustavu koji su izradili Kumar i

suradnici. Njihov sustav skenira slike računa i faktura te ih obrađuje, a zatim očitava pomoću OCR-a kako bi digitalizirali i pohranili podatke s tih dokumenata. [8]

3. Pregled literature

3.1. Prepoznavanje registarskih oznaka

Pregledom literature istraženi su različiti pristupi rješavanju problema prepoznavanja registarskih oznaka vozila. S obzirom da je riječ o području koje ima široku primjenu u svakodnevnom životu kao što je nadzor prometa, kontrola pristupa parkiralištima, naplata cestarine itd. ključno je dobiti točne podatke. Prema tome mnogi radovi bave se ovom temom te istražuju različite pristupe i metode kojima se dolazi do rješenja.

Različiti izazovi otežavaju postupak detekcije i raspoznavanja registarskih oznaka, a oni se mogu podijeliti na varijacije registarskih pločica i varijacije okruženja. Što se tiče varijacija registarskih pločica, one su sljedeće:

- lokacija: položaj registracije se razlikuje na slikama

- količina: slika može sadržavati nekoliko registracija ili niti jednu.

- veličina: registarske pločice mogu biti različite veličine zbog udaljenosti od kamere

- boja: registracije mogu biti različite boje znakova i pozadine ovisno o vrsti ili kameri

- font: različitost veličine i stilova teksta

- standard: svaka država ima svoj raspored i kombinacije znakova na registraciji

- zaklonjenost teksta: moguća su zaprljanja područja registracije te prekrivanje dijela teksta

- nagib: nakrivljene registarske pločice teže je ispravno očitati

- ostalo: mogući su natpisi na okvirima pločica ili vijci.

Varijacije okruženja odnose se na pozadinu (okolinu) i osvjetljenje. Pozadina oko registarske pločice može sadržavati razne uzorke slične registraciji poput natpisa i reklama, branika s okomitim uzorcima, prometnih znakova, podnih natpisa i sl. [9].

Sustavi za prepoznavanje registarskih oznaka prema pristupu dijele se na jednostupanjske i višestupanjske sustave. Jednostupanjski sustavi rade na principu duboke neuronske mreže obučene za detekciju, lokalizaciju i prepoznavanje registarske pločice u jednom prolazu. Višestupanjski sustavi podijeljeni su na tri glavna koraka, a to su

- detekcija i/ili izdvajanje registarske pločice sa slike

- pretprocesiranje izdvojene pločice i/ili izdvajanje tekstualnih znakova sa slike

- primjena tehnika za raspoznavanje znakova [10].

Metode detekcije registarske pločice zasnovane su na detekciji rubova pločice, identifikaciji boje ili tekstura, ispitivanju prisutnosti i lokaliziranju znakova, statističkim klasifikatorima ili tehnikama dubokog učenja. Kao tehnike pretprocesiranja izdvojene registarske pločice najčešće se koriste tehnike ispravljanja rotacije, tj. nagiba registracije, poboljšanja kvalitete slike i binarizacija slike tj. promjena slike samo u crne i bijele piksele. Za ispravljanje nagiba često se koristi bilinearna transformacija, metode prilagodbe linija i druge metode zasnovane na najmanjim kvadratima. Za poboljšanje kvalitete slike koriste se filteri za uklanjanje šuma, tehnike poboljšanja kontrasta, redistribucija intenziteta piksela izjednačavanjem histograma kao i manipulacija intenzitetima sivih piksela. Kod binarizacije slike najčešće se koriste metode prilagodljivog binarnog ograničavanja kao što je lokalno ograničavanje i Niblacksova metoda binarizacije kojom se koristi standardna devijacija i srednja vrijednost oko svakog piksela. Izdvajanje znakova može se provoditi prema povezanosti piksela, pomoću projekcijskih profila, korištenjem prethodnog znanja kao što su omjeri znakova, omjeri boja piksela i sl. te korištenjem dubokog učenja. Nakon toga može se provesti prepoznavanje znakova tehnikama spajanja segmenata s predloškom, tehnikama izdvajanja značajki ili korištenjem dubokog učenja [10].

Iz ranije navedenih razloga prepoznavanje registarskih oznaka može biti poprilično složen i izazovan posao te ne postoji jedinstveno rješenje ovog problema. Zbog toga su mnogi autori razvili svoje pristupe te su kombinacije tehnika prilagodili svojim potrebama.

3.2. Pregled postojećih sustava za prepoznavanje registarskih oznaka

Razvijeno je mnoštvo sustava kojima je zadaća prepoznati registarsku oznaku, a većina ih se zasniva na istom konceptu i sličnim korištenim tehnikama. Međutim, zbog različitih potreba, koje svaki sustav mora zadovoljiti, razlikuju se i njihove izvedbe. Nekoliko takvih sustava opisano je kroz ovo poglavlje.

Rad sustava koji su razvili Islam i suradnici [7] započinje prosljeđivanjem slike s kamere u model za detekciju, dobivena slika se zatim izdvaja kako bi se s nje mogli izdvojiti i prepoznati tekstualni znakovi. Metode koje su primijenili za olakšanje detekcije registarske pločice odnose se na početno zumiranje slike (5x zoom) kako bi smanjili pojavu nepotrebnih informacija iz okoline. Na dobivenoj slici poboljšan je kontrast izjednačavanjem histograma. S tako prilagođene slike pomoću YOLO mreže[11] dubokog učenja je detektirana, a zatim i izdvojena registarska pločica. Na izdvojenu registraciju

primijenjen je histogram izjednačenja razine sivih piksela, a zatim binarizacija globalnim pragom. Potom su prije izdvajanja znakova primijenjeni median filter i „sol i papar" filter šuma. Median filter uzima vrijednosti svih piksela iz određenog područja i zamjenjuje vrijednost trenutnog piksela njihovom srednjom vrijednosti, dok „sol i papar" nasumično postavlja postotak piksela na crnu ili bijelu boju radi oponašanja šuma slike. Tekst je zatim izdvojen korištenjem CCA metode koja grupira međusobno povezane piksele u skupine zasebnih komponenata. Prepoznavanje teksta omogućeno je histogramom orijentiranih gradijenata kojim su dobivene značajke svakog izdvojenog tekstualnog znaka. Prema dobivenim značajkama istrenirana je neuronska mreža te je pomoću tih podataka prepoznat tekst [12].

Sljedeći primjer sustava koji su razvili Ahmed i suradnici [13] dohvaća sliku s kamere te ju uspoređuje s pohranjenim slikama iz baze podataka radi provjere prava ulaska u garažu. Nakon što se u sustav proslijedi slika s kamere na nju se primjenjuju različite tehnike obrade. Slika se prvo pretvara u sive tonove, a potom se binarizira kako bi bila pogodnog oblika za usporedbu svakog pojedinog znaka s referentnom slikom. Slika se dodatno poboljšava morfološkim operacijama radi uklanjanja raznih nečistoća i rupa u znakovima te se nakon toga izdvajaju znakovi i uspoređuju s popisom dozvoljenih vozila [13].

Sljedeće, Al-Mheiri i suradnici [14] prikazali su drugačiji pristup detekciji objekata pomoću detekcije kontura. Naime, u ovom sustavu nakon što se zaprimi slika s kamere prvo se izvršava obrada nad njom. Na sliku se primjenjuje konverzija u sive tonove pa se zatim svjetlina povećava za 50% radi uklanjanja sjena. Nakon toga primjenjuje se uklanjanje šuma Gaussovim zamućenjem koje zaglađuje slike nakon čega se primjenjuje Blackhat transformacija. Blackhat je tehnika koja čini crna područja iznad bijelih izraženijima te na taj način registracije s bijelim pozadinama postaju istaknutije. Idući korak je primjena morfološke operacije zatvaranja koje će proširiti bijela područja, a zatim ih vratiti u izvornu veličinu, time slika postaje pogodnija za prepoznavanje kontura. Slika se potom binarizira pa se na takvu sliku primjenjuju postupci erozije i dilatacije. Erozijom se sužavaju rubovi objekata na slici, a dilatacijom se proširuju Kada su svi navedeni koraci ispunjeni slika postaje pogodna za detekciju kontura. Pomoću OpenCV biblioteke detektiraju se konture koje se sortiraju prema njihovom području, uz zadržavanje samo prvih deset na kojima se dodatno uklanja šum. Svaka kontura sadrži granični okvir s koordinatama pozicije. Za svaki okvir pronalazi se njegov centar, ako je centar okvira između 30% i 70% od ishodišta x-osi i između 50% i 90% od ishodišta y-osi prihvaća se kao kandidat za prepoznavanje teksta.

Kada je detekcija izvršena na „blackhat” sliku se primjenjuje Canny detekcija rubova s koje se izdvajaju konture koje zapravo izdvajaju tekstualne znakove. Konture se sortiraju i 40% najmanjih se odbacuje jer se obično odnose na šum i dodatno se uklanjaju sve one koje se ne sijeku s kandidatima za registarsku pločicu. Za preostale konture izračunava se njihov omjer koji mora biti između 0.15 i 0.90. Konturama se također izbacuju svi oblici unutar znakova te se one jedna po jedna izdvajaju s „blackhat” slike u novu sliku kojoj se invertiraju boje i konačno na njih se primjenjuje OCR [14].

4. Izrada prototipa sustava

Kroz ovo poglavlje detaljno je opisana odabrana metodologija izrade prototipa i postupak izrade sustava za ulaz u garažu korištenjem računalnog vida. Cilj rada bio je razvoj efikasnog algoritma za prepoznavanje registracijske oznake automobila s visokom točnosti. Izrađeni algoritam potrebno je implementirati u demo aplikaciju koja čini osnovni prikaz sustava za omogućavanje ulaska u garažu ovlaštenim vozilima.

Kako bi se izradio sustav, to jest algoritam, koji daje dovoljno precizne rezultate prepoznavanja registarske oznake, potrebno je odabrati odgovarajuće metode. S obzirom na tematiku ovog rada ključan je bio izbor metoda čitanja teksta sa slika i predobrade slika za isti postupak. Uz to, važno je i kako sa cijele slike dobiti samo segment koji sadrži registarsku pločicu.

Pregledom znanstvenih radova istraženi su različiti pristupi rješavanja problema ovog područja. U izradi algoritma primijenjene su različite metode od kojih su najbrojnije one za obradu slike. Sveukupno, većina tih metoda i tehnika već je korištena u brojnim radovima, ali sama kombinacija metoda korištenih u ovom radu dobivena je pomoću pokušaja i pogrešaka. Isti slučaj je i s izborom vrijednosti različitih parametara funkcija. Dakle, odluka o izboru metoda i tehnika koje će se koristiti isključivo je temeljena na dobivanju što boljeg rezultata.

4.1. Registarske pločice

Registarske pločice, odnosno registarske oznake su sredstvo identificiranja vozila i vlasnika u prometu. To su alfanumeričke oznake u različitim kombinacijama uz dodatak oznake države. Specifičnosti poput slovno-brojčanih kombinacija, formata, boje slova i pozadine, dimenzije i dr. ovise o svakoj državi. Smještene su na prednju i (ili) stražnju stranu vozila. Isto tako, vrsta vozila koja se registriraju kao i vrsta registarske pločice koja im se dodjeljuje, razlikuje se od države do države.

U svijetu postoji jako velik broj različitih dimenzija i karakteristika registarskih pločica vozila, a u Hrvatskoj su one pravokutnog oblika, dimenzija 520 mm x 110 mm (520x120 mm u dijelu EU) što odgovara većini europskih država i mnogim njihovim bivšim prekomorskim teritorijima.

Iako svaka država samostalno odlučuje o izgledu registarske pločice, u većini država članica Europske Unije (EU) one su uobičajeno bijele ili žute reflektirajuće pozadine s crnim slovima. EU standard s lijeve strane pločice sadrži i plavu traku s motivom EU i

međunarodnu oznaku države. U Hrvatskoj, a i u nekim drugim državama, još uvijek se koriste i registarske pločice bez oznaka EU [15].

Standardne hrvatske registarske oznake (Slika 1.) sastavljene su od dvoslovne oznake registracijskog područja, kojih je 34, nakon koje je otisnut grb Republike Hrvatske (RH) zatim tri ili četiri brojke nakon kojih je jedno ili dva slova. Dio registarske oznake koji sadrži slova ne može sadržavati slova Č, Ć, Đ, Š i Ž izuzev početne oznake registracijskog područja, a brojčane vrijednosti mogu biti u znamenkama od 0 do 9. Kako u Hrvatskoj tako i u Europi, prema dostupnim podacima, registarska oznaka ne može biti kraća od pet znakova.

Vlasnici vozila registarsku pločicu mogu zatražiti po izboru, osim oznake područja i grba RH one mogu sadržavati od tri do sedam slova ili brojki ili njihove kombinacije. Kod odabira brojčane oznake obvezno je uključiti najmanje jedno slovo odvojeno crticom. Diplomatske i konzularne registarske pločice u RH plave su boje, a njihov tekst je žute boje uz dodatak brojčane oznake zemlje i slovne oznake djelatnosti i statusa predstavništva.

Još jedna vrsta registarske pločice u RH su prenosive pločice na njima se nakon grba RH nalaze slova PP i tri ili četiri brojke. Vozila policije bijele su boje s plavim brojevima, a sadrže dvije trobrojne kombinacije razdvojene grbom RH. Vojna vozila imaju žutu registarsku pločicu, početnu oznaku HV nakon koje slijedi grb RH i tri ili četiri broj pa jedno ili dva slova.

Slika 1. Primjer standardne registarske pločice u RH. Izvor: Narodne Novine.

U hrvatskoj postoje i pokusne registarske pločice za obavljanje tehničkog pregleda neregistriranih ili odjavljenih vozila, one su izrađene od laminata (ljepljive folije) standardnog formata uz natpis „PROBA” s krajnje lijeve strane.

Stranci s odobrenim privremenim ili stalnim boravkom u RH kao i strani trgovinski, kulturni i drugi predstavnici za privremeno registrirano vozila dobivaju registarsku pločicu standardnog formata, ali su im slova i brojke zelene boje.

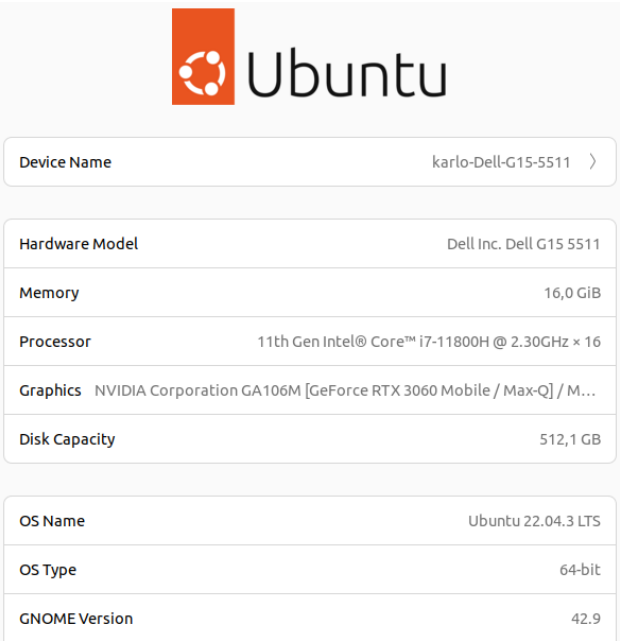
Vozila koja ne odgovaraju propisanim uvjetima prema dimenzijama, tj. s najvećom dopuštenom masom većom od propisane, sadrže slova i brojke crvene brojke na standardnom formatu registarske pločice [16]. Postoji još nekoliko vrsta registarskih pločica u RH, ali one se rijetko pojavljuju pa u ovom nisu od velike važnosti za izradu ovog rada.

Osim Hrvatske, još neke države imaju otisnut grb države ili neku vrstu oznake nakon početnog teksta što, važno za ovu temu, otežava izdvajanje točnog teksta registarske oznake, kao i različiti natpisi na držačima registarske pločice. Za postupak prepoznavanja registarske oznake najpogodnije su one registarske oznake koje ne sadrže dodatne oznake koje razdvajaju tekst. Radi toga ključno je bilo pronaći način za uspješno prepoznavanje registarskih oznaka koje baš takve oznake sadrže, radi dobivanja pouzdanijeg sustava.

4.2. Programi i alati

Cjelokupna izrada programskog koda, treniranje modela kao i sve potrebne instalacije odrađene su na Ubuntu Linux distribuciji. Izrađeni kod moguće je pokretati i na drugim operacijskim sustavima kao i u Google Colab bilježnici. Također, sav potrebni postupak moguće je napraviti i u OS Windows bez većih razlika u postupku.

Korišteni hardver i OS za izradu ovog projekta vidljiv je na Slici 2. Kod hardvera su prvenstveno bitni CPU ili GPU i RAM memorija zbog popriličnog opterećenja računala prilikom treniranja TensorFlow modela.



Slika 2. Prikaz hardvera korištenog za izradu ovog projekta.

4.2.1. Alati i biblioteke

Kako bi se mogao koristiti model za detekciju registracija te izraditi potrebni algoritam instaliran je Python, PyCharm interpreter, TensorFlow te razne biblioteke. Uz to korišten je i LabelImg alat za anotiranje slika.

Programski jezik Python široko se koristi u području umjetne inteligencije, a samim time i kod detekcije objekata i računalnog vida kao i obrade podataka. Koristi se zbog jednostavnosti, preglednosti i čitljivosti koda te raznolikosti biblioteka. Osim toga, za ovaj projekt je odabran jer omogućuje brzo prototipiranje, jednostavno testiranje i prilagodljivost. Python je korišten u razvojnom okruženju PyCharm koji je vrlo popularan i kvalitetan alat koji nudi mnoge alate, virtualno okruženje, podršku za razne tehnologije i sl. što olakšava razvoj Python programa i aplikacija.

TensorFlow je biblioteka otvorenog koda koja pruža širok izbor alata, biblioteka i resursa za izradu i treniranje raznih modela strojnog učenja što uključuje i računalni vid. Pruža velik broj predtreniranh modela za detekciju i klasifikaciju objekata poput SSD [17], Faster R-CNN [18], EfficientDet [19] i dr., a ujedno omogućava evaluaciju i vizualizaciju performansi modela u web aplikaciji TensorBoard.

EasyOCR[20] je biblioteka za prepoznavanje teksta sa slika. Podržava preko 80 svjetskih jezika. Brzo i efikasno prepoznaje tekst i sa slika lošije kvalitete, a također omogućava prepoznavanje teksta u stvarnom vremenu.

OpenCV [21]je biblioteka za obradu slika i računalni vid. Omogućuje razne manipulacije slika poput izmjene boja, prepoznavanje uzoraka, detekciju rubova, binarizaciju i inverziju slike itd.

NLTK [22] je biblioteka za rad s tekstualnim podacima. Pruža mogućnost korištenja različitih funkcija, između ostalog i Levenshteinove udaljenosti [23] koja mjeri minimalni broj uređivanja potrebnih za pretvaranje jednog niza u drugi.

Matplotlib [24] je biblioteka za vizualizaciju podataka kroz razne vrste grafova i dijagrama.

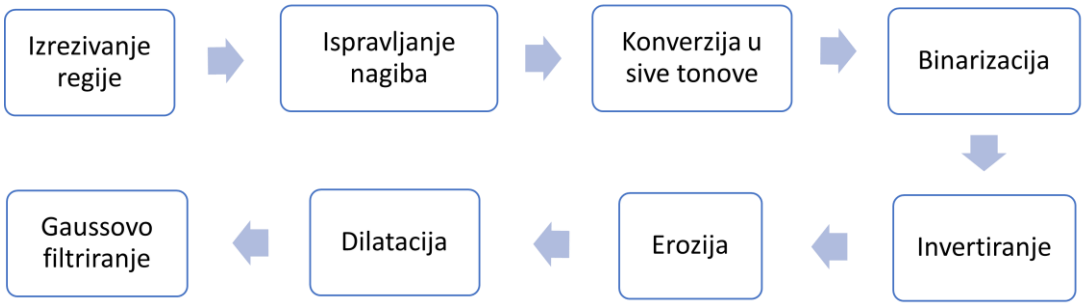
LabelIMG [25] je jednostavan alat za anotiranje područja objekata na slikama u svrhu stvaranja skupova podataka koji se koriste kod treniranja TensorFlow i drugih modela strojnog učenja. Omogućuje dodavanje pravokutne oznake oko objekta i naziva klase. Za svaku sliku tada kreira XML datoteku s podacima o dimenzijama, nazivu, klasi i poziciji oznake.

Gradio je Python biblioteka koja omogućuje jednostavnu izradu prototipova interaktivnih aplikacija koje koriste model strojnog učenja i njihovo dijeljenje. S Gradiom, aplikacije se izrađuju brzo, jednostavno i u svega nekoliko linija koda, a korisnicima omogućuje pokretanje aplikacije u web pregledniku.

4.3. Opis metoda

Za izradu algoritma za detekciju i očitavanje pločica korištene su sljedeće metode. Za prepoznavanje teksta s registarskih oznaka korišten je EasyOCR, Python biblioteka otvorenog koda. Tijekom izrade algoritma testirane su dvije različite metode za dobivanje teksta sa slike, EasyOCR i Tesseract. Zbog bolje učinkovitosti i pokazanih rezultata kao i određenih postavki u konačnici je odabran EasyOCR. U postavkama Reader klase iz paketa EasyOCR mogu se koristiti metode readText, detect i recognize. Svaka od tih metoda omogućuje upravljanje različitim postavkama koje utječu na rezultat prepoznavanja teksta. Opcije koje se mogu mijenjati su razne, a neke od njih su lista dopuštenih ili zabranjenih znakova, minimalna veličina teksta, kontrast, minimalni prag uvjerenosti u rezultat i sl. U ovom sustavu korištene su metode „detail=0" što znači da se za rezultat ispisuje samo tekst, „paragraph=true" tj. spajanje svih rezultata u odlomak, lista dozvoljenih znakova od A do Z i od 0 do 9 te praga uvjerenosti u rezultat od 0.9 [20].

Kod obrade slika korištene su mnoge metode, a važnije među njima su binarizacija, erozija, dilatacija, invertiranje, Gaussovo zamućivanje i Houghova transformacija. Redoslijed primjene obrade slika prikazan je na Slici 3.



Slika 3. Dijagram slijeda obrade slika.

Binarizacija, odnosno binarno ograničavanje je postupak pretvaranja slike sivih tonova u binarni oblik. Uz konverziju slike u sive tonove spada među najosnovnije načine obrade slika. U binarnim slikama svi pikseli ispod definiranog praga se postavljaju na nulu, dok se

svi pikseli iznad tog praga postavljaju na jedan. Na taj način pikseli postavljeni na nulu predstavljaju crnu boju, a oni postavljeni na jedinicu predstavljaju bijelu boju [26].

Ta metoda se može primijeniti iz nekoliko razloga, a to su poboljšanje kontrasta, razdvajanje objekata od pozadine na slici, jednostavnija analiza slike i sl., a sve u svrhu boljeg izdvajanja detalja. Tehnike binarizacije se može klasificirati u dvije grupe, globalno i lokalno ograničavanje, a neke od poznatijih tehnika su jednostavno ograničavanje, globalno ograničavanje, Otsu, prilagodljivo ograničavanje [27]. U izradi algoritma primijenjena je kombinacija jednostavnog i Otsu ograničavanja.

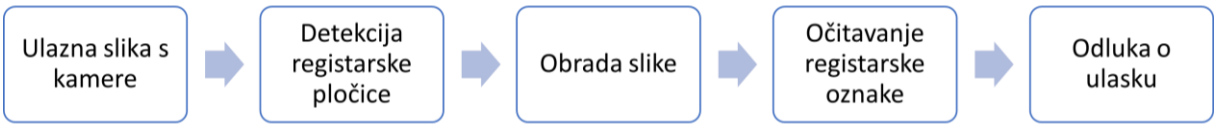
Nakon binarnog ograničavanja na dobivenu sliku primijenjeno je invertiranje slike i odnosi se na promjenu boja tj. intenziteta piksela u suprotno. Dakle, kod binarne slike zamjenjuju se nule i jedinice pa pikseli koji su crni postaju bijeli, dok bijeli pikseli postaju crni [28], [29]. Na taj način dobivena su bijela slova na crnoj pozadini što je u ovom sustavu dodatno poboljšalo prepoznavanje teksta.

Sljedeći primijenjeni postupci su morfološke operacije erozije i dilatacije. Dilatacija služi za proširenje i povećanje rubova tekstualnih znakova na slici. Njome se povećavaju i podebljavaju rubovi (dodavanjem piksela) objekata čime se smanjuju rupe unutar tih područja, time se olakšava prepoznavanje teksta sa slike. Uglavnom se koristi u kombinaciji s erozijom koja ima suprotan učinak odnosno ona sužava (stanjuje) objekte na slici. [28].

Zatim je na tako obrađenu sliku dodatno primijenjen Gaussov filter šuma. Taj filter zapravo primjenjuje zamućenje na sliku te ju time zaglađuje smanjujući šum i zanemarive detalje pomoću Gaussove funkcije. Slika je tako manje pikselizirana. Radi tako da se pikselima bližim središtu filtera dodaju veće, a udaljenijima manje težine. Tako se prilikom zaglađivanja uzimaju u obzir susjedni pikseli pri određivanju stupnja zamućenosti [30].

Osim tehnika koje utječu na kvalitetu slike i jasnoću teksta korištena je i Houghova metoda transformacije kako bi se, na onim slikama kod kojih je moguće, ispravila rotacija objekta u svrhu izravnanja teksta. Zbog neosjetljivosti na šum i isprekidane linije pogodna je i za slike lošije kvalitete [8].

Algoritam se može podijeliti na više dijelova (Slika 4.) ovisno o svrsi, a to su: Učitavanje potrebnih datoteka, modela i slika, Detekcija registarskih pločica, Pretprocesiranje slika za primjenu računalnog vida, Prepoznavanje registarskih oznaka, Sučelje prototipa aplikacije, Testiranje i evaluacija te Rezultati.



Slika 4. Dijagram toka podataka od ulazne slike do odluke o ulasku očitane registracijske oznake.

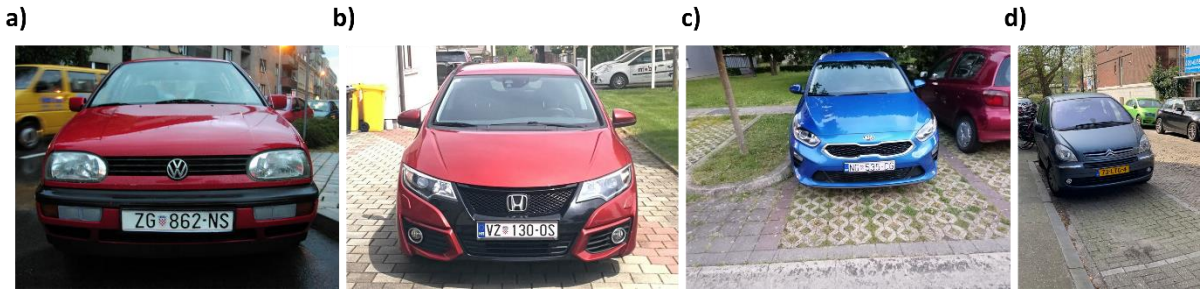
4.4. Priprema podataka i učenje modela za detekciju pločica

Izrada algoritma može se podijeliti na dva glavna dijela, a to su detekcija objekata, u ovom slučaja to su registarske pločice, te raspoznavanje teksta s registarskih pločica.

Za postupak detekcije registarskih pločica korišten je Tensorflow model za duboko učenje SSD [17] s osnovnom mrežom ResNet 50 [31] (**SSD ResNet 50, v1 640x640)**. Uz navedeni model isproban je i EfficientDet [19] model, ali je SSD ResNet 50 dao bolje rezultate. SSD ResNet model često je korišten za različite potrebe detekcije objekata, a odabran je zbog vrlo dobrog omjera brzine i srednje prosječne preciznosti (mAP) te zbog prikladnosti detekciji u realnom vremenu.

4.4.1. Priprema podataka

Prije postupka učenja modela potrebno je prikupiti i organizirati ulazne podatke za treniranje i testiranje. Za te potrebe korišteno je ukupno 1379 različitih fotografija vozila (primarno automobila) iz nekoliko izvora. Svaki pojedini skup je pregledan te su neodgovarajuće fotografije izbačene. Nakon biranja korišteno je 417 fotografija je s repozitorija Fakulteta računarstva i elektrotehnike Zagreb [24], 350 fotografija preuzeto je s Njuškalo oglasnika, 404 fotografije preuzeto je s Kaggle repozitorija [25], a još je korišteno 208 fotografija iz vlastite kolekcije. Primjeri slika iz svakog izvora prikazani su na Slici 5. Na većini fotografija vozila su slikana sprijeda ili straga, dok ostatak sadrži vozila iz različitih pozicija, kutova i udaljenosti. Skup podataka je raznolik pa tako sadrži i fotografije slikane u različita doba dana i noći, fotografije s različitih uređaja, različite kvalitete i sl.



Slika 5. Primjeri slika iz korištenih izvora: a) FER, b) Njuškalo, c) autorska, d) Kaggle.

Od ukupnih 1379 fotografija 908 (≈ 66%) registracija je s područja Republike Hrvatske, a 471 (≈ 34%) registracija je iz nekoliko različitih država. Detaljna podjela prema podskupovima prikazana je u Tablici 1.

Tablica 1. Podjela registracija na domaće i strane.

Izvor	**Ukupno**	**Broj HR registracija**	**Broj inozemnih registracija**
FER [34]	417	350	67
Njuškalo	350	350	-
Kaggle [33]	404	-	404
Autor	208	208	-
ukupno	**1397**	**908**	**471**

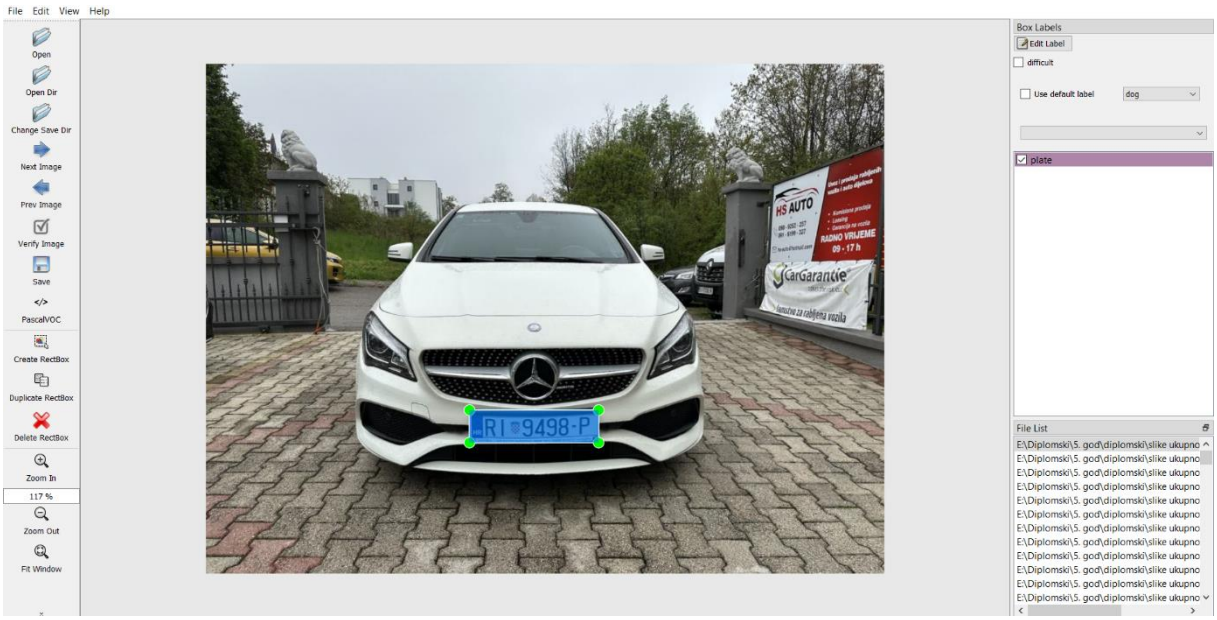
Svaki podskup fotografija podijeljen je otprilike u omjeru 80% (1099) za treniranje i 20% (280) za testiranje. Podjela je napravljena ručno, nasumičnim odabirom kako bi se osiguralo da su svi podskupovi zastupljeni u skupovima za treniranje i testiranje. Brojčana podjela po podskupovima vidljiva je u Tablici 2.

Tablica 2. Podjela registracija na skupove za testiranje i treniranje.

Izvor	**treniranje**	**testiranje**	**ukupno**
FER	333 (280 HR, 53 inozemne)	84 (70 RH, 14 inozemne)	**417**
Njuškalo	280	70	**350**
Kaggle	320	84	**404**
Autor	166	42	**208**
ukupno	**1099**	**280**	**1379**

4.4.2. Označavanje slika

Nakon izrade skupa ulaznih podataka za model sve slike je bilo potrebno anotirati (označiti). Postupak anotacije slika proveden je alatom LabelImg (Slika 6.). Anotiranje podrazumijeva što preciznije označavanje, pravokutnom oznakom, područja slike na kojemu se nalazi traženi objekt te pridruživanje klase objekta. Dakle za temu ovog rada bilo je potrebno označiti područje fotografije na kojemu se nalazi registracija vozila te mu pridružiti klasu „plate”.



Slika 6. Dodjeljivanje oznake slici u LabelImg.

Kada se fotografija označi tada se za nju automatski kreira XML datoteka (Slika 7.) koja sadrži podatke o nazivu i veličini slike, klasi kojoj pripada te koordinatama na kojoj se nalazi oznaka na njoj.

```
<annotation>
        <folder>280503</folder>
        <filename>P5280054.jpg</filename>
        <path>E:\Diplomski\5. god\baza_slika\280503\P5280054.jpg</path>
        <source>
                <database>Unknown</database>
        </source>
        <size>
                <width>640</width>
                <height>480</height>
                <depth>3</depth>
        </size>
        <segmented>0</segmented>
        <object>
                <name>plate</name>
                <pose>Unspecified</pose>
                <truncated>0</truncated>
                <difficult>0</difficult>
                <bndbox>
                        <xmin>217</xmin>
                        <ymin>338</ymin>
                        <xmax>439</xmax>
                        <ymax>373</ymax>
                </bndbox>
        </object>
</annotation>
```

Slika 7. Izgled XML datoteke kreirane nakon označavanja slike.

4.4.3. TFRecords i Labelmap datoteke

Sljedeća je izrađena Label_map datoteka. To je tekstualna datoteka koja sadrži popis klasa objekata i koristi se kod treniranja TensorFlow modela za klasifikaciju. U njoj su brojčano i tekstualno dodijeljeni identifikatori klasa što kasnije omogućuje modelu prikaz rezultata detekcije na način koji je razumljiv ljudima. S obzirom da se u ovoj temi radi samo o jednoj klasi objekta, sadržaj Label_map datoteke prikazan je na Slici 8.

```
item {
  id: 0
  name:'plate'
}
```

Slika 8. Izgled Label_map datoteke.

Nakon toga su kreirane TFRecords datoteke, pojedinačno za skupove za treniranje i testiranje modela. To su datoteke u koje se podaci pohranjuju u binarnom formatu što modelu omogućuje brzu obradu na velikim skupovima podataka. One sadrže podatke o slikama i njihovim pripadajućim klasama. Kreiraju se naredbom u terminalu u kojoj se pokreće Python datoteka generate_tfrecord.py s ulaznim parametrima koji se odnose na putanju do direktorija s ulaznim slikama i Label_map datotekom.

4.4.4. Konfiguriranje modela

Za detekciju objekata preuzet je pretrenirani SSD ResNet50 640x640 TensorFlow model s Zoo repozitorija pretreniranih na COCO skupu podataka. Da bi taj model ispravno radio za potrebe detekcije registarskih pločica na pripremljenom skupu podataka potrebno je prilagoditi njegovu konfiguracijsku datoteku. Datoteka je pronađena u preuzetoj mapi s modelom pod nazivom pipeline.config.

Unutar datoteke određen je broj klasa koje model treba detektirati, veličina ulazne slike, arhitektura modela, postavke neuronskih mreža, veličina serije, putanje do datoteka i dr.

Ovom prilikom postavljen je broj klasa na jedan, uklonjeno je nasumično rotiranje slika prilikom treniranja, određene putanje do TFRecords datoteka, Label_map datoteka, veličina serija postavljena je na 4 te je podešena stopa učenja.

4.4.5. Treniranje modela

Kada je sve pripremljeno pokrenut je trening modela. To je učinjeno naredbom:

python3 model_main_tf2.py --model_dir=models/ssd_resnet50 –pipeline_config_path=models/ssd_resnet50/pipeline.config.

Treniranje je dugotrajan proces te ovisno o postavkama modela, performansama računala, veličini skupa podataka i sl. može trajati od nekoliko sati do nekoliko dana. Za vrijeme trajanja treninga modela u terminalu se ispisuju podaci o gubitcima modela što je vidljivo na Slici 9. Ukoliko su rezultati zadovoljavajući postupak se može prekinuti ranije pritiskom na CTRL + C, u suprotnome treniranje završava kada se dosegne predodređeni broj koraka.

```
INFO:tensorflow:{'Loss/classification_loss': 0.119903825,
 'Loss/localization_loss': 0.0031133885,
 'Loss/regularization_loss': 0.03088433,
 'Loss/total_loss': 0.15390155,
 'learning_rate': 0.00069100846}
I0619 10:43:09.961764 140662577259136 model_lib_v2.py:708] {'Loss/classification_loss': 0.119903825,
 'Loss/localization_loss': 0.0031133885,
 'Loss/regularization_loss': 0.03088433,
 'Loss/total_loss': 0.15390155,
 'learning_rate': 0.00069100846}
```

Slika 9. Prikaz ispisa u terminalu tijekom treniranja modela.

Po završetku treniranja modela potrebno je izvesti tzv. graf zaključivanja kako bi se model mogao kasnije koristiti u sustavu. To je zapravo spremljeni, naučeni model koji se koristi za predviđanje rezultata na neviđenim podacima u prototipu ovog sustava za ulaz u garažu. Graf je izvezen sljedećom naredbom:

python3 exporter_main_v2.py --pipeline_config_path=models/ssd_resnet50/pipeline.config --trained_checkpoint_dir=models/ssd_resnet50 --output_directory=models/ssd_resnet50/exported_models.

Nakon treniranja, provedena je evaluacija performansi modela iz koje je dobivena srednja prosjećna preciznost (mAP) i odziv. Za EfficientDet model mAP je iznosila 35%, odziv 36.4%, dok je za SSD ResNet mAP iznosila 41%, a odziv 45.6%. Budući da je SSD ResNet pokazao bolje rezultate, dodatno je testiran na skupu od 26 slika koje nisu korištene u fazi treniranja. U potpunosti su ispravno detektirane 23 slike, dok su dvije djelomično točne, a na jednoj slici nije ispravna detekcija. Prema tome, model je u provjeri ipak pokazao zadovoljavajuće rezultate. Iz tog razloga SSD ResNet model je primijenjen u izradi prototipa.

U početnom dijelu funkcije za detekciju objekata (Slika 10.) dohvaćena slika s kamere se pretvara u višedimenzionalni niz. Vrijednosti tog niza predstavljaju intenzitet piksela na određenoj poziciji slike. Na takvom obliku slike pokreće se detekcija objekata na modelu pomoću **run_inference_for_single_image** funkcije te se pohranjuje u **output_dict**. **Output_dict** predstavlja rječnik u kojemu su pohranjeni rezultati detekcije objekata na slici. U njemu se nalaze informacije o koordinatama detektiranih objekata, ocjenama detekcije, indeksima klasa objekata te broju detekcija na slici. S obzirom na potrebe aplikacije korišteni su samo podaci o koordinatama detektiranog objekta i ocjeni detekcije koji su pohranjeni u varijable *boxes*, odnosno *scores*. Podaci o indeksu klase nisu potrebni jer se koristi samo jedna klasa „plate”, kao ni podaci o broju detektiranih objekata jer se uzima samo prva detekcija koja ujedno ima najveću ocjenu vjerojatnosti. Dodatno se prije daljnje obrade slika u varijable *width* i *height* pohranjuju širina i visina trenutno učitane slike. Prikaz detektirane registarske pločice na ulaznoj slici prikazan je na Slici 11.

```
def ocr_plates(img):
    image_np = np.array(img)
    # Pokretanje detekcije objekata na modelu
    output_dict = run_inference_for_single_image(model, image_np)
    # Dobivanje rezultata detekcije (ocjene i okviri)
    scores = output_dict['detection_scores']
    boxes = output_dict['detection_boxes']

    # Dobivanje dimenzija slike
    width = image_np.shape[1]
    height = image_np.shape[0]
    # Filtriranje ocjena detekcije i koordinata detekcija
    scores_filtered = [score for score in scores]
    boxes_filtered = boxes[:len(scores_filtered)]
```

Slika 10. Programski kod za pokretanje detekcije objekata i dohvaćanje rezultata.

Slika 11. Prikaz detektirane registarske pločice na ulaznoj slici.

4.5. Algoritam za očitavanje registarskih pločica

Za izradu algoritma za prepoznavanje registarskih oznaka vozila korišten je programski jezik Python u kombinaciji s nekoliko neizostavnih biblioteka kao što su TensorFlow, EasyOCR, OpenCV i NLTK.

Nakon detekcije registarskih pločica na slikama pomoću funkcije **run_inference_for_single_image**, slijedi pretprocesiranje slika prije primjene postupka raspoznavanja teksta s registarskih pločica.

4.5.1. Pretprocesiranje slika

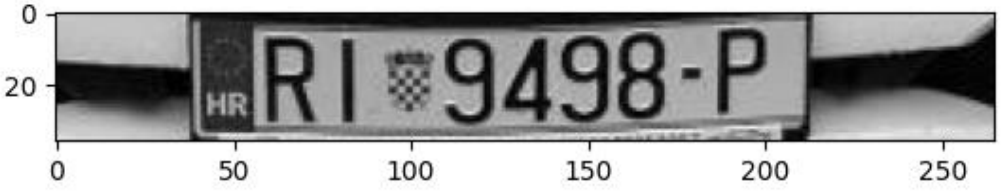
Kako bi prepoznavanje teksta bilo što približnije stvarnom tekstu fotografije je prethodno potrebno dodatno obraditi. Za potrebu toga izrađena je funkcija **ocr_plates** u kojoj se izvršava glavni dio funkcionalnosti cijelog algoritma. **Ocr_plates** kao parametar uzima ulaznu sliku s web kamere. Primjer ulazne slike prikazan je na Slici 12.

Slika 12. Ulazna slika prije primjene tehnika obrade slike.

Početni korak obrade slika odnosi se na izdvajanje regije slike (Slika 13.), na kojoj je detektirana tablica, proširene za fiksni iznos od 100 piksela. Taj se postupak radi kako bi se uklonila nepotrebna pozadina koja bi mogla utjecati na kvalitetu prepoznavanja teksta s pločice.

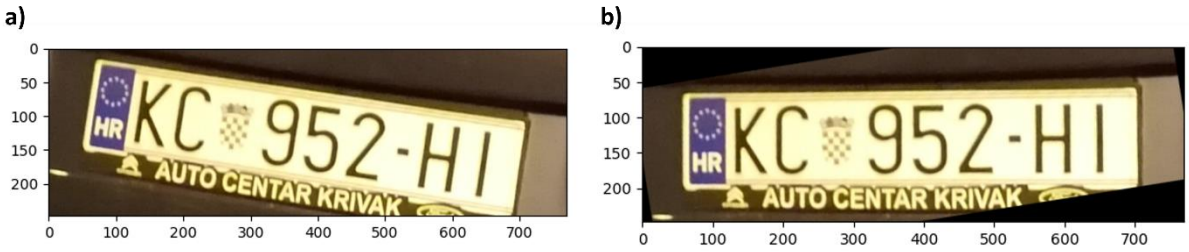
Slika 13. Izdvojena regija slike s detektiranom tablicom.

Kod postupka prepoznavanja teksta vrlo je bitna kontrastna pozadina, crna slova s bijelom pozadinom ili obrnuto, radi dobivanja boljih rezultata. Iz tog razloga izdvojenu regiju slike se pretvara u sive tonove (Slika 14.). Na taj se način lakše manipulira dobivenim tekstom jer je lakše izdvojiti tekst koji „dominira" slikom.



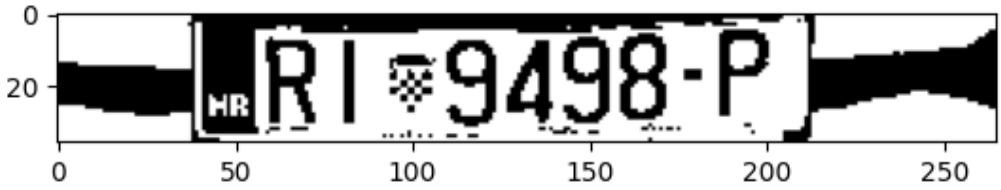
Slika 14. Konverzija izdvojene regije slike u sive tonove.

S obzirom da OCR daje bolje rezultate što je tekst horizontalnije na slici izrađena je funkcija **rotate_image** koja ispravlja nagib registarske pločice na slici. Prikaz izgleda slike nakon ispravljanja nagiba dan je na Slici 15.



Slika 15. Prikaz ispravljanja nagiba slike izdvojene regije.

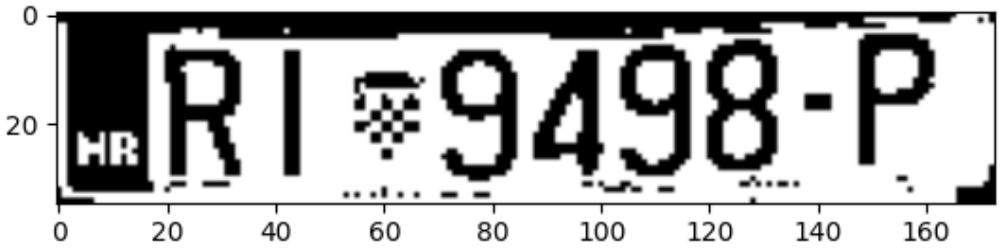
Nakon ispravljanja rotacije slike primijenjena je binarizacija (Slika 16.). Tom prilikom svaki se piksel označava kao crni ili bijeli prema zadanim pravilima. U primjeru ovog rada odlučeno je da se na slike primjenjuje kombinacija jednostavne binarizacije i Otsu binarizacije.



Slika 16. Primjena binarizacije na sliku izdvojene regije.

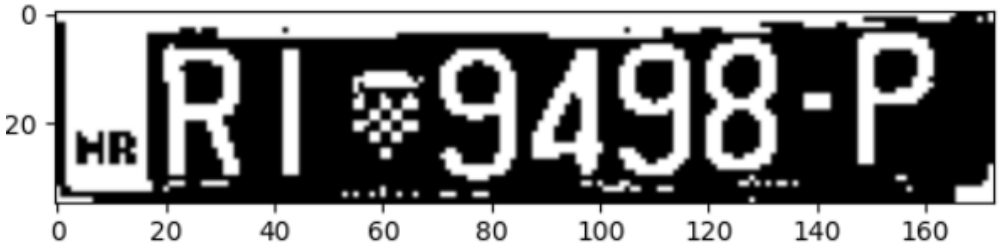
Kod kombinacije jednostavne binarizacije (globalnim pragom) i Otsu metode za ulaz se određuje donja i gornja granica intenziteta piksela. Otsu metoda automatski odabire prag iz donje i gornje granice, a zatim se po tim vrijednostima provodi jednostavna binarizacija.

Nakon binarnog ograničavanja pretražene su konture slike te se prema najvećoj pronađenoj konturi ona dodatno obrezuje oko dobivenih granica (Slika 17.).



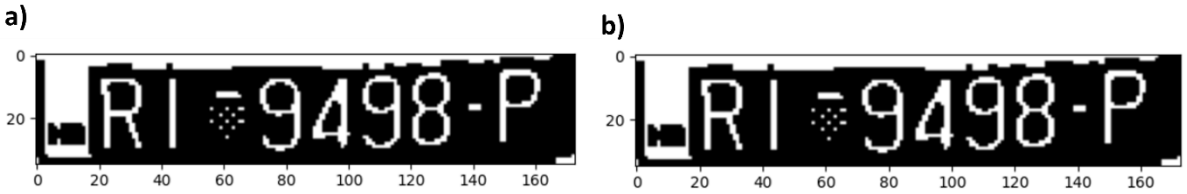
Slika 17. Obrezivanje binarizirane slike.

Zatim je dobivena slika invertirana (Slika 18.), tj. na slici su obrnute pozicije bijele i crne boje, te je tako pozadina postala crna, a slova bijela.



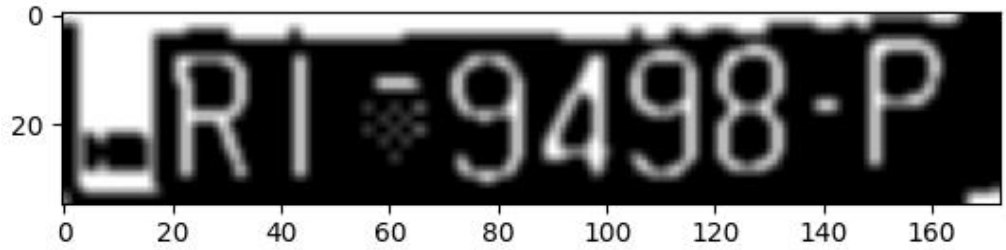
Slika 18. Invertiranje obrezane binarizirane slike.

Invertirana slika potom je *erodirana*, time su proširene rubovi teksta kako bi postali jasnije uočljivi te su joj potom odmah stanjeni rubovi primjenom dilatacije. Za eroziju je korištena struktura 2x2 matrice, a za dilataciju struktura 1x1 matrice.



Slika 19. Slika nakon primjene a) erozije i zatim b) dilatacije.

Iako slike prikazane na slici 19. izgledaju jednako, algoritam bolje prepoznaje tekst nakon primjene erozije te dilatacije na sliku. Obrada slike završena je smanjenjem šuma Gaussovim zamućenjem čime su slici uklonjeni manji detalji i povećana je zaglađenost (Slika 20.). Na Slici 21. prikazan je izgled programskog koda za obradu slika u Pythonu.



Slika 20. Primjena Gaussovog zamućenja na dilatiranu sliku.

```
# Izdvajanje regije slike koja sadrži detektiranu tablicu
region = image_np[int(roi[0]):int(roi[2]), int(roi[1]):int(roi[3])]
# Ispravljanje nagiba slike
convert = rotate_image(region)
# Konverzija u sive tonove
gray = cv2.cvtColor(convert, cv2.COLOR_BGR2GRAY)
# Binarizacija
_,binary = cv2.threshold(gray, 50, 150, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
# Detekcija najveće konture
contours, _ = cv2.findContours(binary, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
largest = max(contours, key=cv2.contourArea)
x, y, w, h = cv2.boundingRect(largest)
# Obrezivanje prema najvećoj konturi
cropped = binary[y:y+h, x:x+w]
# Invertiranje slike
inverted = cv2.bitwise_not(cropped)
# Primjena erozije i dilatacije na invertiranu sliku
erozija = cv2.erode(inverted, np.ones((2,2), np.uint8), iterations=1)
dilatacija = cv2.dilate(erozija, np.ones((1,1), np.uint8), iterations=1)
# Primjena Gaussova filtera šuma
filtered = cv2.GaussianBlur(dilatacija, (3,3), 1)


ocr_result = ocr(filtered)
```

Slika 21. Prikaz postupaka obrade slika u Pythonu.

4.5.2. Funkcija *rotate_image*

Rotate_image radi na principu Houghove transformacije linija. Na slici se detektiraju rubovi kako bi se dobile linije objekata. Prilagođavaju se parametri poput minimalne duljine linije i maksimalnog razmaka između dva dijela linije da bi se pronašle linije željene dužine. Zatim petlja iz dobivenih linija pronalazi najdulju te joj određuje kut nagiba u odnosu na horizontalu. Prema dobivenom kutu se izračunava iznos kuta za rotaciju slike. Primjenom tog kuta na matricu rotacije izravnava se položaj registracije po horizontali.

4.5.3. Prepoznavanje registarskih oznaka

Ključna faza cijelog algoritma je prepoznavanje registarskih oznaka vozila. Njen prvi korak čini prepoznavanje teksta s pretprocesirane slike. Za tu svrhu korištena je biblioteka EasyOCR. U kontekstu sustava za ulaz u garažu nije nužno u potpunosti točno prepoznati registarsku oznaku već je dovoljno osigurati podudaranje u određenom broju znakova između registracije na ulazu i neke od registracija s dozvolom ulaska iz popisa. U ovom slučaju određeno je maksimalno dozvoljeno odstupanje od dva znaka razlike između

detektirane i registracije s popisa. Da bi prepoznati tekst bio što približniji, sadržaj mu je ograničen na slova od A do Z i brojeva od 0 do 9. Tekst ne može sadržavati nikakve specijalne znakove kao ni razmake i ograničen je isključivo na velika slova.

```
def ocr(converted_image):
    chars = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    reader = easyocr.Reader(['en'])
    ocr_result = reader.readtext(converted_image, detail=0, paragraph=True, text_threshold=0.9, allowlist=chars)

    return ocr_result
```

Slika 22. Postavke EasyOCR alata.

Prepoznavanje registarske oznake započinje tako da se slika filtrirana Gaussovim zamućenjem prosljeđuje **readText** funkciji iz EasyOCR biblioteke (Slika 22.). Tekst dobiven upotrebom EasyOCR-a se zatim prosljeđuje funkciji **classiffy_plate2** koja je prikazana na Slici 23. Tom funkcijom provjerava se podudaranje između prepoznatog teksta i popisa dozvoljenih registarskih oznaka na temelju Levenshteinove udaljenosti. Levenshteinova udaljenost provjerava sličnost između dva niza znakova, a određuje ju minimalan broj operacija izmjene znakova da bi jedan niz bio jednak drugom. Funkcija računa udaljenost između prepoznatog teksta i svake registracije iz popisa, uzimajući u obzir iznos apsolutne razlike u duljini između prepoznatog teksta i popisa registracija. Ukoliko je podudaranje pronađeno ono se prihvaća samo u slučaju kada je duljina detektiranog teksta pet ili više znakova, a razlika između tih tekstova nije veća od dva znaka.

```
def classify_plate2(detected_text, allowed, entrances, max_allowed_distance):
    with open(allowed, mode='r', newline='', encoding='utf-8') as file:
        reader = csv.reader(file)
        allowed_plate = [row[0] for row in reader]
        distances = [distance.edit_distance(detected_text, correct_text) - abs(
            len(detected_text) - len(correct_text)) for correct_text in allowed_plate]
        am = np.argmin(distances)
        m = distances[am]
        if detected_text == '':
            return None
        if m <= max_allowed_distance and len(detected_text) >= 5:
            print(allowed_plate[am])
            with open(entrances, mode='a', newline='', encoding='utf-8') as file:
                time_hm = time.strftime("%H:%M:%S", time.localtime())
                writer = csv.writer(file)
                writer.writerow([allowed_plate[am], date, time_hm])
            return allowed_plate[np.argmin(distances)]
        else:
            return None
```

Slika 23. Funkcija za provjeru podudaranja registracije s popisom.

Ukoliko za filtriranu sliku nema podudaranja tada se invertirana slika šalje u funkciju *classify_plate2*, a ako ni tada nema podudaranja tekst se provjerava za izrezanu regiju detektirane tablice. U slučaju da ni nakon sve tri provjere nema podudaranja ne dopušta se ulazak tom vozilu, ako je podudaranje pronađeno ono se bilježi u bazi podataka te se ulaz dopušta. Dio Python programskog koda za provjeru prava ulaska u garažu prikazan je na Slici 24.

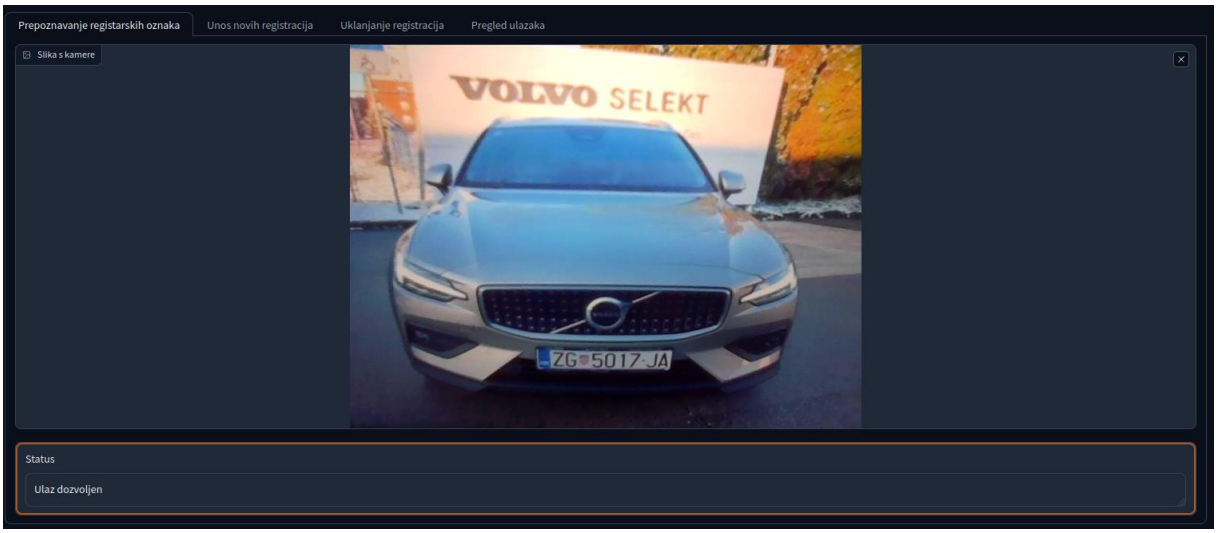
```
if ocr_result_region:
    text = ocr_result_region[0]
    cleaned_txt = re.sub(r'[^a-zA-Z0-9]', '', text).upper()
    print("Detektirani broj tablice 4: " + str(cleaned_txt))
    classified_plate2 = classify_plate2(cleaned_txt, allowed, entrances, max_allowed_distance=2)

    if classified_plate2 is not None:
        print("Tablica pronađena")
        result = "Ulaz dozvoljen"

    else:
        print("Nema podudaranja")
        result = "Ulaz odbijen"
```

Slika 24. Programski kod za provjeru prava ulaska u garažu.

4.6. Prototip aplikacije

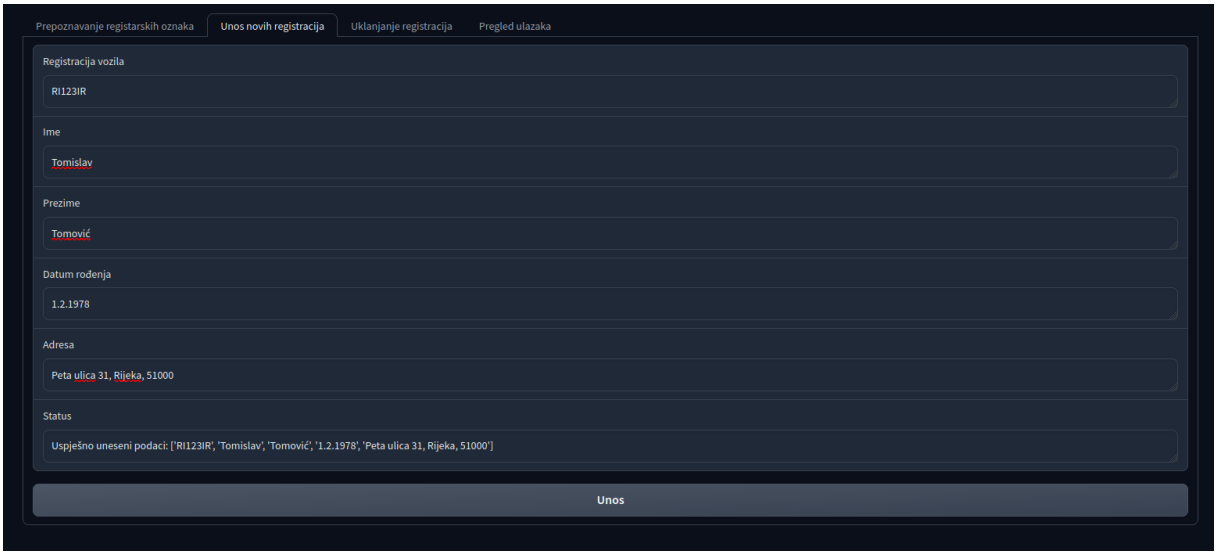
Upotrebom Python biblioteke Gradio izrađen je prototip aplikacije sustava za ulaz u garažu pomoću računalnog vida. Aplikacija je zamišljena kao sučelje za nadzor i upravljanje sustavom ulaska u garažu. Sučelje je podijeljeno na četiri kartice (*engl. Tab*) od kojih svaka sadrži funkciju za različitu zadaću.

Početna kartica (Slika 25.) sadrži prikaz slike s web kamere kao imitacije kamere na ulasku u garažu i ispisa statusa provjere registarske oznake s popisom registracija s dozvolom ulaska. Postupak se odvija neprekidno i u realnom vremenu. Kada se na kameri detektira i prepozna registarska oznaka s dozvolom ulaska ona se upisuje u tablicu ulazaka skupa s datumom i vremenom ulaska uz statusnu poruku „dozvoljeno” ili „odbijeno”.



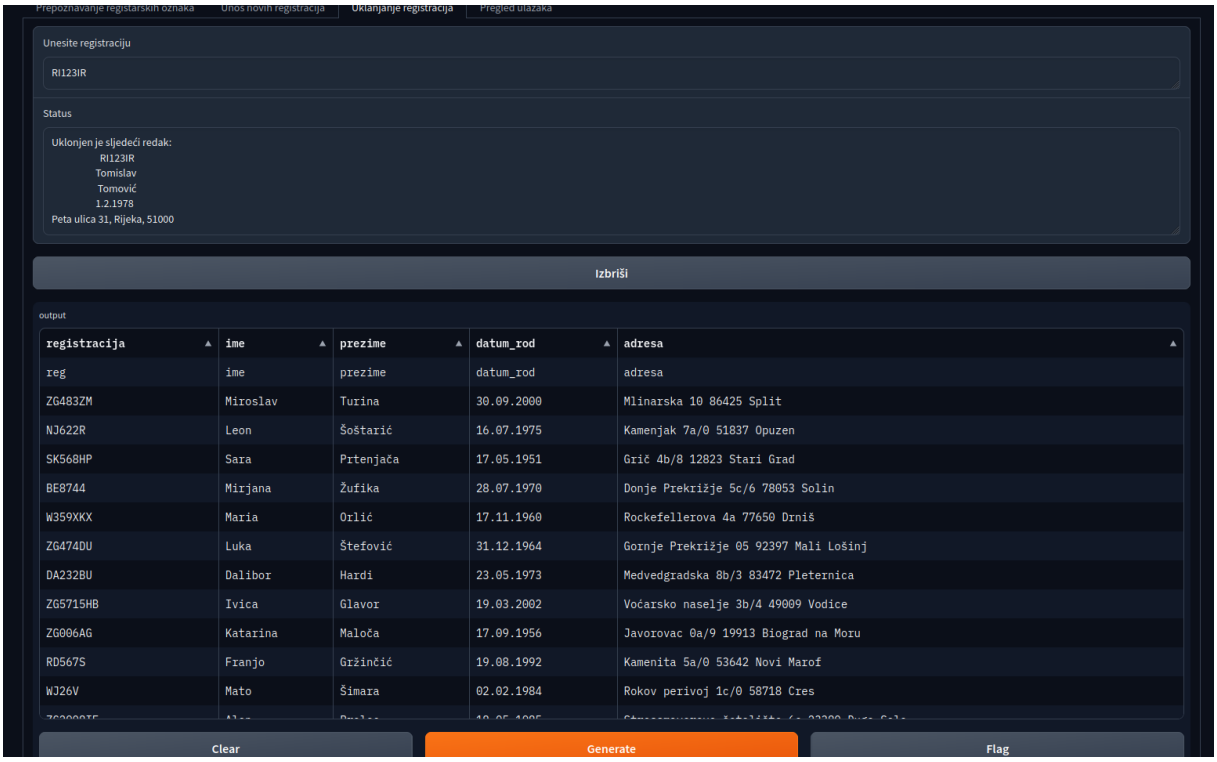
Slika 25. Prikaz početne kartice prototipa aplikacije sustava za ulaz u garažu.

Druga kartica (Slika 26.) ima funkciju unosa novih registarskih oznaka na popis dozvoljenih. Podaci koji su potrebni su registarska oznaka vozila, ime i prezime, datum rođenja i adresa. Nakon potvrde unosa ispisuje se povratna poruka o uspješnom unosu i unesenim podacima.



Slika 26. Prikaz druge kartice prototipa aplikacije sustava za ulaz u garažu.

Treća kartica (Slika 27.) omogućuje uklanjanje registarske oznake iz popisa dozvoljenih. Na njoj se nalazi prikaz cijelog popisa dozvoljenih registarskih oznaka te prostor za unos registracije koja se želi ukloniti. Kada se to izvrši ispisuje se poruka o uspješnom brisanju s prikazom podataka koji su izbrisani. Posljednja kartica prikazuje sve ulaske u garažu toga dana kroz tablicu (Slika 28).



Slika 27. Prikaz treće kartice prototipa aplikacije sustava za ulaz u garažu.

Slika 28. Prikaz posljednje kartice prototipa aplikacije sustava za ulaz u garažu.

4.6.1. Testiranje i evaluacija

Za potrebe testiranja rada algoritma iz početnog skupa slika izbačene su sve one koje se ne mogu očekivati kod stvarne primjene na ulazu u garažu. Prema tome testiranje je provedeno na 989 slika koje prikazuju samo prednju ili stražnju stranu različitih vozila s različitih udaljenosti.

Izračunate su točno pozitivne (TP), točno negativne (TN), lažno pozitivne (FP) i lažno negative (FN) vrijednosti te preciznost i točnost.

Za provjeru podudaranja izrađene su dvije funkcije kako bi se utvrdilo za koju se dobivaju bolji rezultati u konačnom rješenju. Prva funkcija, nazvana classify_plate1 koristi Levenshteinovu metodu za provjeru sličnosti dvaju nizova, a druga funkcija, nazvana

classify_plate2, za provjeru koristi Levenshteinovu metodu umanjenu za razliku duljina dvaju nizova. Obje funkcije ne uzimaju u obzir nizove kraće od pet znakova jer u Europi ne postoje registarske oznake kraće od pet znakova, a time se povećava i mogućnost pogrešnog prepoznavanja registracije. Također, obje funkcije dozvoljavaju određen broj znakova koje detektirani tekst može odstupati. Za obje funkcije isproban je različit broj dozvoljenih znakova odstupanja te su konačne dodijeljene vrijednosti pokazale najbolje rezultate. Prva funkcija dozvoljava određen broj znakova koje detektirani tekst smije odstupati ovisno o njegovoj duljini. Ukoliko mu je dužina pet znakova tada odstupanje smije biti maksimalno jedan znak. Povećavanjem dužine detektiranog teksta, za svaki dodatni znak, dozvoljeno je odstupanje za jedan znak više do maksimalno četiri znaka razlike između nizova. Kod druge funkcije situacija je jednostavnija zbog njenog principa rada, ukoliko je detektirani tekst dulji od četiri znaka dozvoljeno je maksimalno odstupanje od dva znaka.

Izrađen je popis stvarnih vrijednosti za sve slike, naknadno su mu vrijednosti sortirane, pohranjene u novi popis koji je podijeljen na pola i u njemu su zadržane samo jedinstvene vrijednosti. U ovom slučaju to je bilo potrebno jer skup ulaznih slika sadrži nekoliko registarskih oznaka koje se ponavljaju na različitim slikama. Dobiveni popis od 494 slike, korišten je kao popis registarskih oznaka s dopuštenjem ulaska u garažu.

Testiranje je provedeno tako da se za svaku sliku nakon obrade detektirao tekst koji se funkcijom za pronalazak podudaranja uspoređivao s referentom vrijednosti (*engl. ground truth*) i s popisom dozvoljenih registarskih oznaka.

Ukoliko detektirani tekst odgovara stvarnoj vrijednosti za trenutnu sliku i ako je ona na popisu dozvoljenih registracija tada se on bilježi kao TP, a ako stvarni tekst nije na popisu dozvoljenih označen je kao TN. Kada su detektirani i stvarni tekst različiti, ali se oba teksta nalaze na popisu dozvoljenih tada se detektirani tekst označava kao TP. U slučaju da su detektirani i stvarni tekst različiti i uz to se ne nalaze na popisu dozvoljenih tada je to TN. Za situaciju kada su detektirani i stvarni tekst različiti, a detektirani tekst je na popisu dok stvarni tekst nije, detektirani tekst se označava kao FP. U obrnutoj situaciji u kojoj su tekstovi različiti, ali je stvarni tekst na popisu, a detektirani nije, detektirani tekst označen je kao FN. Iz navedenih vrijednosti izračunate su preciznost i točnost.

5. Rezultati

Nakon provedenog testiranja sustava dobiveni su različiti rezultati. Za najznačajnije metode obrade slike u Tablici 3. prikazan je broj slika za koje je pronađeno podudaranje s popisom registarskih oznaka. Za izdvojenu regiju slike s detektiranom tablicom na koju nisu primijenjene metode obrade promatrajući obje funkcije ukupno je 802 podudaranja s registracijama iz popisa. Iz Tablice 3. je vidljivo da se primjenjivanjem više tehnika obrade na sliku povećava broj pronađenih podudaranja. Dodatnom primjenom pojedinih metoda, npr. binarizacija nakon konverzije slike u sive tonove, broj podudaranja se smanjuje. Međutim, one kasnije utječu na povećanje broja poduranja dodavanjem dodatnih metoda obrade, npr. inverzija slike nakon binarizacije i konverzije u sive tonove. Kako bi se postigli što bolji rezultati proveden je dodatni postupak provjere podudaranja. Ako za filtriranu sliku nije pronađeno podudaranje tada se provjerava podudaranje za invertiranu, a potom za izdvojenu regiju detektirane tablice. Takvom kombinacijom provjere postignuti su najbolji rezultati od 931 podudaranja.

Tablica 3. Prikaz broja podudaranja nakon različitih obrada slike

Metoda obrade	**classify_plate1 (cp1)**	**Točnost cp1 (%)**	**classify_plate2 (cp2)**	**Točnost cp2 (%)**
Izdvojena regija	688	81.8	702	87.97
Sivi tonovi	720	83.72	781	89.28
Binarizacija	667	83.62	729	88.67
Inverzija	742	87.05	798	91.1
Filtriranje šuma	727	87.76	757	89.38
Dodatna provjera	**852**	**93.43**	**908**	**96.36**

Dakle, promatrajući skupno obje funkcije za prepoznavanje registarskih oznaka od ukupno 989 registracija ukupno je prepoznata 931 registracija, za 48 nije pronađeno podudaranje, a za 10 ih je pronađena pogrešna registracija. Od 48 registracija koje nemaju podudaranja s popisom registracija, 13 ih nema ispravno detektiran (objekt) položaj registarske pločice, a 35 ih nije prepoznato zbog različitih faktora, ali su im ispravno detektirani položaji pločica. Neprepoznavanje kao i pogrešno prepoznavanje registarskih oznaka dogodilo se kod slika lošije kvalitete, prevelike udaljenosti, velikog kuta fotografiranja i dr., što je utjecalo na to da tekst bude samo djelomično ili potpuno neočitan. Osim toga pogrešno prepoznavanje se dogodilo zbog kombinacije vrlo velike sličnosti između više registracija i prethodno navedenih razloga, dok je samo u jednom slučaju pogrešna detekcija teksta iz okoline dovela do pronađenog podudaranja s popisom dozvoljenih registracija. Pojedinačnom analizom funkcija za prepoznavanje registarskih

oznaka iz Tablice 4 je vidljivo da su nakon dodatne provjere podudaranja funkcijom classify_plates1 točno pronađene 852 (≈86%) ispravne registarske oznake, a za 137 (≈14%) nije pronađeno podudaranje. Funkcijom classify_plate2 ispravno je prepoznato 908 (≈91.8%) slika, a za 81 (≈8.2%) ih nije pronađeno podudaranje.

Tablica 4. Statistika funkcija za pronalaženje podudaranja između očitanog teksta i popisa dopuštenih registracija.

Broj slika	classify_plate1	classify_plate2
za koje je očitan bilo kakav tekst	856	928
s točno očitanim tekstom	852	908
bez prepoznatog teksta	137	81
s pogrešno prepoznatim tekstom	4	20

Dakle, kod classify_plate1 prepoznat je tekst s 4 slike za koje nije pronađeno podudaranje, dok ih je za classify_plate2 20 bez pronađenog podudaranja. Taj očitani tekst može biti posljedica ili loše kvalitete slika s kojih zbog raznih uvjeta (kvaliteta, udaljenost, osvjetljenje) nije dovoljno dobro očitan tekst ili je detektiran objekt iz okoline (reklama, prometni znak i sl.), a ne registarska pločica.

U Tablicama 5 i 6 prikazane su matrice konfuzije za funkcije classify_plate1 i classify_plate2. Iz Tablice 5 vidljivo je da je funkcija classify_plate1 ispravno klasificirala 924 registarske oznake dok je pogrešno klasificirala njih 65. S obzirom na FP i FN vrijednosti jasno je da je sustav pogrešno dopustio ulazak jednoj registraciji, a neopravdano je odbio njih 65.

Tablica 5. Matrica konfuzije funkcije classify_plate1.

classify_plate1	Pozitivno	Negativno
Pozitivno	TP = 430	FP = 1
Negativno	FN = 64	TN = 494

Prema dobivenim podacima dobiveno je da je preciznost funkcije classify_plate1 iznosila 99.77%, a njena točnost 93.43%.

Iz Tablice 6 vidljivo je da je funkcija classify_plate2 ispravno klasificirala 953 registarske oznake, a 36 pogrešno. U slučaju ove funkcije prema FP i FN vrijednostima 9

registracija je pogrešno označeno kao pozitivno, a 27 kao negativno. Preciznost funkcije classify_plate2 iznosila je 98.11%, a točnost 96.36%.

Tablica 6. Matrica konfuzije funkcije classify_plate2.

classify_plate2	Pozitivno	Negativno
Pozitivno	TP = 467	FP = 9
Negativno	FN = 27	TN = 486

Prema prikazanim rezultatima može se zaključiti kako je funkcija classify_plate2 dala pouzdanije rezultate, odnosno veći broj ispravno klasificiranih slučajeva. Iz tog razloga odlučeno je da se za primjenu u prototipu sustava koristi funkcija classify_plate2.

6. Diskusija

Tijekom izrade ovog sustava uočeni su različiti problemi i ograničenja koja utječu na dobivene rezultate. Jedan od značajnijih problema je položaj državnog grba na pojedinim registarskim pločicama među kojima su i hrvatske. Zbog njegovog položaja među tekstualnim znakovima OCR često grb zamjenjuje s nekim slovom koje odgovara njegovim obliku, kao što su npr. „U“ ili „V“. Također OCR često identificira grb jednako kao prvi nadolazeći znak, npr. u „RI1234AB“ na mjestu grba bi bila još jedna jedinica „RI11234AB“, što bi uz pogrešno očitanje još nekog znaka dalo teže prepoznatljiv tekst, osobito u području raspoznavanja registarskih oznaka gdje nema puno prostora za pogrešku. Još jedan problem odnosi se na često pogrešno prepoznavanje sličnih znakova, od kojih su najčešći prikazani u Tablici 6. Problem je posebno izražen na hrvatskim registracijama gdje se za nulu i slovo O koristi jednaki simbol, a slovo I i broj 1 su vrlo slični. Uz to dodatan problem često stvara i tekst na okviru registarske pločice, ali njega se lakše može izbaciti.

Tablica 7. Popis najčešće zamijenjenih znakova.

O	0
2	Z
7	Z
2	7
6	G
I	1
V	W
W	M

Vidljivo je kako u rezultatima ima odstupanja, pogrešno identificiranih registarskih oznaka, ne prepoznatih registracija itd., što je posljedica korištenja slika iz različitih izvora, najizraženije na slikama s Njuškala. Slike su fotografirane različitim uređajima, s različitih udaljenosti, pozicija, kutova, različite kvalitete, drugačijih vremenskih uvjeta i sl. Iz jednog gledišta to može biti povoljno jer se unatoč tome pokazuje sposobnost sustava da i dalje s visokom točnošću prepozna takve registracije.

Iz toga razloga korištene su razne tehnike obrade slika, od kojih neke uglavnom pomažu samo kod teže očitljivih slika, slika lošije kvalitete. Kako bi rezultati bili dodatno poboljšani te kako bi sustav bio pogodniji za stvarnu primjenu, potrebno je koristiti ujednačen skup slika koji bi bio fotografiran s istom kamerom, s iste udaljenosti i pozicije, na mjestu primjene sustava. Također, za primjenu je potrebno istrenirati model na takvim slikama kao i prilagoditi parametre korištenih tehnika obrade slika, kako bi najbolje odgovarale takvom skupu slika. Uz to, sustav bi bolje radio kada bi detekcija registarskih pločica bila bolja za što bi bilo potrebno koristiti još veći skup slika za učenje modela. Pretpostavka je da bi se kod stvarne primjene, u slučaju korištenja kamere s infracrvenim (IC) senzorom lakše postigli visoki rezultati, prvenstveno u različitim vremenskim uvjetima. Takvi senzori emitiraju infracrvenu svjetlost koja se zbog reflektirajuće pozadine odbija od registarske pločice natrag prema kameri. Na taj se način poboljšava kontrast između tamnog teksta i svjetle pozadine što rezultira jasnijom i oštrijom slikom, uz smanjenje sjena i odsjaja. Time bi se smanjila potreba za korištenjem nekih tehnika obrade slika i olakšalo očitavanje registarskih oznaka u svim vremenskim uvjetima [34].

7. Zaključak

U ovom radu prikazana je izrada prototipa sustava za ulaz u garažu korištenjem računalnog vida. Izrađen je algoritam za detekciju registarskih pločica i očitavanje registarskih oznaka. Detekcija registarskih pločica provedena je upotrebom **SSD ResNet 50, v1 640x640** modela koji je treniran na skupu od 1379 slika prikupljenih iz četiri različita izvora. Prije postupka prepoznavanja teksta slike su obrađene postupcima konverzije slike u sive tonove, binarizacije, inverzije, erozije i dilatacije te Gaussovim filterom šuma. Očitavanje teksta s registarske pločice izvršeno je s bibliotekom **EasyOCR**, nakon čega je za detektirani tekst provjeravano podudaranje s popisom dozvoljenih registarskih oznaka.

Za pronalaženje podudaranja između detektiranog teksta i popisa dozvoljenih registracija izrađene su dvije funkcije te je nakon testiranja funkcija s različitim parametrima za primjenu u prototipu sustava odabrana funkcija **classify_plate2** koja je dala bolje rezultate. Prototip sustava izrađen je s bibliotekom Gradio koja omogućuje brzu i jednostavnu izradu te pokretanje prototipa u web pregledniku. Algoritam je testiran na 989 slika izdvojenih iz početnog skupa te je postignuta preciznost od 98.11% i točnost od 96.36%.

Izradu ovakvih sustava otežavaju pojedine karakteristike samih registarskih oznaka. Kroz izradu ovog rada uočeno je da OCR državne grbove često pogrešno identificira kao slovo što smanjuje mogućnost ispravnog prepoznavanja registracije. Uz to, pojedini znakovi se zbog međusobne sličnosti pogrešno prepoznaju kao drugi znakovi. Za izradu ovog rada korišteni su različiti izvori slika što je omogućilo bolju prilagodljivost modela na različite uvjete prilikom detekcije, a istovremeno otežalo prepoznavanje njihovog teksta. Unatoč tome izrađen je prototip sustava koji daje visoke rezultate, ali za stvarnu primjenu moguće je unaprjeđenje sustava prilagodbom funkcijskih parametara, učenjem modela na većem broju slika snimljenih u stvarnim uvjetima garaže i moguće upotrebom kamere s IC senzorom.

8. Reference

[1] R. C. Gonzalez and R. E. Woods, *Digital image processing*. New York, NY: Pearson, 2018.

[2] R. Szeliski, "Computer Vision: Algorithms and Applications, 2nd Edition".

[3] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 5, pp. 550–554, May 1994, doi: 10.1109/34.291440.

[4] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[5] N. Ganatra, "A Comprehensive Study of Applying Object Detection Methods for Medical Image Analysis," 2021.

[6] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K. Ghosh, "Optical Character Recognition Systems," in *Optical Character Recognition Systems for Different Languages with Soft Computing*, A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K Ghosh, Eds., in Studies in Fuzziness and Soft Computing. , Cham: Springer International Publishing, 2017, pp. 9–41. doi: 10.1007/978-3-319-50252-6_2.

[7] N. Islam, Z. Islam, and N. Noor, "A Survey on Optical Character Recognition System," *Journal of Information*, vol. 10, no. 2, 2016.

[8] C.-W. Kang, R.-H. Park, and K.-H. Lee, "Extraction of straight line segments using rotation transformation: generalized hough transformation," *Pattern Recognition*, vol. 24, no. 7, pp. 633–641, Jan. 1991, doi: 10.1016/0031-3203(91)90030-9.

[9] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, Feb. 2013, doi: 10.1109/TCSVT.2012.2203741.

[10] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," *IEEE Access*, vol. 9, pp. 11203–11225, 2021, doi: 10.1109/ACCESS.2020.3047929.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." arXiv, May 09, 2016. Accessed: Dec. 09, 2023. [Online]. Available: http://arxiv.org/abs/1506.02640

[12] K. T. Islam *et al.*, “A Vision-Based Machine Learning Method for Barrier Access Control Using Vehicle License Plate Authentication,” *Sensors*, vol. 20, no. 12, Art. no. 12, Jan. 2020, doi: 10.3390/s20123578.

[13] A. U. Ahmed, T. M. Masum, and M. M. Rahman, “Design of an Automated Secure Garage System Using License Plate Recognition Technique,” *IJISA*, vol. 6, no. 2, pp. 22–28, Jan. 2014, doi: 10.5815/ijisa.2014.02.03.

[14] M. Al-Mheiri, O. Kais, and T. Bonny, “Car Plate Recognition Using Machine Learning,” in *2022 Advances in Science and Engineering Technology International Conferences (ASET)*, Feb. 2022, pp. 1–6. doi: 10.1109/ASET53988.2022.9734830.

[15] “EUR-Lex - 31998R2411 - EUR-Lex.” Accessed: Nov. 15, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/HR-EN/TXT/?uri=CELEX%3A31998R2411

[16] “Pravilnik o registraciji i označavanju vozila.” Accessed: Nov. 15, 2023. [Online]. Available: https://narodne-novine.nn.hr/clanci/sluzbeni/2017_12_130_2993.html

[17] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[18] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” arXiv, Jan. 06, 2016. Accessed: Dec. 09, 2023. [Online]. Available: http://arxiv.org/abs/1506.01497

[19] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and Efficient Object Detection,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 10778–10787. doi: 10.1109/CVPR42600.2020.01079.

[20] “Jaided AI: EasyOCR documentation.” Accessed: Nov. 15, 2023. [Online]. Available: https://www.jaided.ai/easyocr/documentation/

[21] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[22] “NLTK :: Natural Language Toolkit.” Accessed: Dec. 09, 2023. [Online]. Available: https://www.nltk.org/

[23] “NLTK :: nltk.metrics.distance module.” Accessed: Dec. 09, 2023. [Online]. Available: https://www.nltk.org/api/nltk.metrics.distance.html#nltk.metrics.distance.edit_distance

[24] “Matplotlib — Visualization with Python.” Accessed: Dec. 09, 2023. [Online]. Available: https://matplotlib.org/

[25] “HumanSignal/labelImg.” HumanSignal, Dec. 08, 2023. Accessed: Dec. 09, 2023. [Online]. Available: https://github.com/HumanSignal/labelImg

[26] Bryan S. Morse, “Lecture 4: Thresholding,” *Brigham Young University*, 2000, [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf

[27] T. D. Piyadasa, “Image Binarization in a Nutshell,” Medium. Accessed: Nov. 15, 2023. [Online]. Available: https://medium.com/@tharindad7/image-binarization-in-a-nutshell-b40b63c0228e

[28] Robert B Fisher, S. Perkins, A. Walker, and E. Wolfart, *Hypermedia Image Processing Reference*. 1996. Accessed: Nov. 15, 2023. [Online]. Available: https://www.dsi.unive.it/~atorsell/Hipr.pdf

[29] “Image Inversion - MATLAB & Simulink Example.” Accessed: Nov. 15, 2023. [Online]. Available: https://www.mathworks.com/help/supportpkg/beaglebone/ref/image-inversion.html

[30] “Gaussian Blur - an overview | ScienceDirect Topics.” Accessed: Nov. 15, 2023. [Online]. Available: https://www.sciencedirect.com/topics/engineering/gaussian-blur

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[32] “Projekt ‘License Plates.’” Accessed: Nov. 21, 2023. [Online]. Available: https://www.zemris.fer.hr/projects/LicensePlates/hrvatski/rezultati.shtml

[33] “License plates on vehicles.” Accessed: Nov. 21, 2023. [Online]. Available: https://www.kaggle.com/datasets/pcmill/license-plates-on-vehicles

[34] “Sensors | Free Full-Text | Automated License Plate Recognition for Resource-Constrained Environments.” Accessed: Dec. 09, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/22/4/1434?fbclid=IwAR13O-4WeMSKdXVbiRhszq_MQ8AkjhBgeqhhekGY5A6wEPPYLAe-dneiZaE

9. Popis priloga

S radom je priložena arhiva koja sadrži:

- main.py – programski kod za pokretanje prototipa
- test.py – programski kod za testiranje algoritma
- label_map.pbtxt – datoteka s popisom klasa
- graf zaključivanja/model

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija
www.inf.uniri.hr

Rijeka, 1.6.2023.

Zadatak za diplomski rad

Pristupnik: Karlo Calvi

Naziv diplomskog rada: Prototip sustava za ulaz u garažu korištenjem računalnog vida

Naziv diplomskog rada na eng. jeziku: Prototype of a garage entrance system using computer vision

Sadržaj zadatka: Implementirati prototip softverskog rješenja za automatsko otvaranje ulaza u garažu na osnovu prepoznavanja registarskih oznaka automobila pomoću računalnog vida. Sustav treba omogućiti unos dozvoljenih registarskih oznaka, te na osnovu slike automobila ispred garaže odlučiti treba li otvoriti garažu ili ne. Opisati prototip sustava i njegove pojedinačne komponente kao što je detekcija i raspoznavanje registarskih oznaka. Testirati prototip sustava na skupu slika automobila slikanih iz kuta koji je očekivan za slike na nastalih ulazu u garažu.

Mentor: doc. dr. sc. Miran Pobar

Voditeljica za diplomske radove:

izv. prof. dr. sc. Ana Meštrović

Komentor:

Zadatak preuzet: 01.06.2023.

Karlo Calvi

(potpis pristupnika)

Adresa: Radmile Matejčić 2
51000 Rijeka, Hrvatska
Tel: **+385(0)51 584 700**
E-mail: **ured@inf.uniri.hr**
OIB: **64218323816**
IBAN: **HR1524020061400006966**