

# Detekcija automobilskih tablica open cv bibliotekom

---

**Banjavčić, Josipa**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Pula / Sveučilište Jurja Dobrile u Puli**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:137:653500>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-01-12**



*Repository / Repozitorij:*

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli  
Fakultet informatike u Puli

**Josipa Banjavčić**

**DETEKCIJA AUTOMOBILSKIH TABLICA OPENCV BIBLIOTEKOM**

Diplomski rad

Pula, lipanj, 2020. godine.

Sveučilište Jurja Dobrile u Puli  
Fakultet informatike u Puli

**Josipa Banjavčić**

**DETEKCIJA AUTOMOBILSKIH TABLICA OPENCV BIBLIOTEKOM**

Diplomski rad

**JMBAG: 0303047593, redovita studentica**

**Studijski smjer: Sveučilišni diplomski studij informatike**

**Predmet: Umjetna inteligencija**

**Znanstveno područje: Društvene znanosti**

**Znanstveno polje: Informacijske i komunikacijske tehnologije**

**Znanstvena grana: Informacijski sustavi i informatologija**

**Mentor: doc. dr. sc. Darko Etinger**

Pula, lipanj, 2020. godine



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Josipa Banjavčić, kandidat za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

Josipa Banjavčić

U Puli, lipanj, 2020. godine



**IZJAVA**  
o korištenju autorskog djela

Ja, Josipa Banjavčić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Detekcija automobilskih tablica OpenCV bibliotekom“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 25.06.2020. (datum)

Potpis

*Josipa Banjavčić*

## Sadržaj

1. Uvod.....	1
2. Percepcija.....	3
2.1. Vid.....	4
2.2. Vizualna percepcija i dizajn korisničkog sučelja.....	5
3. Računalni vid.....	6
3.1. Glavne karakteristike.....	7
3.2. Primjena računalnog vida.....	8
4. Strojno učenje i prepoznavanje registracijskih oznaka.....	13
4.1. Stvaranje jezgre ML modela.....	14
4.2. Primjene strojnog učenja i kompjuterskog vida.....	15
5. Neuronske mreže i prepoznavanje registracijskih oznaka.....	16
5.1. Faze prepoznavanja korištenjem neuronskih mreža.....	16
5.2. Rad klasičnog ANPR sustava.....	17
6. Detekcija korištenjem Open CV biblioteke.....	20
6.1. Open CV.....	20
6.2. Ugrađene strukture podataka i ulaz / izlaz.....	22
6.2.2. Detekcija objekta.....	22
6.2.3. Detekcija teksta.....	22
7. Prepoznavanje registracijskih oznaka OpenCV bibliotekom.....	23
7.1. Grayscale.....	25
7.2. Otkrivanje rubova.....	26
7.3. Moguće registracijske oznake u sceni.....	29
7.4. Izdvajanje pronađenih područja.....	31

7.5. Potencijalne registracijske oznake.....	33
7.6. Obrada odabranih područja.....	34
7.7. Segmentacija.....	35
7.8. Rezultat prepoznavanja.....	37
7.8.1. Ispravna detekcija.....	39
7.8.2. Djelomična detekcija.....	40
Zaključak.....	42
Literatura.....	44
Popis slika.....	47
Sažetak.....	49
Summary.....	49

## 1. Uvod

Umjetna inteligencija – pojam koji se u današnje vrijeme pojavljuje u raznim područjima. Od SIRI-ja do automobila koji samostalno voze, umjetna inteligencija (AI) naglo napreduje. Dok znanstvena fantastika često predstavlja AI kao robote s karakteristikama sličnim ljudskima, AI može obuhvatiti sve, od Googleovih algoritama pretraživanja do autonomnog oružja. Upravo ta rasprostranjenost utjecaja AI motivira razna istraživanja na mnogim područjima, od ekonomije i prava do tehničkih tema kao što su provjera, valjanost, sigurnost i kontrola. Postaje sve važnije da AI sustav radi ono što treba ukoliko kontrolira neki automobil, neki avion, pejsmejker, neko automatizirano trgovanje sustava ili neku mrežu napajanja. Osim autonomije vozila, jedno od područja gdje se dosta veliki značaj stavlja na AI je i sam nadzor prometa. Budući da broj vozila na prometnicama konstantno raste, ljudski nadzor postaje nedovoljan i ponekada spor. Detekcija registracijskih oznaka, naplata cestarina i kontrola ograničenja brzine tako postaju dio procesa koji se pokušavaju u potpunosti prebaciti na računala. Detekcija registracijskih oznaka je posebno važna, posebno kod provedbe prometnih propisa jer je detekcija vozila i detekcija registracijske oznake početni korak u sustavima za nadzor prometa. Većina sustava koristi AI kako bi se cijeli proces nadzora olakšao. S time se dolazi i do teme ovog rada.

Detekcija registracijskih oznaka uz pomoć AI-a se bazira na ideji automatizacije. Radnje provjere registracijske oznake se obavljaju uz pomoć računala. Proces radi na način da se iz snimke vozila treba prepoznati područje koje sadrži registracijsku oznaku te iščitati podatke iz iste. Kako bi se moglo doći do tih rezultata, koriste se metode računalnog vida (Computer Vision – CV). Računalni vid je polje proučavanja kako računala vide i razumiju digitalne slike i videozapise. Računalni vid obuhvaća sve zadatke koje obavljaju sustavi biološkog vida, uključujući „gledanje“ ili osjet vizualnog podražaja, razumijevanje onoga što se vidi i vađenje složenih informacija u oblik koji se može koristiti u drugim procesima. Ovo interdisciplinarno polje simulira i automatizira ove elemente sustava ljudskog vida pomoću senzora, računala i algoritama strojnog učenja.



Računalni vid se temelji na sposobnosti sustava umjetne inteligencije da vide i razumiju svoje okruženje. Točnije, za potrebe rada se koristi biblioteka programskih funkcija uglavnom usmjerena na računalni vid u stvarnom vremenu, ili kraće OpenCV – biblioteka otvorenog koda koja se može koristiti sa različitim programskim jezicima i koja je besplatna za korištenje. Cilj rada je objasniti načine na koji se može provesti detekcija registracijske oznake, od toga najveći naglasak će biti stavljen na OpenCV detekciju.

## 2. Percepcija

Percepcija kod ljudi je nesvjesni proces kojim mozak organizira podatke dospjele iz raznih osjetila i interpretira ih u smislenu cjelinu. Percepcija ljudima omogućava da razne mrlje boje vide kao određeni predmet, da zvukove čuju kao govor, itd. Percepcija nije samo pasivno primanje ovih signala, već je oblikovana i primateljevim učenjem, pamćenjem, očekivanjem i pažnjom. Senzorni unos je proces koji pretvara ove podatke niske razine u informacije više razine (npr., izdvaja oblike za prepoznavanje predmeta). Proces koji slijedi povezuje čovjekove koncepte i očekivanja (ili znanje), restorativne i selektivne mehanizme (poput pažnje) koji utječu na percepciju. Percepcija ovisi o složenim funkcijama živčanog sustava, ali subjektivno se uglavnom događa bez napora jer se ta obrada događa izvan svjesne svijesti. Percepcijski sustavi mozga omogućuju pojedincima da svijet oko sebe vide kao stabilan, iako su osjetilne informacije obično nepotpune i brzo se mijenjaju. Ljudski i životinjski mozgovi strukturirani su na modularan način, s različitim područjima koja obrađuju različite vrste osjetilnih informacija. Neki od ovih modula imaju oblik senzornih karata, mapirajući neki aspekt svijeta preko dijela moždane površine (Epstein et al.). Ti različiti moduli međusobno su povezani i utječu jedan na drugog.

Percepcija kod računala se bazira na istom principu. Računala dobivaju informaciju o svijetu u kojem obitavaju tumačeći odziv senzora. Senzor mjeri neki aspekt okoliša u obliku koji se može koristiti kao ulaz. Senzor bi mogao biti jednostavan kao sklopka, gdje govori je li uključen ili isključen ili je složen kao oko. Umjetnim računalnim agentima dostupni su različiti senzorni modaliteti. Oni koje računalo dijeli s ljudima uključuju vid, sluh i dodir. Modaliteti koji nisu dostupni čovjeku uključuju radio, infracrvenu vezu, GPS i bežične signale. Neki roboti mogu aktivno promatrati, što znači da šalju signal, kao što je radar ili ultrazvuk, i osjećaju odaziv tog signala iz okoline (G. F. Luger, 2009). Umjesto da pokuša sve to obuhvatiti, ovo poglavlje će dubinski obraditi jedan modalitet: vid.

## 2.1. Vid

Vizualna percepcija zahtijeva niz transformacija neuronskih signala u mrežnici oka, stvarajući obrasce aktivnosti koji predstavljaju informacije potrebne za prepoznavanje, lokaciju i usmjeravanje aktivnosti.

Ovi prikazi počivaju na grupiranju informacija, u kojem pojedinačni neuroni nose jedinstvene informacije, a svaki neuron djeluje kao obilježena linija. Proces grupiranja stvaraju prikaze strukture, koji služe kao osnova za prepoznavanje predmeta, lica i scena u specijaliziranim područjima oka (Shimojo, Paradiso, Fujita, 2001).

Za vid, senzorski model može se podijeliti u dvije komponente:

- Objektivi model koji opisuje objekte koji obitavaju u vizualnom svijetu - ljude, zgrade, drveće, automobile itd. Objektivi model može uključivati precizan 3D geometrijski model snimljen iz računala - sustav potpomognutog dizajna (CAD), ili bi to mogla biti nejasna ograničenja, poput činjenice da ljudske oči obično daju model na udaljenosti od 7 cm.
- Model prikazivanja koji opisuje fizičke, geometrijske i statističke procese koji proizvode poticaj iz svijeta. Modeli ispisivanja prilično su precizni, ali su nejasni. Na primjer, bijeli objekt pri slabom svjetlu može se pojaviti iste boje kao i crni predmet pod jakom svjetlošću. Mali objekt u blizini može izgledati jednako kao i veliki udaljeni objekt.

Na primjer, sustav vida za autonomni automobil možda neće moći interpretirati predmete koji su daleko u daljini, ali agent može odlučiti da ignorira problem, jer je malo vjerojatno da će se sudariti u objekt koji je miljama daleko.

Proces pretvaranja zbilje u sliku narušava izgled predmeta. Na primjer, slika snimljena nizbrdo kroz dugačak pravac željezničkih pruga sugerira da se tračnice konvergiraju i sastaju. Kao još jedan primjer, ako čovjek pomiče ruku naprijed-natrag ili naginje tijekom stvaranja slike, ruka na slici će se smanjiti ili rasti, ali to u stvarnosti nije tako (P. Singh, 2018). Modeli ovih učinaka ključni su za prepoznavanje i rekonstrukciju.

## 2.2. Vizualna percepcija i dizajn korisničkog sučelja

Vizualna percepcija boja, uzoraka i struktura od posebnog je interesa u odnosu na grafička korisnička sučelja (GUI) jer se one percipiraju isključivo kroz viziju. Razumijevanje vizualne percepcije omogućava dizajnerima stvaranje učinkovitijih korisničkih sučelja.

U dizajnu GUI-a naširoko se koriste različite osobine vizualne percepcije. Mnogi dizajneri primjenjuju Gestalt principe (tj. kako ljudi strukturiraju vizualne podražaje) na dizajn GUI-ja kako bi stvorili sučelja koja korisnici lako mogu primijetiti i razumjeti. Vizualna percepcija djelovanja na okruženje još je jedan primjer kako je razumijevanje vizualne percepcije kritična stavka u alatima svakog dizajnera.

### 3. Računalni vid

Rani eksperimenti u računalnom vidu odvijali su se još 1950-ih, koristeći neke od prvih neuronskih mreža za otkrivanje rubova objekta i razvrstavanje jednostavnih objekata u kategorije poput krugova i kvadrata. U 1970-ima, prva komercijalna upotreba računalnog vida interpretirala je tipkani ili rukom pisani tekst koristeći optičko prepoznavanje znakova. To su napredovanje koristili za tumačenje pisanog teksta za slijepe.

Kako je internet postao široko dostupan u 1990-ima i kasnije, to je dovelo do nakupljanja velikih skupova slika koje su dostupne na mreži, a sa njima pojavili su se i programi prepoznavanja. Kako se ti skupovi podataka i dalje konstantno nadograđuju, oni uvelike olakšavaju proces prepoznavanja, pa tako imamo različite vrste prepoznavanja, od prepoznavanja određenih ljudi na fotografijama i videozapisima do prepoznavanja vrste životinja ili objekata.

Računalni vid (Computer Vision), često skraćeno kao CV, definira se kao polje studija koje nastoji razviti tehnike pomoću kojih će računala moći "vidjeti" i razumjeti sadržaj digitalnih slika, poput fotografija i video zapisa. To je multidisciplinarno polje koje bi se moglo nazvati ogrankom umjetne inteligencije i strojnog učenja i koje može uključivati upotrebu specijaliziranih metoda i korištenje općih algoritama učenja. Jedan od pokretačkih faktora rasta računalnog vida je količina podataka koju generiramo i koja se tada koristi za osposobljavanje i poboljšanje računalnog vida (D. Michie, et al, 1994).

Uz ogromnu količinu vizualnih podataka (više od 3 milijarde slika svakodnevno se dijeli na internetu), sada je dostupna i računalna snaga potrebna za njihovu analizu. Kako se polje računalnog vida povećalo s novim hardverima i algoritmima, tako se povećava i stupanj točnosti za identifikaciju objekata. U razdoblju manjem od desetljeća, sustavi su dostigli visok postotak točnosti što ih čini preciznijima od ljudi pri brzom reagiranju na vizualne ulaze.

### 3.1. Glavne karakteristike

Glavni čimbenici koji danas najviše utječu na pomake u računalnom vidu su:

- Mobilna tehnologija: konstantni napredak u mobilnoj industriji je doveo do ugradnje boljih kamera i samim time i boljom kvalitetom fotografija i videozapisa

Problem računalnog vida čini se jednostavnim jer ga trivijalno mogu riješiti ljudi, čak i vrlo mala djeca. Ipak, to uglavnom ostaje neriješen problem koji se temelji na ograničenom razumijevanju biološkog vida (Machinelearningmastery).

Da bismo maksimalno iskoristili slikovne podatke, potrebna su nam računala da "vide" sliku i razumiju sadržaj. Razumijevanje sadržaja digitalnih slika može uključivati izdvajanje opisa sa slike, koji može biti objekt, opis teksta, trodimenzionalni model i tako dalje. Kada su u pitanju ljudi, oni mogu:

- Opisati sadržaj fotografije koju su vidjeli jednom.
- Sažeti videozapis koji su vidjeli samo jednom.
- Prepoznati lice koje su vidjeli samo jednom prije.

Kod računala, kako bi se indeksirale i pretraživale slike algoritmi trebaju znati što slike sadrže. Dugo vremena sadržaj slika i videozapisa ostao je računalu nerazumljiv, potkrijepljen samo opisom koje je dala osoba koja ih je prenijela.

Računalni vid se razlikuje od obrade slike. Obrada slike proces je stvaranja nove slike iz postojeće slike, obično pojednostavljivanje ili poboljšanje sadržaja na neki način. To je vrsta digitalne obrade signala i ne bavi se razumijevanjem sadržaja slike. Dati se objekt može vidjeti iz bilo koje orijentacije, u bilo kakvim uvjetima osvjetljenja, bilo koje vrste odstupanja od drugih objekata, i tako dalje. Pravi sustav vida mora biti u stanju "vidjeti" u bilo kojem od beskonačnog broja scena i još uvijek izvući nešto smisljeno.

Problem se pojavljuje u činjenici da računala dobro funkcioniraju za čvrsto ograničene probleme, a ne za otvorene neograničene probleme poput vizualne percepcije

(Machinelearningmastery). Unatoč tome, u praksi je došlo do napretka, posebno posljednjih godina sa sustavima za optičko prepoznavanje znakova i prepoznavanje lica u fotoaparatom i pametnim telefonima.

### 3.2. Primjena računalnog vida

Napredak se najviše može uočiti na sljedećim područjima:

- Optičko prepoznavanje znakova (OCR)
- Pregled strojeva
- Navigacija za robote
- Maloprodaja (npr. Automatizirana naplata)
- Izgradnja 3D modela (fotogrametrija)
- Medicinsko snimanje
- Automobilska sigurnost
- Usklađivanje pokreta (npr. Spajanje CGI-a sa živim glumcima u filmovima)
- Snimanje pokreta (mocap)
- Nadzor
- Prepoznavanje otisaka prstiju i biometrija.

Mnoge popularne aplikacije za računalni vid uključuju pokušaje prepoznavanja stvari na fotografijama; na primjer (Brownlee, 2019):

- Klasifikacija objekata: Koja je kategorija predmeta na fotografiji?
- Identifikacija objekta: Koja je vrsta određenog objekta na fotografiji?
- Provjera objekta: Je li objekt na fotografiji?
- Otkrivanje objekta: Gdje su predmeti na fotografiji?
- Otkrivanje orijentira objekta: Koje su ključne točke objekta na fotografiji?
- Segmentacija objekta: Koji pikseli pripadaju objektu na slici?
- Prepoznavanje objekata: Koji su predmeti na fotografiji i gdje se nalaze?

Ostali su uobičajeni primjeri povezani s pronalaženjem informacija. Organizacija sustava računalnog vida ovisi o aplikaciji. Neki sustavi su samostalne aplikacije koje rješavaju određeni problem s mjerenjima ili detekcijom, dok drugi čine podsustav većeg dizajna koji, na primjer, također sadrži podsustave za upravljanje, planiranje, itd.

Specifična primjena sustava računalnog vida također ovisi o tome je li njegova funkcionalnost unaprijed određena ili se neki dio nje može naučiti ili izmijeniti tijekom rada. Mnoge su funkcije jedinstvene za aplikaciju. Postoje, međutim, tipične funkcije koje se nalaze u mnogim sustavima računalnog vida:

- Dohvaćanje slike - Digitalnu sliku proizvode jedan ili više senzora slike, koji osim različitih vrsta svjetlosno osjetljivih kamera uključuju senzore dometa, topografske uređaje, radar, ultrazvučne kamere itd.
- Predobrada - Prije nego što se računalna vizija može primijeniti na slikovne podatke da bi se izvukao neki određeni podatak, obično je potrebno obraditi podatke da bi se uvjerilo da zadovoljavaju određene pretpostavke koje metoda podrazumijeva.
- Ekstrakcija značajki - Iz slikovnih podataka izvlače se značajke slike na različitim razinama složenosti.
- Detekcija / segmentacija - U nekom trenutku obrade donosi se odluka o tome koje su točke slike ili regije slike relevantne za daljnju obradu.
- Obrada na visokoj razini - Na ovom koraku unos je obično mali skup podataka, na primjer, skup točaka ili područje slike za koje se pretpostavlja da sadrže određeni objekt.
- Donošenje odluka - Donošenje konačne odluke potrebne za aplikaciju. Na primjer:
  - Uspjeh/neuspjeh na aplikacijama za automatsku inspekciju.
  - Podudaranje/ne podudaranje u aplikacijama za prepoznavanje.

Prije pojave dubokog učenja, zadaci koje je računalni vid mogao obavljati bili su vrlo ograničeni i zahtijevali su mnogo ručnog kodiranja i napora od strane programera i ljudskih operatora. Zapravo je bilo uključeno vrlo malo automatizacije i većina se posla odvijala ručno (Russell, Norvig, 2010). A razina pogreške bila je i dalje velika.



Na primjer, za izvršiti prepoznavanje lica, bilo je potrebno izvršiti sljedeće korake:

- Stvorite bazu podataka: snimiti pojedinačne slike svih predmeta koji se žele pratiti u određenom formatu.
- Bilješke: za svaku pojedinačnu sliku potrebno unijeti nekoliko ključnih podataka, kao što su udaljenost između očiju, širina mosta nosa, udaljenost između gornje usne i nosa i deseci drugih mjerenja koja definiraju jedinstvene karakteristike svake osobe.
- Snimite nove slike: bilo da se radi o fotografijama ili video sadržaju koji zatim mora ponovo proći kroz postupak mjerenja, označavanjem ključnih točaka na slici.

Strojno učenje pružalo je drugačiji pristup rješavanju problema s računalnim vidom. Uz strojno učenje, programerima više nije potrebno ručno kodirati svako pravilo u svoje vizualne aplikacije. Umjesto toga, programirali su "značajke", manje aplikacije koje bi mogle otkriti specifične obrasce u slikama. Zatim su upotrijebili algoritam za statističko učenje poput linearne regresije, logističke regresije ili stabla odluka za otkrivanje uzoraka i klasificiranje slika i otkrivanje predmeta u njima.

Duboko učenje pružilo je bitno drugačiji pristup strojnom učenju. Duboko učenje oslanja se na neuronske mreže, općenitu funkciju koja može riješiti svaki problem koji se može prepoznati kroz primjere. Ako se neuronskoj mreži pruže podaci s mnogim obilježenim primjerima određene vrste podataka, može se doći do zajedničkih obrazaca između tih primjera i pretvoriti ih u matematičku jednadžbu koja će pomoći klasificirati buduće podatke (Mihajlovic, 2019). Na primjer, stvaranje aplikacije za prepoznavanje lica s dubokim učenjem zahtijeva samo da se razvije ili odabere unaprijed izgrađeni algoritam i uvježba ga s primjerima lica ljudi koje mora prepoznati. Ako ima dovoljno primjera (puno primjera), neuronska mreža moći će detektirati lica bez daljnjih uputa o značajkama ili mjerenjima.

Duboko učenje vrlo je učinkovita metoda računalnog vida. U većini slučajeva stvaranje dobrog algoritma dubokog učenja svodi se na prikupljanje velike količine označenih podataka o treningu i podešavanje parametara kao što su vrsta i broj slojeva neuronske mreže i epohe treninga. U usporedbi s prethodnim vrstama strojnog učenja, duboko se učenje lakše i brže razvija i implementira.

Primjene računalnog vida se danas protežu kroz razna područja, a najčešća primjena je:

- **Autonomna vozila:** Računalni vid omogućuje autonomnim vozilima da vide svoju okolinu. Kamere snimaju video iz različitih uglova oko automobila i prosljeđuju ga računalnim vidom koji potom slike u stvarnom vremenu obrađuje kako bi se pronašli rubovi cesta, pročitali prometni znakovi, otkrili drugi automobili, predmeti i pješaci. Takav automobil može samostalno upravljati prometnim ulicama i autocestama, izbjegavajući sudare i prepreke te sigurno voziti putnike do odredišta. Područja primjene autonomnih vozila uključuju i podmornice, kopnena vozila (mali roboti s kotačima, automobilima ili kamionima), zrakoplove i bespilotne letjelice (UAV). Razina autonomije kreće se od potpuno autonomnih (bespilotnih) vozila do vozila gdje sustavi utemeljeni na računalnom vidu podržavaju vozača ili pilota u različitim situacijama. Potpuno autonomna vozila obično koriste računalni vid za navigaciju, npr. za saznanje gdje se nalazi ili za izradu karte okruženja (SLAM) i za otkrivanje prepreka. Može se koristiti i za otkrivanje određenih događaja specifičnih za zadatak, npr. sustav koji traži šumske požare. Primjeri potpornih sustava su sustavi upozoravanja na prepreke u automobilima i sustavi za autonomno slijetanje zrakoplova. Nekoliko proizvođača automobila pokazalo je sustave za autonomnu vožnju automobila, ali ova tehnologija još uvijek nije dostigla razinu na kojoj se može staviti na tržište. Svemirska istraživanja već se obavljaju s autonomnim vozilima pomoću računalnog vida, npr., NASA-ine Curiosity i YNS-2 rovera CNSA-e.
- **Prepoznavanje lica:** Već prije spomenuto, računalni vid također igra važnu ulogu u aplikacijama za prepoznavanje lica, tehnologiji koja omogućava računalima da uslikane slike ljudi povežu s njihovim identitetima. Algoritmi računalnog vida

otkrivaju crte lica na slikama i uspoređuju ih s bazama profila lica. Potrošački uređaji (najviše smartphone) koriste prepoznavanje lica radi provjere identiteta svojih vlasnika. Aplikacije društvenih medija koriste prepoznavanje lica za otkrivanje i označavanje korisnika. Agencije za provođenje zakona također se oslanjaju na tehnologiju prepoznavanja lica za prepoznavanje kriminalaca na fotografijama i u video sažetcima.

- Proširena stvarnost: Računalni vid također igra važnu ulogu u proširenoj i mješovitoj stvarnosti, tehnologiji koja omogućuje računalnim uređajima poput pametnih telefona, tableta i pametnih naočala preklapanje i ugrađivanje virtualnih predmeta u slike stvarnog svijeta. Pomoću računalnog vida, otkrivaju se objekti u stvarnom svijetu kako bi se odredile lokacije na zaslonu uređaja za postavljanje virtualnog objekta.

## 4. Strojno učenje i prepoznavanje registracijskih oznaka

Strojno učenje (Machine learning) je podskupina umjetne inteligencije (AI) koja omogućuje sustavu za automatsko učenje da uči iz različitih scenarija, pružajući pritom relevantna, ali bolja iskustva. Kao primjer se može navesti Core ML, okvir dan od strane Apple-a koji podržava značajke kao što su praćenje lica, prepoznavanje lica, orijentacija, prepoznavanje teksta, otkrivanje pravokutnika, otkrivanje bar koda, praćenje objekata i prepoznavanje slike. Core ML je snažno integriran i podržava okvire Vision (analiza slike), Natural Language Processing (NLP) i GameplayKit. Istaknutiji okviri koji koriste Core ML su Tesseract i SwiftOCR. Apple je oba okvira koristio jedan za drugim. Dok je Tesseract okvir zahtijevao dosta vremena za postavljanje, SwiftOCR je bio relativno brži; međutim, nedostajalo je točnosti (Wilson, 2019). Eksperimentirajući s oba, Apple je lansirao Vision okvir za prepoznavanje lica, tekst, crtične kodove, QR kodove itd.

Neke od ključnih značajki Vision Framework-a su:

- Otkriva pravokutnik lica i orijentire lica
- Pronalazi površinu projicirane pravokutne regije.
- Pronalazi i prepoznaje bar kodove.
- Pronađite regije vidljivog teksta.
- Određuje kut horizonta u slici.
- Otkriva transformacije potrebne za usklađivanje sadržaja dviju slika.
- Obrađuje slike s Core ML modelom.
- Prati kretanja prethodno identificiranog proizvoljnog objekta preko više slika ili video okvira.

Apple je uz to sve htio kreirati Core ML model koji bi mogao prepoznati registarske tablice. Međutim, dobiven je modelom koji je mogao prepoznati samo tekst i ne samo cijelu registarsku tablicu. Model je proširen sa Keras bibliotekom koja se je bazirana na neuronskim mrežama otvorenog koda napisanima u Pythonu (Manchanda, 2020).

Model je dalje dizajniran da omogući brzo eksperimentiranje s dubokim neuronskim mrežama.

Prepoznavanje i otkrivanje registarskih tablica ključna je tehnika u većini aplikacija povezanih s kretanjem vozila. Postoje različite tehnike i algoritmi za otkrivanje i prepoznavanje registarskih tablica, ali rješenje temeljeno na obradi slike u obliku strojnog učenja je najpreciznije zbog sljedećih razloga:

- Na slici je pohranjeno puno podataka koje se mogu koristiti iz različitih perspektiva,
- Sposobnost rješavanja složenih zadataka vizualnog prepoznavanja,
- Identificiranje registarske tablice na učinkovit način bez potrebe za velikim ljudskim resursima.

Prepoznavanje registracijske tablice korištenjem strojnog učenja učinkovit je postupak otkrivanja i prepoznavanja iz razloga što se prepoznavanje obavlja pomoću koncepta optičkog prepoznavanja znakova (ili OCR) na slikama. To je obično proces koji uključuje:

- Otkrivanje registracijske pločice: smatra se najvažnijim korakom sustava koji određuje položaj same pločice. Slika vozila se smatra kao ulaz, što vraća registracijsku tablicu kao izlaz.
- Segmentacija znakova: U ovoj se fazi alfanumerički znakovi s registracijske tablice izvlače i segmentiraju u pojedinačne slike.
- Prepoznavanje znakova: svaki pojedini lik koji je ranije segmentiran prepoznaje se u ovoj fazi. Za obavljanje ovog koraka koristi se strojno učenje.

Sustavi za prepoznavanje registarskih tablica mogu se izraditi i bez korištenja strojnog učenja, tehnikama poput podudaranja predložaka, vađenja značajki itd. Međutim, korištenje strojnog učenja za razvoj sustava prepoznavanja poboljšava točnost sustava zbog boljeg trenajnog procesa.

## 4.1. Stvaranje jezgre ML modela

Dvije sljedeće funkcije moraju se obavljati pomoću okvira računalnog vida:

- 1) Otkrijte regije vidljivog teksta i podijelite sliku riječi u znakove
- 2) Obradite slike s Core ML modelom

Za glatku obradu model treba moći prepoznati cijelu registarsku tablicu. Dakle, da bi se to dogodilo, najčešće se koristi model kroz Keras - biblioteku neuronske mreže otvorenog koda napisanu u Pythonu za brzo eksperimentiranje s dubokim neuronskim mrežama. Modelu u python-u se dalje dodaju sljedeći okviri:

- Keras za izradu modela iz skupa podataka
- Dohvat piksela ciljne slike
- Numpy za višedimenzionalni niz visokih performansi
- Jezgrani ML alati za pretvorbu u ML model
- Sklearn za dijeljenje skupa podataka za trening i testiranje

Treniranje model se dalje vrši u svom preferiranom okviru. Dobiveni model se dalje koristi u ugrađenoj aplikaciji.

## 4.2. Primjene strojnog učenja i kompjuterskog vida

Neke of primjena gdje se strojno učenje koristi za prepoznavanje registracijskih oznaka mogu biti:

- Provjera napuštenih vozila: Ovo je bitna značajka za policijske snage u svrhu provođenja zakona. Ovdje im aplikacija može omogućiti da brzo provjere autentičnost registarskih tablica instaliranih na napuštenim vozilima putem mobilnih uređaja.

- Automatizacija elektronske naplate cestarine: Regulirani ulazak vozila kroz naplatne barijere može se učinkovito automatizirati s LPR-om. Pločice s brojevima navedenim u bazi podataka o naplati cestarine mogu se prepoznati sa uporabom LPR-a i sustava kontrole pristupa.

## 5. Neuronske mreže i prepoznavanje registracijskih oznaka

U zemljama koje se brzo razvijaju, broj vozila iz dana u dan se povećava. Paralelno s tim povećava se potreba za prepoznavanjem vozila i njihovih registarskih tablica. Da bi se osigurala ta potreba, konstantno se razvijaju novi sustavi za prepoznavanje registarskih tablica. Jedan od takvih učinkovitih sustava automatskog prepoznavanja registarskih oznaka vozila temeljen je na umjetnim neuronskim mrežama (ANN). Ovaj se sustav sastoji od tri glavne značajke. Ovi su; lokaliziranje regije pločice s slike automobila, segmentiranje znakova s slike registarske pločice i prepoznavanje segmentiranih znakova.

### 5.1. Faze prepoznavanja korištenjem neuronskih mreža

Prva faza sustava prepoznavanja registarskih tablica je pronalazak mjesta pločice na slici vozila. Područje registracije obično se sastoji od bijele pozadine i crnih znakova. Stoga su prijelazi između crne i bijele boje u ovoj regiji vrlo intenzivni. Pronalaženje regije koja uključuje većinu prijelaznih točaka bilo bi primjereno za lokalizaciju regije. U tu svrhu, Canny detektor rubova može biti jedan način utvrđivanja prijelaznih točaka.

Canny detektor ruba koristi filter na temelju prvog derivata Gaussovog izgladivanja. Nakon izravnavanja slike i uklanjanja buke, sljedeći korak je pronalazak čvrstoće ruba uzimanjem gradijenta slike. Za ovaj postupak ovaj operator koristi matrice dimenzije 3x3. Izračunava se rubna snaga gradijenta. Ovaj nam podatak daje rubne točke, pa se može odrediti regija intenzivnih prijelaznih točaka.

U ovom su radu slike sive ploče poboljšane primjenom produženja kontrasta i medijane filtriranja. Dakle, razlike u kontrastu između slika i zvukova poput prljavih područja na bijeloj pozadini tanjura mogu se ukloniti. Nakon faze poboljšanja slike, primijenjena je metoda bojanja mrlja kako bi se odredile granice znakova.



Korak u prepoznavanju čini i proširenje kontrasta slike. Drugim riječima, kontrastni nastavak čini sliku oštrijom i olakšava daljnje korake. Izjednačavanje histograma popularna je tehnika za poboljšanje izgleda loše slike s kontrastom.

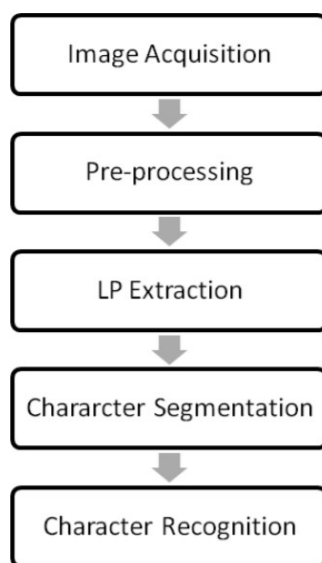
Kada su u pitanju neuronske mreže, mogu se koristiti razni pristupi i različiti modeli, a najčešći je obično onaj sa dva sloja. Od danog ulaza (uključuje informacije koje se koriste za donošenje odluke – koje proizlaze iz prethodnih koraka), informacije ulaze u skriveni sloj (pomaže mreži u izračunavanju složenijih asocijacija) i izlazni sloj (uključuje rezultirajuću odluku). Svaki se neuron u ulaznom sloju napaja izravno u neurone skrivenog sloja pomoću niza utega. Zbroj produkata utega i ulaza izračunava se u svakom čvoru. Izračunate vrijednosti dovode se izravno u neurone izlaznog sloja putem niza utega. Kao i u skrivenom sloju, zbroj produkata utega i izlaza neurona skrivenog sloja izračunava se u svakom čvoru u izlaznom sloju. Ako je pogreška između izračunate izlazne vrijednosti i željene vrijednosti veća od omjera pogreške, tada započinje trening (promjena utega i izračunavanje novog izlaza korištenjem novih utega). Ovaj se postupak obuke može završiti dobivanjem željene stope pogreške za sve kombinacije ulaza. Kad sustav dođe do minimalne pogreške stopa koja je definirana od strane korisnika, iteracije će se zaustaviti.

## 5.2. Rad klasičnog ANPR sustava

Automatsko prepoznavanje registracijskih tablica (ANPR), kao važno istraživačko polje koje se koristi u računalnom vidu, prepoznavanju uzoraka, obradi slike i umjetnoj inteligenciji, jedan je od najvažnijih aspekata inteligentnog transportnog sustava ljudskog društva u 21. stoljeću. Automatsko prepoznavanje registracijskih tablica sastoji se od četiri glavna odjeljka: pred obrada slike, vađenje (lokalizacija) pločica broja (registracijske tablice), segmentacija znakova i prepoznavanje znakova. ANPR se može koristiti za spremanje snimljenih slika kao i teksta s registracijske pločice. Tehnologija ANPR obično je specifična za regiju, zbog varijacije ploča od mjesta do mjesta. Cilj ANPR-a je uspješno lociranje registracijskih tablica vozila, segmenata i njihovo

prepoznavanje s obzirom na sliku automobila (Jain, Kundargi, 2015). Sustav mora imati različite kutove, udaljenosti, ljestvice, razlučivosti i uvjete osvjetljenja.

ANPR sustav koji iz određene slike izvlači registracijske tablice, može se sastojati od četiri faze. Prva faza je dohvaćanje slike automobila pomoću kamere gdje se parametri poput vrste kamere, rezolucije fotoaparata, brzine zatvarača, orijentacije i svjetla, moraju se uzeti u obzir. Druga faza je izdvajanje registracijskih pločica sa slike na temelju nekih značajki, kao što su granica, boja ili postojanje znakova. Treća faza je segmentiranje registracijske pločice i izdvajanje znakova projiciranjem njihovih informacija u boji, označavanjem ili podudaranjem njihovih položaja s predlošcima. Posljednja faza je prepoznavanje iščitanih znakova pomoću podudaranja predložaka ili korištenjem klasifikatora, poput umjetnih neuronskih mreža, SVM-ova i drugog (Jain, Kundargi, 2015). Slika 1 prikazuje strukturu ANPR procesa. Performanse ANPR sustava oslanjaju se na robusnost svake pojedine faze.



Slika 1. Faze ANPR sustava (Jain, Kundargi, 2015)

Uspjeh razvijenih metoda temelji se na njihovoj otpornosti na okolišne uvjete kao što su rasvjeta, složene pozadine, nedostaci na površini ploče, kao i na razmaku udaljenosti i pogleda između vozila i kamere. Većina ANPR može učinkovito raditi u različitim uvjetima.

Postoji sedam primarnih algoritama koje softver treba za prepoznavanje registarske pločice:

- Lokalizacija ploče - odgovorna je za pronalaženje i izoliranje ploče na slici.
- Orijehtacija i veličina ploče - kompenzira nagib i dimenzije prilagođava potrebnoj veličini.
- Normalizacija - podešava svjetlinu i kontrast slike.
- Segmentacija znakova - pronalazi pojedine znakove na pločicama.
- Optičko prepoznavanje znakova.
- Sintaktička / geometrijska analiza - provjerava znakove i položaje prema pravilima specifičnim za zemlju.
- Razdvajanje prepoznate vrijednosti na više polja / slika da bi se dobio pouzdaniji ili sigurniji rezultat. Pogotovo jer svaka pojedina slika može sadržavati odblesak svjetlosti, biti djelomično zatamnjen ili drugi privremeni efekt.

Složenost svakog od tih pododjeljka programa određuje točnost sustava. Postoji niz mogućih poteškoća s kojima se softver mora suočiti (Constant, 2009). Neke su: loša razlučivost datoteke (obično zato što je registracijska oznaka predaleko, ali ponekad rezultat upotrebe nekvalitetne kamere), zamućene slike, loša rasvjeta i nizak kontrast (zbog prekomjerne ekspozicije, refleksije ili sjene), predmet koji zakriva (dio) tablicu, promjena vozne trake vozila u kutu gledanja tijekom čitanja registarskih tablica, drugačiji font, tehnike obrezivanja, nedostatak koordinacije između država, itd.

ANPR je ponekad poznat i po raznim drugim izrazima:

- Automatsko (ili automatizirano) prepoznavanje registarske pločice (Automatic (or automated) license-plate recognition (ALPR))

- Automatski (ili automatizirani) čitač registarskih tablica (Automatic (or automated) license-plate reader (ALPR))
- Automatska identifikacija vozila (Automatic vehicle identification (AVI))
- Prepoznavanje automobilske ploče (Car plate recognition (CPR))

## 6. Detekcija korištenjem Open CV biblioteke

Kao što je već bilo spomenuto u prethodnim poglavljima, računalni vid je ljudski vid imitiran od strane računala. Glavna svrha je opis predmeta koje vidimo te rekonstrukcija njegovih značajki u obliku, osvjetljenju ili boji. Računalni vid koristi se u mnogim različitim područjima. Jedna od njih je obrada slike i videozapisa u svrhu otkrivanja objekata na slikama ili video datotekama.

Danas je vrlo česta primjena detekcije registracijskih oznaka na slikama automobila. Već postoji nekoliko sustava otkrivanja registracijskih tablica, koji se većinom koriste na parkiralištima sa plaćenim parkiranjem ili u područjima s ograničenim ulazom za vozila (prethodno spomenuti ANPR sustavi), što znači automatsko prepoznavanje registracijskih tablica. U ovom radu se želi predstaviti rješenje za otkrivanje registarskih tablica.

### 6.1. Open CV

Open CV (Open Source Computer Vision Library) je knjižnica softvera otvorenog koda za računalni vid i strojno učenje. Službeno pokrenut 1999. godine, projekt Open CV u početku je bio Intel Research inicijativa za unapređivanje aplikacija koje intenzivno koriste CPU, kao dio niza projekata, uključujući praćenje zračenja u stvarnom vremenu i 3D prikaz zidova (OpenCV). Glavni sudionici projekta bili su brojni stručnjaci za optimizaciju iz Intela Rusija. U ranim danima Open CV-a ciljevi projekta opisani su kao:

- Unaprjeđivanje istraživanja kompjuterskog vida pružajući ne samo otvoreni, već i optimizirani kod za osnovnu vidnu infrastrukturu.
- Širenje znanja o računalnoj viziji pružajući zajedničku infrastrukturu koju bi programeri mogli nadograditi kako bi taj kod bio čitljiviji i prenosivi.
- Stvaranje naprednih komercijalnih aplikacije temeljenih na računalnom vidu stavljajući prijenosni, optimizirani kod besplatno na raspolaganje. Budući da je

licenciran BSD (Berkeley Software Distribution) proizvod, Open CV tvrtkama olakšava upotrebu i izmjenu koda.

Biblioteka posjeduje više od 2500 optimiziranih algoritama, koji uključuju sveobuhvatan skup klasičnih i najmodernijih algoritama računalnog vida i strojnog učenja. Ovi se algoritmi mogu upotrijebiti za otkrivanje i prepoznavanje lica, prepoznavanje objekata, klasificiranje ljudskih radnji u videozapisima, praćenje pokreta kamere, praćenje pokretnih objekata, izvlačenje 3D modela objekata, spajanje slika zajedno za stvaranje slika visoke rezolucije cijelog prizora, pronalaženje sličnih slike iz baze podataka, uklanjanje crvenila u očima sa slika snimljenih bljeskalicom, praćenje pokreta očima, prepoznavanje scenografije, itd (Rosebrock, 2018). Open CV ima više od 47 tisuća ljudi korisnika i oko 18 milijuna preuzimanja. Knjižnica se uvelike koristi u tvrtkama, istraživačkim skupinama i državnim tijelima.

Zajedno s dobro poznatim tvrtkama kao što su Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, koje koriste biblioteku, postoje mnogi start up-ovi poput Applied Minds, VideoSurf i Zeitera, koji uveliko koriste Open CV. Neka od korištenja Open CV-a obuhvaćaju raspon zajedničkog spajanja slika s ulice, otkrivanja upada u nadzorni videozapis u Izraelu, praćenja minske opreme u Kini, pomaganja robotima u navigaciji i podizanju predmeta u Willow Garageu, otkrivanja nesreća utapanja u bazenima u Europi, izvođenja interaktivne umjetnosti u Španjolska i New York-u, provjeravanja piste u zračnim lukama za krhotine u Turskoj, pregledavanju naljepnice na proizvodima u tvornicama širom svijeta, brzom prepoznavanju lica u Japanu, itd..

Open CV ima sučelja u C ++, Python, Java i MATLAB-u i podržava Windows, Linux, Android i Mac OS. Open CV uglavnom naginje aplikacijama za detekciju u stvarnom vremenu i koristi MMX i SSE upute kada su dostupne (OpenCV). Trenutno se aktivno razvijaju potpuno opremljena CUDA i Open CL sučelja. Postoji preko 500 algoritama i otprilike 10 puta više funkcija koje ih sastoje ili podržavaju. Open CV je izvorno napisan na C ++ i ima predloženo sučelje koje neprimjetno radi sa STL spremnicima.

## 6.2. Ugrađene strukture podataka i ulaz / izlaz

OpenCV pruža puno ugrađenih osnova za rukovanje operacijama koje se odnose na obradu slike i kompjuterski vid. Ako je potrebno napisati nešto ispočetka, morat će se definirati stvari, kao što su slika, točka, pravokutnik i tako dalje. To je osnova za gotovo svaki algoritam kompjuterskog vida (Rosebrock, 2018). OpenCV sve ove osnovne strukture dobiva izvan okvira, a nalaze se u jezgri modula. Još jedna prednost je što su ove strukture već optimizirane za brzinu i memoriju, tako da se ne mora brinuti o detaljima implementacije. Na primjer, modul *imgcodecs* upravlja s čitanjem i pisanjem slikovnih datoteka.

### 6.2.1. Procesiranje slike

OpenCV dopušta radnje poput filtriranja slike, morfoloških operacija, geometrijskih transformacija, pretvorbe u boji, crtanja na slikama, histograma, analize oblika, analize pokreta, otkrivanja značajki i tako dalje.

### 6.2.2. Detekcija objekta

Otkrivanje objekta odnosi se na otkrivanje lokacije objekta na nekoj datoj slici. Ovaj postupak se ne odnosi na vrstu objekta. Ako se dizajnira detektor vozila, on će samo vratiti mjesto vozila na zadanoj slici. Otkrivanje položaja objekata vrlo je kritičan korak u mnogim sustavima kompjuterskog vida.

### 6.2.3. Detekcija teksta

Prepoznavanje teksta u određenoj sceni i prepoznavanje sadržaja također postaje sve važnije. Neke aplikacije uključuju prepoznavanje natpisnih pločica, prepoznavanje prometnih znakova za vlastite automobile, skeniranje knjiga radi digitalizacije sadržaja itd. Postoji modul zvan *tekst* koji sadrži različite algoritme za upravljanje otkrivanjem i prepoznavanjem teksta.

## 7. Prepoznavanje registracijskih oznaka OpenCV bibliotekom

Proces prepoznavanja registracijskih oznaka je najbolje opisati kroz primjer. Tako je u sklopu rada izrađen je i prateći projekt kojemu je cilj uspješno prepoznavanje registracijskih oznaka i uz pomoć kojega se lakše mogu opisati koraci prepoznavanja. Za potrebe projekta korišten je:

- Python, verzija 3.7
- OpenCV, verzija 4.2.0

Za potrebe testiranja rada programa korišteno je više različitih slika. Za početak, stavit će se fokus na dvije slike. Prva slika (slika 2) prikazuje dio osobnog automobila kojemu je registracijska oznaka je jasno vidljiva.



*Slika 2. Osobni automobil*

Druga slika (slika 3) prikazuje kolonu gradskih autobusa, ali samo na prvome je vidljiva registracijska oznaka. Razlika u ove dvije slike je u tome što su registracijske oznake, iako vidljive, različite veličine i postavlja se pitanje hoće li obje biti ispravno identificirane.





*Slika 3. Gradski autobus*

Kako bi se što jasnije iščitala registracijska oznaka, odabrane slike se pojednostavljaju. Prvo se radi pretvorba iz boje u nijanse sive, a zatim se otkrivaju rubovi koji se dalje pojednostavljaju dok se ne dođe do potencijalne registracijske oznake. Na slici 4 je dio koda koji radi te pretvorbe.

```
def preprocess(imgOriginal):
    imgGrayscale = extractValue(imgOriginal)

    imgMaxContrastGrayscale = maximizeContrast(imgGrayscale)

    height, width = imgGrayscale.shape

    imgBlurred = np.zeros((height, width, 1), np.uint8)

    imgBlurred = cv2.GaussianBlur(imgMaxContrastGrayscale, GAUSSIAN_SMOOTH_FILTER_SIZE, 0)

    imgThresh = cv2.adaptiveThreshold(imgBlurred, 255.0, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                     cv2.THRESH_BINARY_INV, ADAPTIVE_THRESH_BLOCK_SIZE,
                                     ADAPTIVE_THRESH_WEIGHT)

    return imgGrayscale, imgThresh
```

*Slika 4. OpenCV kod za procesiranje slike*

## 7.1. Grayscale

Postupak kojim se slika pretvara u nijanse sive (crne i bijele). Mnogo funkcija u OpenCV-u može pretvoriti sliku u sivi ton prije obrade (Rouse, 2010). Grayscale se koristi zato što pojednostavljuje sliku i ubrzava vrijeme obrade jer je na slici manje podataka (jer su sive boje pohranjene u dvodimenzionalnim nizovima). Slika 5 i slika 6 prikazuju grayscale odabranih slika.



*Slika 5. Grayscale osobnog vozila*



*Slika 6. Grayscale gradski autobus*

## 7.2. Otkrivanje rubova

Detekcija ruba uključuje razne matematičke metode koje imaju za cilj identificiranje točaka na digitalnoj slici kod kojih se svjetlina slike oštro mijenja ili ima diskontinuitete. Točke u kojima se svjetlina slike naglo mijenja, obično su organizirane u skup zakrivljenih segmenata linija nazvanih rubovima. Isti problem pronalaženja diskontinuiteta u jednodimenzionalnim signalima poznat je kao detekcija koraka, a problem pronalaženja diskontinuiteta signala tijekom vremena poznat je i kao otkrivanje promjena (Sicara.ai). Otkrivanje rubova temeljno je sredstvo za daljnju obradu.

Nakon što smo izračunali mjeru čvrstoće ruba (obično veličine gradijenta), sljedeća faza je primjena praga i odlučivanje postoje li rubovi u slici ili ne. Niži prag, više rubova će se otkriti, a rezultat će biti sve osjetljiviji na buku i otkrivanje rubova nebitnih značajki na slici. Suprotno tome, visok prag može propustiti suptilne rubove ili rezultirati fragmentiranim ivicama.

Canny otkrivanje ruba je tehnika za izdvajanje korisnih strukturnih informacija iz različitih vidnih objekata i dramatično smanjenje količine podataka koje treba obraditi. Široko se primjenjuje u raznim sustavima računalnog vida (OpenCV Tutorialspoint, 2019).

Opći kriteriji za otkrivanje ruba uključuju:

- Detekcija ruba s niskom stopom pogreške, što znači da bi otkrivanje trebalo točno uhvatiti što više rubova prikazanih na slici
- Rubna točka trebala bi se točno lokalizirati na sredini ruba.
- Dati rub na slici treba biti označen samo jednom, a kad je to moguće, šum slike ne bi trebao stvarati lažne rubove.

Kao korak pred-obrade do otkrivanja ruba, gotovo se uvijek izvodi faza izgladivanja, tipično zvana Gaussovo izgladivanje. Ovaj korak će malo izgladiti sliku kako bi se umanjili učinci očite buke na rubnom detektoru (slika 7 i 8).



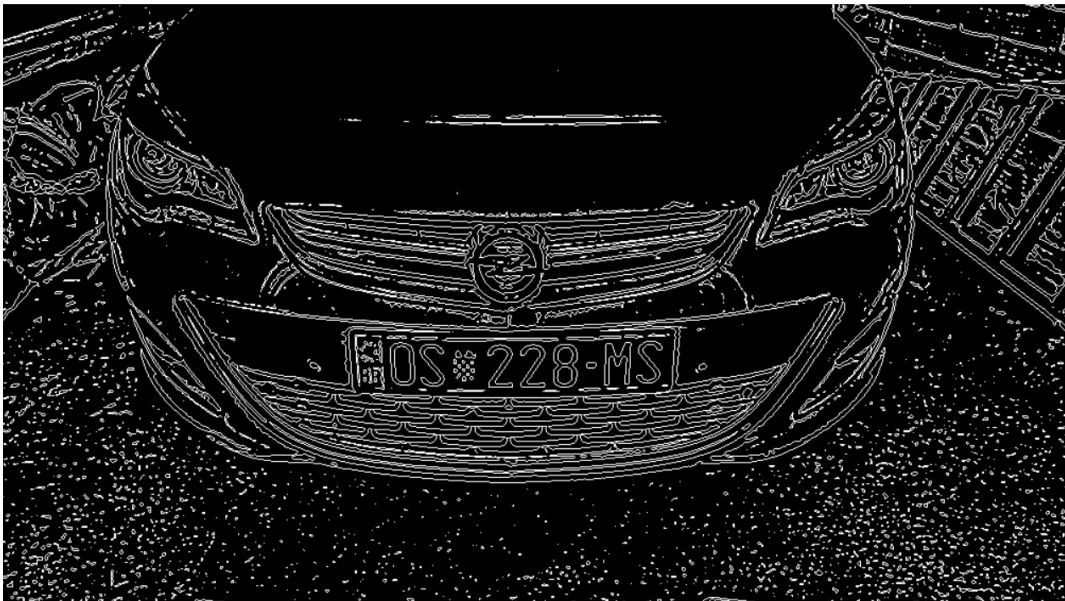
*Slika 7. Osobni automobil – korak otkrivanja rubova*



*Slika 8. Gradski autobus – korak otkrivanja rubova*

Sljedeći korak je prilagođavanja otkrivenih rubova (OpenCV). Kod OpenCV-a koristi se metoda Adaptive threshold u kojoj se vrijednost praga izračunava za manje regije pa će zbog toga postojati različite vrijednosti praga za različite regije (slika 9 i 10). Ova metoda prihvaća sljedeće parametre:

- src - Objekt klase Mat koji predstavlja izvornu (ulaznu) sliku.
- dst - Objekt klase Mat koji predstavlja odredišnu (izlaznu) sliku.
- maxValue - varijabla dvostrukog tipa koja predstavlja vrijednost koja se mora dati ako je vrijednost piksela veća od granične vrijednosti.
- adaptiveMethod - varijabla cjelobrojnog tipa koja predstavlja prilagodljivu metodu koja se koristi. To će biti bilo koja od sljedeće dvije vrijednosti: ADAPTIVE\_THRESH\_MEAN\_C - vrijednost praga je srednja vrijednost susjednog područja. ADAPTIVE\_THRESH\_GAUSSIAN\_C - vrijednost praga je ponderirani zbroj susjednih vrijednosti gdje su utezi Gaussov korak.
- PragmaType - varijabla cijelog broja koja predstavlja vrstu praga koji se koristi.
- blockSize - varijabla cjelobrojnog tipa koja predstavlja veličinu susjedstva piksela koja se koristi za izračunavanje vrijednosti praga.
- C - Varijabla dvostrukog tipa koja predstavlja konstantu koja se koristi u obje metode (oduzima se od srednje ili ponderirane srednje vrijednosti).



*Slika 9. Osobni automobil – korak Adaptive threshold*



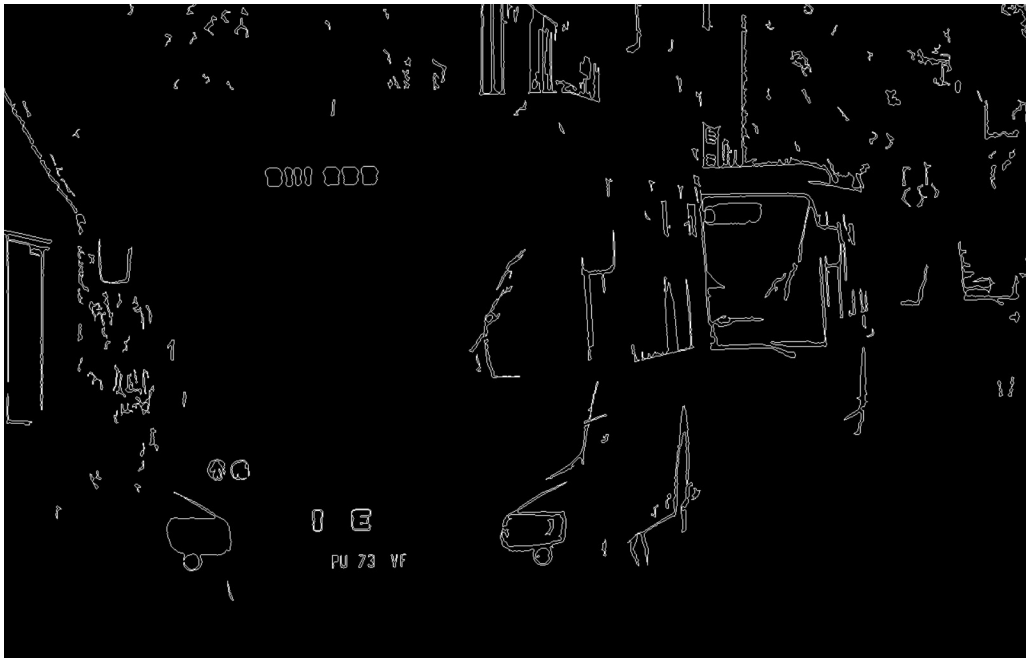
Slika 10. Gradski autobus – korak Adaptive threshold

### 7.3. Moguće registracijske oznake u sceni

Najjednostavnije objašnjeno, sljedeći korak iz slike pokušava iščitati područja koja bi mogla sadržavati moguće registracijske oznake.

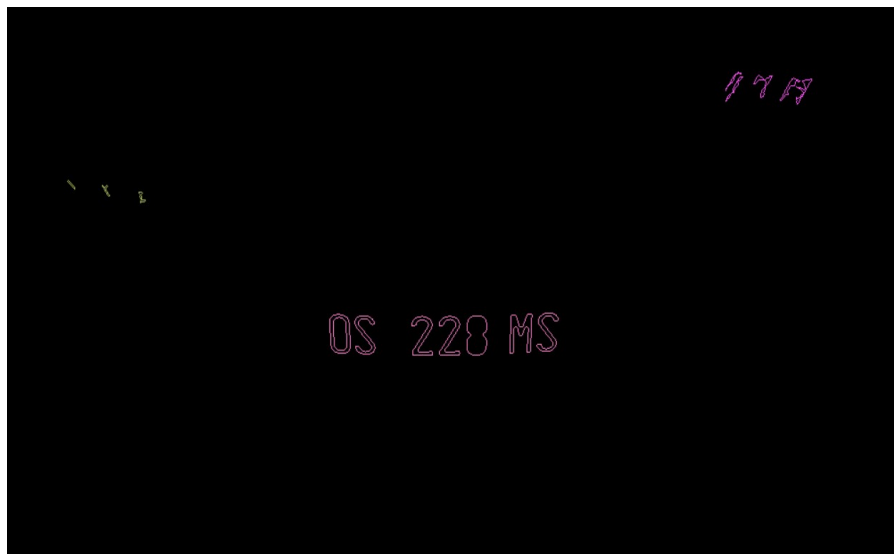


Slika 11. Osobni automobil – sužavanje područja na moguće registracijske oznake

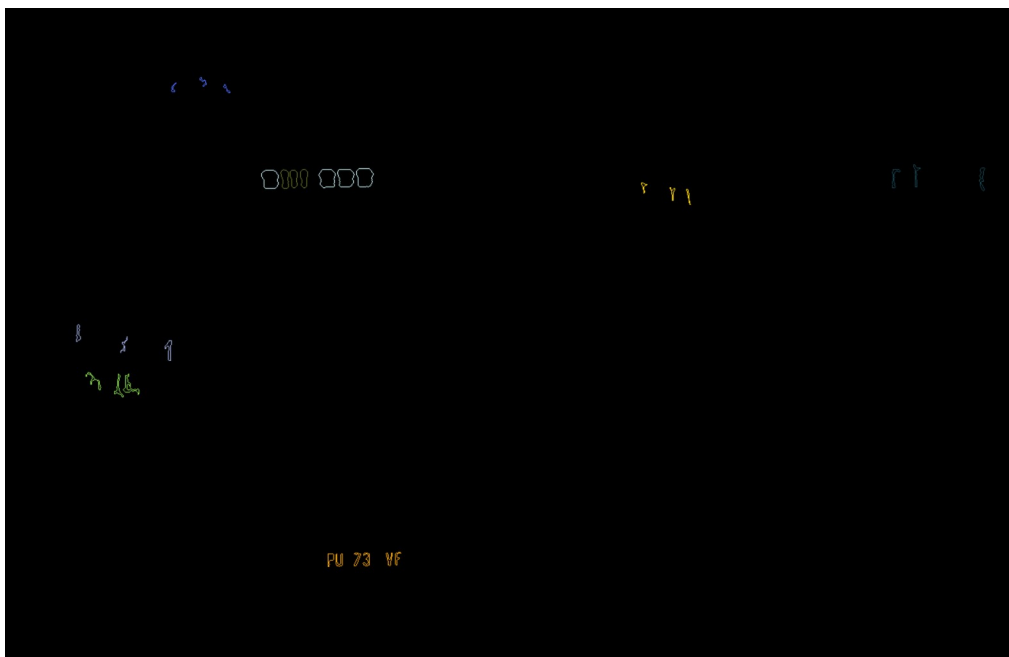


*Slika 12. Gradski autobus - sužavanje područja na moguće registracijske oznake*

Nakon što su utvrđena sva moguća područja, sljedeći korak je pronalazak grupa podudaranja. Svaka skupina podudaranja će pokušati biti prepoznana kao potencijalna registracijska oznaka. Ta područja se obilježavaju različitim bojama, a svaka boja je potencijalna oznaka (slike 13 i 14).



*Slika 13. Osobni automobil – izdvojena područja*



*Slika 14. Gradski autobus – izdvojena područja*

Kod slučaja osobnog vozila, program pronalazi 3 grupe koje kategorizira u potencijalne registracijske oznake (slika 13). Kod slučaja sa autobusom, program pronalazi 7 grupa (slika 14).

#### 7.4. Izdvajanje pronađenih područja

Iz prethodno pronađenih grupa, potrebno je izdvojiti područja potencijalnih registracijskih oznaka. Kako bi se oznake uspješno izdvojile potrebno je izračunati gdje se nalazi centar pronađene oznake (slika 15).

```
# izracun centra moguće oznake
fltPlateCenterX = (listOfMatchingChars[0].intCenterX + listOfMatchingChars[len(listOfMatchingChars) - 1].intCenterX) / 2.0
fltPlateCenterY = (listOfMatchingChars[0].intCenterY + listOfMatchingChars[len(listOfMatchingChars) - 1].intCenterY) / 2.0
ptPlateCenter = fltPlateCenterX, fltPlateCenterY
```

*Slika 15. Kod za izračun centra potencijalne registracijske oznake*



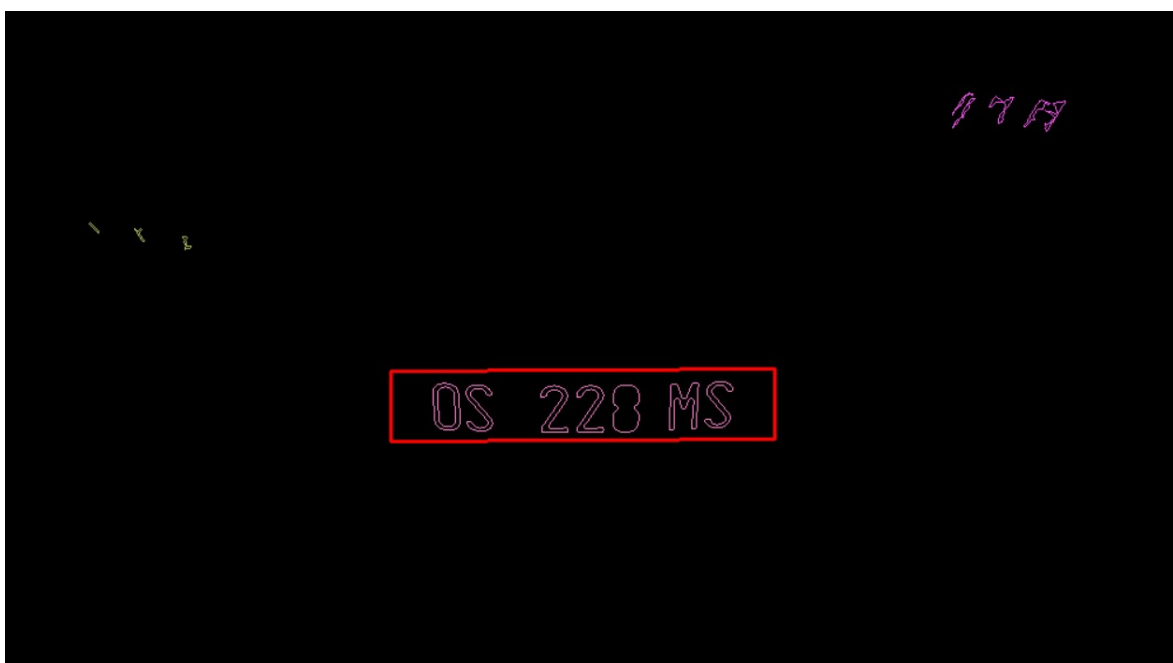
Također, važno je izračunati i visinu i širinu potencijalne oznake (slika 16). Dalje se rade potencijalne korekcije kako bi se oznaka što više prilagodila.

```
# izracun visine i sirine
intPlateWidth = int((listOfMatchingChars[len(listOfMatchingChars) - 1].intBoundingRectX +
                    listOfMatchingChars[len(listOfMatchingChars) - 1].intBoundingRectWidth -
                    listOfMatchingChars[0].intBoundingRectX) * PLATE_WIDTH_PADDING_FACTOR)

intTotalOfCharHeights = 0
```

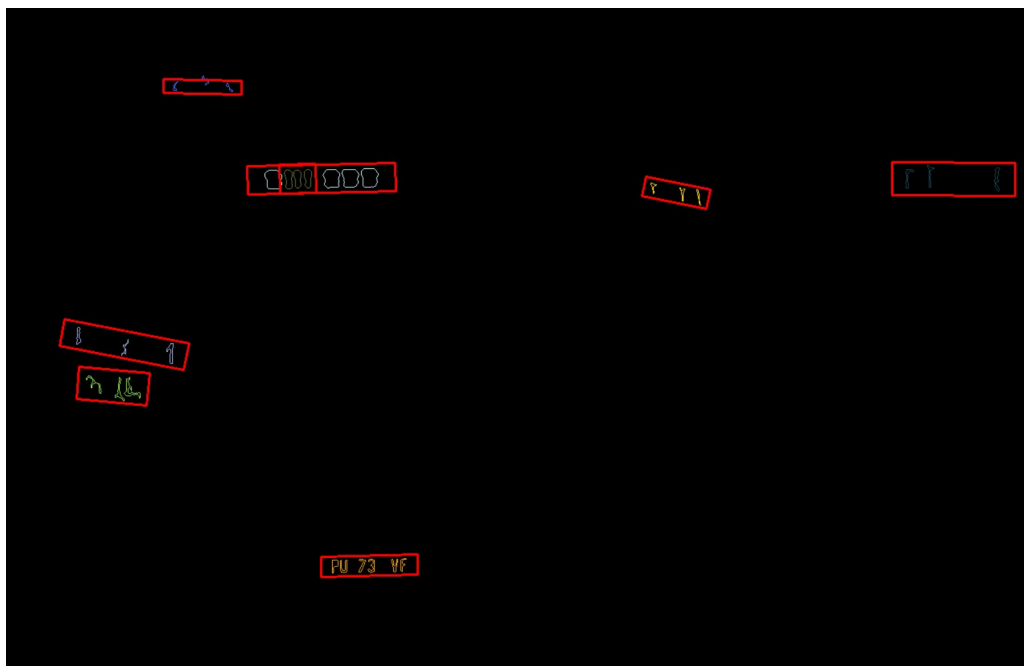
*Slika 16. Kod za izračun visine i širine potencijalne registracijske oznake*

Nakon svih korekcija, područja potencijalnih registracijskih oznaka se označavaju na način da program treba nacrtati područje oko potencijalne oznake u obliku okvira (slika 17 i 18).



*Slika 17. Osobni automobil – područje oznake*

Kod slučaja sa osobnim automobilom, iako su u prethodnim koracima bila utvrđena tri područja potencijalnih oznaka, program označava samo jednu crvenim okvirom.

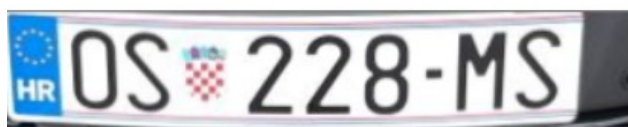


*Slika 18. Gradski autobus – područje oznake*

S druge strane, kod slučaja gradskog autobusa, program označava svih sedam prethodno prepoznatih područja i stavlja okvire oko njih.

### 7.5. Potencijalne registracijske oznake

Prethodno pronađene grupe se dalje koriste na način da se izdvojena područja detektiraju na polaznim slikama (slika 2 i 3). Područja se izdvajaju (izrezuju) od originalnih slika. U slučaju osobnog automobila, program izdvaja tri potencijalna mjesta koja bi mogla sadržavati registracijsku oznaku (slike 19, 20 i 21).



*Slika 19. Slučaj 1 – osobni automobil*



*Slika 20. Slučaj 2 – osobni automobil*

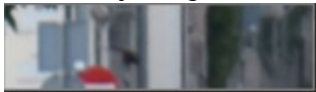


*Slika 21. Slučaj 3 – osobni automobil*

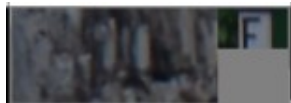
Kod slučaja gradskog autobusa, program izdvaja sedam potencijalnih mjesta koja bi mogla sadržavati registracijsku oznaku(slike 22, 23, 24, 25, 26, 27 i 28) .



*Slika 22. Slučaj 1 – gradski autobus*



*Slika 23. Slučaj 2 – gradski autobus*



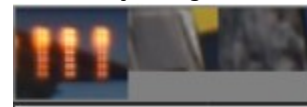
*Slika 24. Slučaj 3 – gradski autobus*



*Slika 25. Slučaj 4 – gradski autobus*



*Slika 26. Slučaj 5 – gradski autobus*



*Slika 27. Slučaj 6 – gradski autobus*

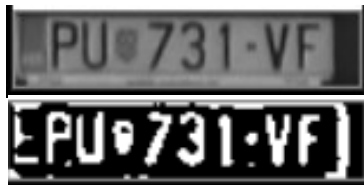


*Slika 28. Slučaj 7 – gradski autobus*

## 7.6. Obrada odabranih područja

Nakon uspješnog izdvajanja, prethodno dobivene slike ponovno prolaze kroz već opisane korake: pretvorba u sive tonove i odabir rubova (slika 29 i 30).

*Slika 29. Osobni automobil*



*Slika 30. Gradski autobus*

Nakon toga, koristi se novi korak – prepoznavanje znakova. Proces prepoznavanja se naziva segmentacija.

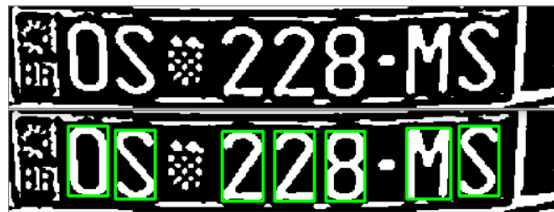
## 7.7. Segmentacija

Segmentacija znakova najvažniji je korak za bilo koji OCR (optički sustav prepoznavanja znakova). Odabir algoritma segmentacije koji se koristi ključni je faktor u odlučivanju o točnosti OCR sustava (Sharma, 2019). Ako postoji dobra segmentacija znakova, točnost prepoznavanja će također biti visoka.

Metode segmentacije mogu biti razvrstane u tri kategorije: metode praga, ruba i regije. Metode praga općenito koriste histogram ulazne slike za otkrivanje pojedinih ili više pragova. Tehnike segmentacije na temelju ruba imaju za cilj otkriti rubove ulazne slike. Tako se segmentacijom upravlja određivanjem granica regije na ulaznoj slici. Tehnike

segmentacije slike temeljene na regiji u početku pretražuju neke rubne točke u ulaznoj slici i primjenjuju se pristupi rastućeg područja kako bi se dosegle granice objekata.

Mnogo je čimbenika koji uzrokuju poteškoće u segmentaciji kao što su šum slike, okvir, razmak, rotacija i variranje osvjetljenja.



*Slika 31. Segmentacija – osobni automobil*

Kod slučaja sa osobnim automobilom (slika 31), rezultat segmentacije je zeleni okvir oko svakog karaktera u registracijskoj oznaci, tj. program je označio ono što je i trebao. S druge strane, slučaj gradskog autobusa (slika 32) daje malo drukčiji rezultat. Jedan karakter je ostao neprepoznat, a ostali su, kao i kod slučaja sa osobnim automobilom, označeni zelenim okvirom.



*Slika 32. Segmentacija – gradski autobus*

Kod iza cijelog procesa je sljedeći (slika 33).

```

def recognizeCharsInPlate(imgThresh, listOfMatchingChars):
    strChars = ""
    height, width = imgThresh.shape
    imgThreshColor = np.zeros((height, width, 3), np.uint8)
    listOfMatchingChars.sort(key = lambda matchingChar: matchingChar.intCenterX)

    cv2.cvtColor(imgThresh, cv2.COLOR_GRAY2BGR, imgThreshColor)

    # for petlja za svaki karakter u oznaki
    for currentChar in listOfMatchingChars:
        pt1 = (currentChar.intBoundingRectX, currentChar.intBoundingRectY)
        pt2 = ((currentChar.intBoundingRectX + currentChar.intBoundingRectWidth),
              (currentChar.intBoundingRectY + currentChar.intBoundingRectHeight))

        # zeleni okvir oko karaktera
        cv2.rectangle(imgThreshColor, pt1, pt2, Main.SCALAR_GREEN, 2)

        # izdvajanje karaktera
        imgROI = imgThresh[currentChar.intBoundingRectY : currentChar.intBoundingRectY + currentChar.intBoundingRectHeight,
                          currentChar.intBoundingRectX : currentChar.intBoundingRectX + currentChar.intBoundingRectWidth]

        # promjena velicine slike, vazno za prepoznavanje
        imgROIResized = cv2.resize(imgROI, (RESIZED_CHAR_IMAGE_WIDTH, RESIZED_CHAR_IMAGE_HEIGHT))

        #pretvara sliku u 1d numerički niz
        npaROIResized = imgROIResized.reshape((1, RESIZED_CHAR_IMAGE_WIDTH * RESIZED_CHAR_IMAGE_HEIGHT))

        # pretvara 1d numpy niz int-ova u 1d numpy niz float-ova
        npaROIResized = np.float32(npaROIResized)

        retVal, npaResults, neigh_resp, dists = kNearest.findNearest(npaROIResized, k = 1)

        strCurrentChar = str(chr(int(npaResults[0][0])))          # dohvaca karakter iz rezultata

        strChars = strChars + strCurrentChar                    # dodaje trenutni znak cijelom nizu
    # end for

```

*Slika 33. Kod za segmentaciju*

## 7.8. Rezultat prepoznavanja

Nakon uspješne segmentacije, prepoznati karakteri se iščitavaju i program daje rezultat prepoznavanja. Ukoliko je cijeli proces prepoznavanja uspješan, rezultat će biti točan ispis registracijske oznake, a ukoliko su se pojavili nedostaci u prepoznavanju, rezultat će biti djelomičan.



*Slika 34. Osobni automobil – rezultat*



*Slika 35. Gradski autobus – rezultat*

U prethodnim koracima, program je na obje slike dobro označio područje gdje se nalazi registracijska oznaka, ali budući da u dijelu segmentacije nije prepoznat jedan znak u slučaju sa gradskim autobusom, rezultat očitavanja također ne vrati ispravno očitavanje.

Tako za registracijsku oznaku PU 731 VF, program daje djelomično očitavanje PU 73 VF. Razloga za djelomično očitavanje može biti više. Različite veličine registracijskih oznaka na slikama, različito osvjetljenje slike ili čak razlika u poziciji registracijske oznake na slikama.

### 7.8.1. Ispravna detekcija

Slijede dva primjera ispravne detekcije. Slika 36 prikazuje samo registracijsku oznaku i služila je kao jedna od početnih slika.



*Slika 36. Ispravna detekcija – primjer samo sa tablicom*

Slika 37. je također primjer ispravne detekcije, a riječ je o personaliziranoj registracijskoj oznaci.



*Slika 37. Ispravna detekcija – personalizirana oznaka*



### 7.8.2. Djelomična detekcija

Slijede tri primjera djelomične detekcije. Slika 38 prikazuje da je za tablicu KA 219 IK, dan rezultat KA 28 K.



Slika 38. Djelomična detekcija – primjer 1

Slika 39 prikazuje još jedan primjer sa gradskim autobusom. Ponovno, rezultat je djelomičan jer za oznaku PU 412 LL dolazi rezultat PU 12 LL.



Slika 39. Djelomična detekcija – primjer 2

Na kraju, slika 40 je također primjer djelomične detekcije. Na slici je još jedna personalizirana registracijska oznaka GOOGLE. Rezultat iščitavanja te tablice je 6006LE.



*Slika 40. Djelomična detekcija – primjer 3*

Na početku procesa prepoznavanja, postavljeno je pitanje hoće li detekcija biti ispravna. Gledajući rezultate, može se zaključiti da u određenim situacijama hoće, dok u drugim baš i ne. Kao što je već spomenuto, razlozi za to mogu biti razni, od veličine registracijske oznake, kvaliteti snimljene fotografije, pozicije, pa do osvjetljenja. Naravno, točnost prepoznavanja se može poboljšati na način da se provedu daljnja treniranja i daljnje dorade. Upravo ta konstantna poboljšanja bi mogla opisati cijeli proces korištenja računalnog vida – potrebne su konstantne dorade kako bi se osiguralo da će točnost biti što veća. Kada je riječ o prepoznavanju registracijskih oznaka, da bi se postigla što veća točnost potrebno je istrenirati računalni vid na način da može prepoznati oznaku u bilo kojem osvjetljenju, da ima sposobnost prepoznati oznaku i na snimci lošije kvalitete, da se oznaka prepozna iz različitih kutova gledanja, da prepozna razne oblike registracijskih oznaka (budući da se oznake razlikuju ovisno o tome koja država ih daje), i tako dalje.

## Zaključak

Sigurnost je oduvijek bila glavna briga za čovječanstvo. Informacijska tehnologija ubrzano napreduje s obzirom na automatizirane sustave. U takvim sustavima ljudi koriste računalno utemeljene stručne sustave za analizu i rješavanje stvarnih problema. Od bolnica, škola, prometnica pa do svih drugih javnih mjesta, uvijek se radi na tome da se usavrši proces prepoznavanja. Problem računalnog vida čini se jednostavnim jer ga trivijalno rješavaju ljudi, čak i vrlo mala djeca. Ipak, to uglavnom ostaje neriješen problem koji se temelji i na ograničenom razumijevanju biološkog vida, ali i zbog složenosti percepcije vida u dinamičnom i gotovo beskonačno promjenjivom fizičkom svijetu. Na računalu je slika - kao i sve slike - niz piksela, numeričkih vrijednosti koje predstavljaju nijanse crvene, zelene i plave. Jedan od izazova s kojim se računalni znanstvenici suočavaju od pedesetih godina 20. stoljeća jest stvaranje sustava koji mogu imati smisla za razumijevanje fotografija i videozapisa kao što to čine ljudi. U današnje vrijeme se čini kako su prepoznavanje lica, prepoznavanje otiska prsta i prepoznavanje vozila tri najbitnije vrste prepoznavanja i najviše sustava se razvija na temelju toga.

Prepoznavanje registarskih tablica je potpuno automatizirana tehnika u stvarnom vremenu koja se široko koristi za identifikaciju, kontrolu krađe i provjeru sigurnosti vozila. Potreban je razvoj brzih i učinkovitih automatiziranih sustava koji uklanjaju pogreške uzrokovane ljudskim čimbenicima i daju dosljedne rezultate. Za prepoznavanje i čitanje željenih područja registarske pločice vozila koriste se različiti algoritmi. Prepoznavanje registracijskih oznaka je posebno važno i konstantno se radi na sustavima koji će točno prepoznati registracijsku oznaku u bilo kojim uvjetima, a budući da su ljudski naponi najčešće prespori za takve procese, radi se na automatizaciji tih sustava – gdje u cijelu priču ulazi umjetna inteligencija. Prepoznavanje registracijskih oznaka korištenjem umjetne inteligencije je važan dio za nadzor i kontrolu prometa.

Bez obzira koja se kombinacija metoda koristi, bit razvoja takvih sustava je u njihovoj točnosti. Cilj rada je na jedan način i bio da se provjeri koliko će točno program prepoznati dane registracijske oznake. Također, bitan je i proces kojim se radi prepoznavanje. Za potrebe ovoga rada korištene su prethodno snimljene fotografije vozila, a uz dorade, cijeli projekt bi se mogao prenamijeniti da analizira snimke u realnom vremenu. Specifičnost OpenCV-a je upravo u tome da se može koristiti i za realno vrijeme snimanja i za prethodno napravljene snimke. Također, privlačnosti OpenCV-a je u tome da pruža pristup više od 2500 vrhunskih i klasičnih algoritama. OpenCV pruža algoritamsku učinkovitost uglavnom za obradu programa u stvarnom vremenu.

## Literatura

Knjige:

1. Stuart J. Russell, Peter Norvig (2010): Artificial Intelligence: A Modern Approach (Third Edition)
2. George F. Luger (2009): AI: Structures and Strategies for Complex Problem Solving (Six Edition)
3. Wolfgang Ertel (2009): Introduction to Artificial Intelligence
4. Nils J. Nilsson (2009): The quest for artificial intelligence
5. Michael Nielsen: Neural Networks and Deep Learning
6. D. Michie, D.J. Spiegelhalter, C.C. Taylor (1994): Machine Learning, Neural and Statistical Classification

Članci i internet izvori:

7. W. Epstein, W. N. Dember, L. Jolyon West (2005): Perception: <https://www.britannica.com/topic/perception> (2020-02-10)
8. Jason Brownlee (2019): A Gentle Introduction to Computer Vision: <https://machinelearningmastery.com/what-is-computer-vision/> (2019-11-10)
9. Mike Constant (2009): An Introduction to ANPR: [https://www.cctv-information.co.uk/i/An\\_Introduction\\_to\\_ANPR](https://www.cctv-information.co.uk/i/An_Introduction_to_ANPR) (2019-05-10)
10. D. Lippaiová, M. Michalko, O. Kainz, F. Jakab (2018): Automatic license plate recognition system using OpenCV library: [https://www.academia.edu/37196246/AUTOMATIC\\_LICENSE\\_PLATE\\_RECOGNITION\\_SYSTEM\\_USING\\_OPENCV\\_LIBRARY](https://www.academia.edu/37196246/AUTOMATIC_LICENSE_PLATE_RECOGNITION_SYSTEM_USING_OPENCV_LIBRARY) (2019-11-10)

11. Ilija Mihajlovic (2019): Everything You Ever Wanted To Know About Computer Vision: <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e> (2019-12-30)
12. Amit Manchanda (2020): Using Apple's Machine Learning for License Plate Recognition: <https://www.netsolutions.com/insights/using-apples-machine-learning-for-license-plate-recognition/> (2020-02-15)
13. Adrian Rosebrock (2018): OpenCV Tutorial: A Guide to Learn OpenCV: <https://www.pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/> (2019-05-10)
14. Margaret Rouse (2010): Grayscale: <https://whatis.techtarget.com/definition/grayscale> (2019-12-10)
15. Anand S. Jain, Jayshree M. Kundargi (2015): Automatic Number Plate Recognition Using Artificial Neural Network: International Research Journal of Engineering and Technology (IRJET): Volume: 02 Issue: 04: <https://pdfs.semanticscholar.org/0102/638e8a5343f56856a3eb97796a1459640d93.pdf> (2019-05-10)
16. Jelmer de Vries: Object Recognition: A Shape-Based Approach using Artificial Neural Networks: <http://www.ai.rug.nl/~mwiering/ObjectRecognition.pdf> (2019-10-10)
17. How to Implement Artificial Intelligence for Solving Image Processing Tasks: <https://www.apriorit.com/dev-blog/599-ai-for-image-processing> (2019-06-03)
18. Paramvir Singh (2018): AI capabilities in Image Recognition: <https://towardsdatascience.com/ai-capabilities-in-image-recognition-7d79aec5222f> (2019-11-10)
19. Rodrigo Verschae, Javier Ruiz-del-Solar(2015): Object Detection: Current and Future Directions: [https://www.researchgate.net/publication/284167004\\_Object\\_Detection\\_Current\\_and\\_Future\\_Directions](https://www.researchgate.net/publication/284167004_Object_Detection_Current_and_Future_Directions) (2019-10-15)

20. Sicara.ai (2019): Edge Detection in Opencv 4.0, A 15 Minutes Tutorial: <https://www.sicara.ai/blog/2019-03-12-edge-detection-in-opencv> (2019-12-30)
21. Shinsuke Shimojo, Michael Paradiso, and Ichiro Fujita (2001): What visual perception tells us about mind and brain: <https://www.pnas.org/content/98/22/12340> (2020-04-20)
22. Pulkit Sharma (2019): Computer Vision Tutorial: A Step-by-Step Introduction to Image Segmentation Techniques: <https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/> (2019-11-10)
23. Shubham Gupta (2018): Understanding image recognition and its uses: <https://www.einfochips.com/blog/understanding-image-recognition-and-its-uses/> (2019-11-10)
24. Oleksii Kharkovyna (2019): Facial Recognition And AI: Latest Developments And Future Directions: <https://becominghuman.ai/facial-recognition-and-ai-latest-developments-and-future-directions-39d22201d88b> (2019-10-05)
25. OpenCV: <https://opencv.org/about/> (2019-04-15)
26. OpenCV tutorial (2019): [https://docs.opencv.org/master/d9/df8/tutorial\\_root.html](https://docs.opencv.org/master/d9/df8/tutorial_root.html) (2019-12-15)
27. OpenCV Tutorialspoint (2019): <https://www.tutorialspoint.com/opencv/index.htm> (2019-11-30)
28. James Wilson (2019): *Machine learning & license plate recognition: An ideal partnership*: <https://bigdata-madesimple.com/machine-learning-license-plate-recognition-an-ideal-partnership/> (2019-12-09)

## Popis slika

Slika 1. Faze ANPR sustava (Jain, Kundargi, 2015).....	18
Slika 2. Osobni automobil.....	23
Slika 3. Gradski autobus.....	24
Slika 4. OpenCV kod za procesiranje slike.....	24
Slika 5. Grayscale osobnog vozila.....	25
Slika 6. Grayscale gradski autobus.....	25
Slika 7. Osobni automobil – korak otkrivanja rubova.....	27
Slika 8. Gradski autobus – korak otkrivanja rubova.....	27
Slika 9. Osobni automobil – korak Adaptive threshold.....	28
Slika 10. Gradski autobus – korak Adaptive threshold.....	29
Slika 11. Osobni automobil – sužavanje područja na moguće registracijske oznake.....	29
Slika 12. Gradski autobus - sužavanje područja na moguće registracijske oznake .....	30
Slika 13. Osobni automobil – izdvojena područja.....	30
Slika 14. Gradski autobus – izdvojena područja.....	31
Slika 15. Kod za izračun centra potencijalne registracijske oznake.....	31
Slika 16. Kod za izračun visine i širine potencijalne registracijske oznake.....	32
Slika 17. Osobni automobil – područje oznake.....	32
Slika 18. Gradski autobus – područje oznake.....	33
Slika 19. Slučaj 1 – osobni automobil.....	33
Slika 20. Slučaj 2 – osobni automobil.....	34
Slika 21. Slučaj 3 – osobni automobil.....	34



Slika 22. Slučaj 1 – gradski autobus.....	34
Slika 23. Slučaj 2 – gradski autobus.....	34
Slika 24. Slučaj 3 – gradski autobus.....	34
Slika 25. Slučaj 4 – gradski autobus.....	34
Slika 26. Slučaj 5 – gradski autobus.....	34
Slika 27. Slučaj 6 – gradski autobus.....	34
Slika 28. Slučaj 7 – gradski autobus.....	34
Slika 29. Osobni automobil.....	35
Slika 30. Gradski autobus.....	35
Slika 31. Segmentacija – osobni automobil.....	36
Slika 32. Segmentacija – gradski autobus.....	36
Slika 33. Kod za segmentaciju.....	37
Slika 34. Osobni automobil – rezultat.....	38
Slika 35. Gradski autobus – rezultat.....	38
Slika 36. Ispravna detekcija – primjer samo sa tablicom.....	39
Slika 37. Ispravna detekcija – personalizirana oznaka.....	39
Slika 38. Djelomična detekcija – primjer 1.....	40
Slika 39. Djelomična detekcija – primjer 2.....	40
Slika 40. Djelomična detekcija – primjer 3.....	41

## Sažetak

Pojam umjetne inteligencije je danas prisutan posvuda. Njeno korištenje uvelike olakšava pojedine procese za koje je prije bilo potrebno mnogo vremena i rada. Jedan od tih procesa je i prepoznavanje registracijskih oznaka. Postoje razni načini kako napraviti dobar sustav za detekciju i prepoznavanje. Najčešće se koriste kombinacije računalnog vida sa strojnim učenjem, neuronskim mrežama i dubokim učenjem. Također se koristi i pristup otvorenog koda, točnije kada je riječ o prepoznavanju registracijskih oznaka, popularno je korištenje OpenCV-a. Cilj ovog rada je objasniti poveznicu između metoda i dati primjer prepoznavanja registracijske oznake koristeći OpenCV.

**Ključne riječi:** umjetna inteligencija, OpenCV, registracijska oznaka, računalni vid, detekcija vozila, prepoznavanje registracijske oznake, strojno učenje, neuronske mreže, biblioteka otvorenog koda

## Summary

The concept of artificial intelligence is present in everyday life. Its use greatly facilitates certain processes that have previously taken a long time to complete. One of these processes is the recognition of registration plates. There are various ways to build a good detection and recognition system. Computer vision combined with machine learning, neural networks or deep learning is the one most commonly used. Open source coding is also used, more specifically when it comes to identifying registration tags, the use of OpenCV is very popular. The aim of this work is to explain the link between methods and provide an example of registration plate recognition using OpenCV.

**Keywords:** artificial intelligence, OpenCV, registration plate, computer vision, vehicle detection, registration plate detection, machine learning, neural networks, open code library