

Koncept realizacije sistema pametnog parkinga zasnovanog na mašinskom učenju

Tatjana Šestić¹, Kovač Miloš², Tomović Nemanja³, Miloš Arbanas⁴, Vladimir Rajs⁵, Milica Kisić⁶

Depatman za energetiku, elektroniku i telekomunikacije

Fakultet tehničkih nauka Novi Sad

Novi Sad, Srbija

tatjanasestic@gmail.com¹, kovac.pm1.2021@uns.ac.rs², haccerra@aol.com³, milos.arbanas@uns.ac.rs⁴, vladimir@uns.ac.rs⁵, mkisic@uns.ac.rs⁶

Sažetak—Sa razvojem gradova sve veći problem predstavlja pronalazak slobodnog mesta za parkiranje vozila mesto. Zbog toga se u sklopu novoizgrađenih stambenih objekata sve češće nudi mogućnost otkupa parking mesta. Kako bi se investicija zaštitila, privatni parking ima ograničen i rezervisan broj vozila. U ovom radu je opisan sistem koji obradom slike obezbeđuje prepoznavanje vozila kojima je odobren parking. Sistem je zasnovan na konceptu mašinskog učenja, a za realizaciju se koristi mikroracunar RPi4 i kamera i IoT platforma.

Ključne riječi—mašinsko učenje, mikroracunarski sistem, Raspberry Pi, Kamera, obrada slike

I. UVOD

Zbog velike brzine razvoja gradova naglo se povećava broj automobila, te je sve veći problem pronaći slobodno parking mesto. Zbog toga se u sklopu novoizgrađenih stambenih objekata sve češće nudi mogućnost otkupa parking mesta. Kako bi se investicija zaštitila, često se pravi sistem sa privatnim parkigom sa ograničenim brojem vozila koje imaju posebnu dozvolu za ulazak.

Većini korisnika parking mesta je bitno da poseduju sistem pomoću kojeg će na što jednostavniji način izvršavati svakodnevnu rutinu, parkiranje automobila. Korisnici sad uglavnom preferiraju beskontaktno otključavanje i pokretanje vozila, s toga je jasno da bi njihove želje bile da ne moraju mahati sa ulaznom karticom svaki put kad pristupaju parkingu. Takođe, ako zaborave karticu, ne mogu da pristupe sistemu i ostaju zaključani sa jedne ili druge strane parkinga.

Za korisnike privatnog parkinga bio bi idelan sistem koji omogućava raspoznavanje registracionih oznaka na osnovu unetog podatka u internu bazu podatka u vidu slike registracione tablice. Sistem bi nakon dobijanja slike proverio da li vozilo koje pristupa privatnom parkingu ima odobrenje.

Danas, pametni parkinzi koji koriste kamere za raspoznavanje registracionih oznaka su još uvek retki. Glavni razlog je veća cena, koja nije isključivo posledica činjenice da je skuplje napraviti softver za očitavanje registracione oznake sa fotografije, već i neophodnost za postojanje alternativne

metode koja omogućava korisniku da pristupi parkingu ukoliko tablice nisu dobro očitane.

Međutim jedna od glavnih prednosti sistem sa kamerom je što u svakom trenutku je moguće trajno zabeležiti sliku vozila koje je pristupilo parkingu. Ulaznu karticu je moguće dati prijatelju koji dolazi u posetu, ali zamena registracionih tablica je poprilično nezgodan proces, a ujedno i ostavlja trag do pronalaska stanara koji je narušio privatnost parkinga.

II. STRUKTURA SISTEMA I OSNOVNE KARAKTERISTIKE HARDVERSKIH KOMPONENTI

IoT (*eng. Internet of things*) sistemi se najčešće primenjuju u povezivanju embeded sistema sa bežičnim senzorskim mrežama, kontrolnim i automatizovanim sistemima.

A. Iot platforma

IoT predstavlja mrežu fizičkih objekata (stvari) čija interakcija se ostvaruje međusobnim povezivanjem putem interneta, a njihova isključiva namena je razmena informacija sa senzora. [1]-[3]. IoT sistemi se najčešće primenjuju u povezivanju embeded sistema sa bežičnim senzorskim mrežama, kontrolnim i automatizovanim sistemima.

Za realizaciju ovog projekta ostvarena je komunikacija sa platformom Wolkabout [4] na kojoj se čuvaju podaci. Ova platforma je pogodna kao deo sistema za nadzor jer omogućava da se na jednostavan način izvrši sprega sa sistemom za kontrolu ulaza. Takođe, implementirani su i sigurnosni protokoli kako bi se sprečili pristupi privatnim podacima.

B. Raspberry Pi (model 4B) sa kamerom.

U ovom projektu korišten je savremeni model 4B mikroracunarskog sistema Raspberry Pi sa pratećim softverom [5]- [7].

Važno je napomenuti da se na bilo kojoj verziji mikroracunarskog sistema ovog proizvođača može iskoristiti izrađena aplikacija u neizmenjenom obliku, ali performanse rada će u najvećoj meri zavistiti od količine RAM memorije i procesorskog takta korištenog uređaja.

Iako je moguće povezati bilo kakvu kameru na RPi (npr kamere sa USB konektorom), za realizaciju projekta korištena je kamera koja je isključivo razvijena za RPi za koji postoji specifično namenjen konektor na samom uređaju. Kod kamere, najčešće je najbitnija informacija o rezoluciji samog senzora. Kod kamere verzije 1, rezolucija senzora je 5 Mpix, dok je sama slika po osnovi veličine 2592 x 1944 piksela. Posledica velike rezolucije je nepotreban utrošak memorije [8]. Zbog toga, u aplikaciji sve slike se obrađuju u rezoluciji od 640 x 480 piksela.

C. KY038 senzor zvuka

KY038 [9] je senzor sa ugrađenim mikrofonom. Karakteriše ga dva tipa izlaza. Jedan je analogni i na svom izlazu daje direktan odziv sa mikrofona, dok je drugi digitalan i koristi komparator signala na čijim ulazima se poredi signal sa mikrofona i podešena vrednost na potencijometru. Što je niža otpornost na potencijometru, to je potrebna veća snaga zvuka kako bi se odrazilo na izlaz. Digitalni izlaz je uvek na visokom naponskom nivou. Detekcija niskog nivoa signala označava da je senzor detektovao zvuk željene snage.

Pri razvoju aplikacije sa ovim senzorom, ključne su dve diode: L1 označava da senzor ima odgovarajuće napajanje, dok dioda L2 predstavlja izlaz sa komparatora, odnosno, kada sija to je znak da je senzor odreagovao na određeni zvuk.

III. SOFTVERSKA STRUKTURA I ALGORITAM RADA

Program je realizovan korišćenjem dva programska jezika: *python* i *shell*. Izvršni algoritam programa smešten je u *python* skriptu. Razlog za postojanje *shell* skripti je uobičajen – potrebno je na što jednostavniji mogući način nakon preuzimanja programa omogućiti njegov rad. Ceo proces je automatizovan što olakšava običnom korisniku rad sa aplikacijom.

Potrebno je naglasiti da se program pokreće preko *omnirun.sh* skripte. Racionalnost ovakvog načina rada leži u tome da je moguće pozvati različite modove programa, koji ne moraju biti isključivo vezani za njegov rad.

U okviru projekta korišćene su sledeće biblioteke za RPi:

- *raspistill* – RPi biblioteka za upravljanje sa kamerom;
- *raspivid* – RPi biblioteka za snimanje videa korišćenjem kamere;
- *openalpr* – biblioteka za pronalaženje registracionih oznaka na unetoj slici korišćenjem takozvane OCR metode (eng. *optical character recognition*);
- *PiCamera* – biblioteka za upravljanje sa kamerom korišćenjem *python* programskog jezika;
- *opencv* – biblioteka za rad sa informacijama koje predstavljaju slike;
- *wolk* – biblioteka koja omogućava komunikaciju mikroracunarskog sistema sa udaljenom platformom

A. OpenALPR biblioteka za detekciju registracionih oznaka

OpenALPR je biblioteka sa kodom koji ima otvoren pristup. Koristi se za automatizovano raspoznavanje registracionih oznaka. Biblioteka može da analizira slike ili video podatke sa kojih može očitati registracione tablice. Napisana je u C++ jeziku, ali poseduje i okruženje za C#, Java, Node.js i Python programske jezike. Rezultat rada biblioteke je podatak u vidu teksta sa očitanim karakteristikama na registracionoj tablici. [10]. Rezultati se mogu pretražiti na veoma jednostavan način. Baza podataka se može smestiti i na IoT platformu.

OpenALPR se za pravilan rad oslanja na biblioteku *tesseract-ocr*, koju razvija kompanija Google i podešava se za određena geografska područja. To znači da automobil sa arapskim tablicama, gde tablice sadrže karaktere nepoznate za naš geografski region, neće biti u mogućnosti da se protumače. Drugim rečima, ako se podesi za područje Srbije, s obzirom na namenu (detekcija karaktera na registarskim tablicama), biblioteka se obučava za prepoznavanje karaktera iz srpskog jezika sa fontom koji se koristi na srpskim tablicama. U okviru biblioteke postoje kontrolni testovi za trening.

B. OCR metoda

OCR (eng. *Optical Character Recognition*) je elektronski vid identifikacije i digitalnog enkodiranja slike na kojoj se nalazi tekst. Korišćenjem ove metode, omogućava se da kompjuter izvrši konverziju slike u tekstualni format. OCR [11] tipično sadrži tri faze:

- otvaranje i skeniranje dokumenta
- prepoznavanje dokumenta u OCR softveru i
- čuvanje podataka dobijenih analizom dokumenta.

Slika uneta u OCR softver se prvo analizira pretragom svetlih i tamnih područja na slici. Tamna područja na slici se smatraju tekstem koji je potrebno obraditi, dok je svetli deo slike pozadina na kojoj su karakteri odštampani (bez obzira na boju pozadine) [12]. Sa slike se potom izvlače tamni segmenti i puštaju na analizu. U okviru analize traga se za slovima, brojevima ili specijalnim znakovima. Iako tehnike iznalaženja teksta sa slike mogu da variraju, najveći broj OCR softvera pretragu vrši karakter po karakter. Pronađeni karakteri se obrađuju korišćenjem posebnog algoritma: Važno je napomenuti da biblioteke za OCR koje se tipično koriste u aplikacijama rade korišćenjem *Pattern recognition* metode. Ova metoda je pouzdanija i oslanja se na mašinsko učenje. Napravljena neuronska mreža se obučava na leksičke karakteristike nekog jezika.

IV. MODOVI RADA PROGRAMA

Nakon pokretanja glavnog programa pojavljuje se meni sa sledeći modovima rada programa:

- *install mod* - preuzimaju se sve instalacije i biblioteke potrebne da bi program mogao da radi;

- *setup* – jednostavan mod kojim se omogućava podešavanje *alijasa lpltdt* što olakšava upotrebu programa (ukoliko se iz nekog razloga često poziva program). Da bi *alijas* radio, neophodno je program pokretati iz *root* nivoa aplikacije;

- *start* – glavni mod programa kojim se otpočinje automatizovan rad bez nadzora i uticaja korisnika;

- *editor* – grafički interfejs pomoću kojeg se upisuju, brišu i dodaju stanari u bazu podataka;

- *demo* – prezentacioni mod rada programa (može se koristiti i kao kontrolni mod), gde je automatizacija isključena. Tok programa je direktno određen komandama koje stižu od korisnika. U ovom modu neke od funkcionalnosti, poput konekcije sa udaljenim serverom, nisu prisutne.

A. Start mod

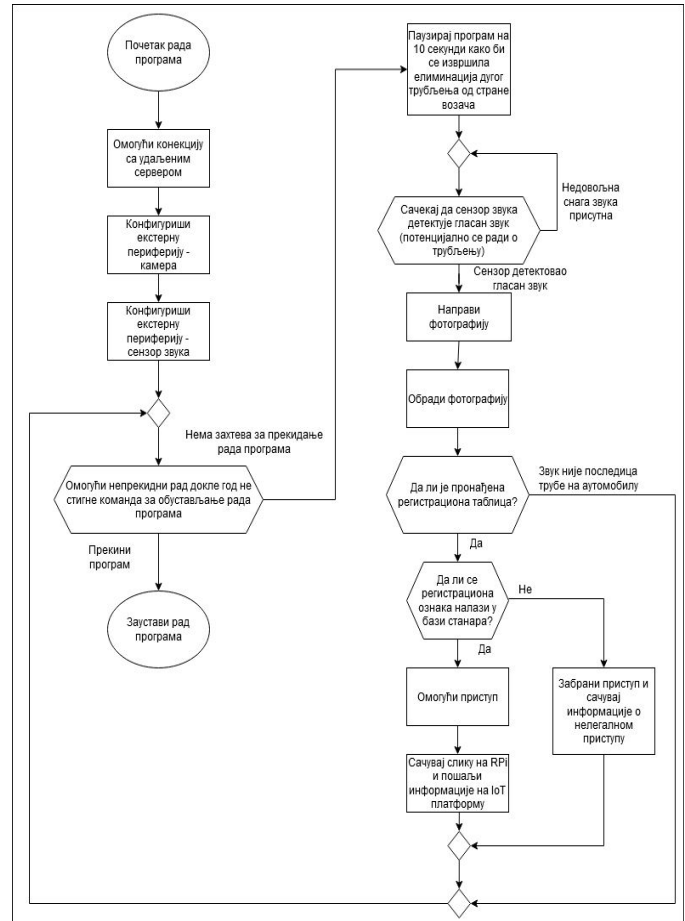
Algoritam rada programa u automatizovanom modu, dat je na SI 1. Kada stanar pristupi parkingu, potrebno je da zatrubi. Ovo je signal za naš sistem da treba da kamerom fotografiše i proveri da li postoje tablice na fotografiji. Neophodno je znati da li su pronađene registracione tablice na slici, zbog činjenice da u okolini rada sistema može odjeknuti veoma glasan zvuk, zbog kog bi naš sistem mislio da stanar traži pristup. Ukoliko jeste neočekivani zvuk koji je pokrenuo rad sistema, potrebno je ignorisati fotografiju i sistem će se ponašati kao da se ništa nije desilo. Sistem je zaštićen na taj način što svako trubljenje kraće od 5 sekundi neće pokrenuti novu sekvencu za registraciju pristupa istog vozila. Pretpostavka je da nijedan stanar neće trubiti duže od ovog vremena zbog neprijatnosti po okolinu koji zvuk trube izaziva. Ipak, vreme od 5 sekundi je minimalno vreme zaštite koje može trajati nešto duže (zavisno od potrebnog vremena za obradu slike i izvršavanje celog algoritma). Motiv za aktivaciju ovog sistema je isključivo originalnost. Ceo sistem je modularan te se bez problema može dodati arhitektura bazirana na PIR senzoru kao detektoru prisustva automobila ispred rampe. Ovo bi rešilo i problem nefuncionalnosti sistema u slučaju kvara zvučnog senzora.

Ukoliko je, kao posledica detekcije zvuka velike snage, pronađena registraciona tablica, tada je sistem dužan da proveri da li pronađena registraciona oznaka postoji u bazi podataka.

Ukoliko postoji, stanaru će biti odobren pristup parkingu. U suprotnom, sistem će ove tablice zabeležiti kao nedozvoljeni pristup. Stoga, ukoliko vozilo koje ne želi da pristupi parkingu, a ni nema dozvolu za njega, ne zatrubi, nikada neće biti zabeleženo kao vozilo koje je pokušalo da uđe u garažu za koju nema dozvolu.

Sve fotografije o pristupu vozila se čuvaju na mikroracunaru RPi. Unutar direktorijuma *savedata* se pri prvom pristupu stanara pravi jedinstveni direktorijum u koji će biti smeštene sve fotografije o dolasku/odlasku iz garaže. Direktorijum se formira na osnovu *LamelaBROJ_ImePrezime* kako bi bio jedinstven. Unutar su fotografije dodatno razvrstane po direktorijumima zavisno od datuma, dok su same slike memorisane na osnovu vremena kad je fotografija uslikana. Važno je napomenuti da u slučaju da se stanar obriše iz baze podataka, svi prethodni unosi (slike) se neće automatski obrisati sa mikroracunara, ali će vozilo tog stanara preći u

nelegalnu kategoriju ukoliko nakon što se odseli pokuša da pristupi svom nekadašnjem parkingu. Sve fotografije o pristupu vozila kojima odobrenje nije dozvoljeno se čuvaju u direktorijumu *savedata/illegal*. Ove fotografije su memorisane brojevima (iterativno se generiše novi broj sa svakim novim nelegalnim pokušajem pristupa parkingu). Resetovanjem rada sistema se resetuje i iterativni brojač što za posledicu ima prebrisavanje preko prethodno sačuvanih slika.



Slika 1 Algoritam rada automatizovanog programa

Tekstualne informacije se šalju na IoT platformu i one sadrže informacije o stanaru (ime, prezime, broj stana) i vozilu stanara (model, marka i registracione tablice) uz vreme kad je pristup odobren. U slučaju da je nelegalni pristup, tada se informacije formatiraju o trenutku nelegalnog pristupa, registracionim oznakama i pod kojim brojem je fotografija sačuvana.

Najbitnije je za napomenuti da se baza podataka sa stanarima može paralelno menjati bez zaustavljanja rada programa. Zaustavljanje rada programa nije predviđeno ni poželjno. Program se zaustavlja komandom CTRL+C.

B. Demo mod

Pokretanje ovog moda i ispis sa svim opcijama dat je na SI 2.

```

pi@raspberrypi:~/cod3s/lpltdt $ lpltdt --demo
Input the demo command. To see more information enter help: help
*****
Enter one of the following commands:
  captimg :: take an image from camera.
  captvid :: take series of images from camera.
  process  :: try to find license plates on an image.
  imgshow  :: preview image taken by the camera.
  imgtst  :: path to test image (does not take an image from camera but from memory drive).
  demoesc  :: abort the program execution.
*****
Input the demo command. To see more information enter help:

```

Slika 2 Prikaz svih opcija u demo modu

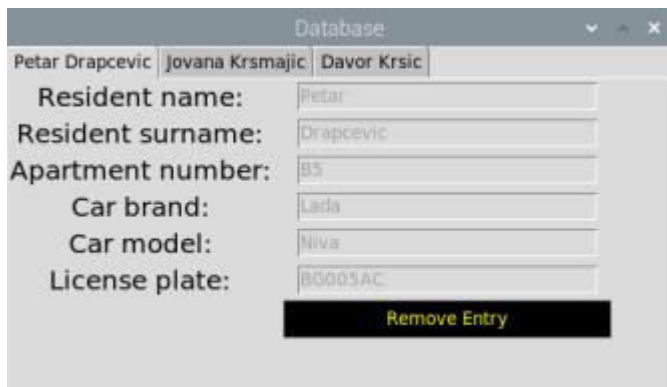
U okviru ovog moda moguće su sledeće akcije:

- *captimg* – fotografiše sliku u traženom momentu. Samo jedna slika se čuva u svakom trenutku;
- *captvid* – snimi video (5 slika u sekundi);
- *process* – procesira poslednju fotografisanu fotografiju;
- *imgshow* – prikazuje poslednju fotografisanu fotografiju.
- *imgtst* – u memoriju učitava fotografiju sa memorijske kartice (umesto preuzimanja fotografije sa kamere);
- *demoesc* – prekida rad programa (alternativno može se koristiti CTRL+C opcija)

C. Demonstracija rada programa

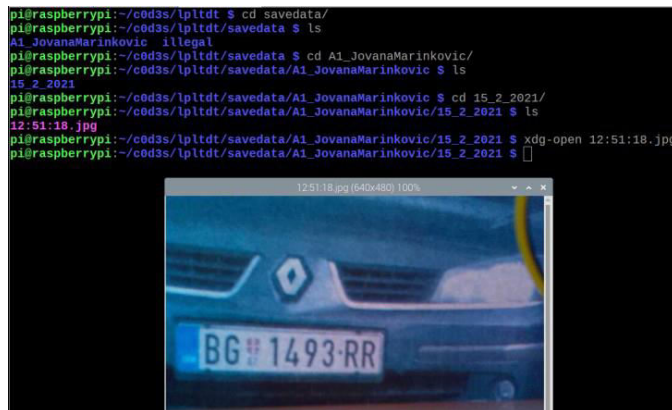
Pri prvom pokretanju programa potrebno je navigirati se u željeni folder. Unošenjem komande `lpltdt --start` program će otpočeti sa normalnim radom.

Na samom početku potrebno je definisati bazu podataka. Potrebno je otvoriti novi proces u terminalu i uneti `lpltdt --editor` komandu. Na Sl. 3 prikazane su informacije za unete stanare.



Slika 3 Prikaz dela informacija iz baze podataka za stanare sa odobrenim pristupom

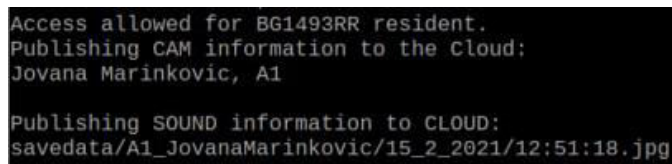
Nije potrebno prekidati rad programa kako bi se izvršila bilo kakva izmena na bazi podataka, već je dovoljno program pokrenuti kroz drugi nezavisni proces. Informacije o stanarima iz baze podataka se prate u realnom vremenu.



Slika 4 Prikaz uslikane tablice na vozilu koje pristupa parkingu

Dolaskom vozila sa registracionim oznakama BG1493RR (Sl.4) na privatni parking, vozač zadrubi kako bi signalizirao sistemu da želi da pristupi parkingu.

Na slici 4 prikazano je da je vozilo sa ovim registracionim oznakama dobilo odobrenje za pristup. Informacije o stanaru su takođe prikazane i šalju se na platformu.



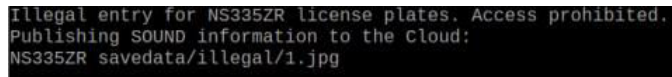
Slika 5 Prikaz odobrenja pristupa parkingu za stanare

Informacije koje se šalju na udaljenu platformu su podeljene na informacije sa kamere i na informacije sa senzora zvuka.

Informacije sa kamere šalju osnovne informacije o stanaru koji je pristupio parkingu (ime, prezime i stan). Informacije o lokaciji sačuvane slike se šalju uz informacije o senzoru zvuka.

Bitno je za napomenuti da se slike isključivo čuvaju na uređaju kako bi se povećala privatnost i smanjila upotreba *cloud-a*.

U slučaju nedozvoljenog pristupa, jedine informacije koje se šalju na platformu se šalju preko senzora zvuka jer ne postoje informacije u bazi podataka za vozača sa prikazanim oznakama. Fotografija pribavljena sa kamere je prikazana na Sl. 4. Sledeći prikaz dat je na Sl. 6 je ponašanje programa u slučaju da vozilo bez odobrenja pristupa parkingu.



Slika 6 Prikaz informacija u slučaju nedozvoljenog pristupa

Za tačnost sistema može se reći da je oko 100% ukoliko kamera jasno uhvati tablicu sa jasnim oznakama registracije. Međutim pouzdanost sistema zna da bude problem usled loše postavke automobila ili prljavih tablica. Prilikom testiranja prototipa vođeno je računa da vozila moraju biti pozicionirana

na određenom mestu sa priličnom jasnim oznakama na tablicama. Testovi su rađeni na desetak vozila.

Zaključak

Ovaj projekat pokazuje mogućnost povezivanja jednog embeded sistema sa udaljenim serverom (cloud) povezujući mikroračunar RPi zajedno sa Wolkabout IoT platformom. Mikroračunar se koristi kako bi povezao odgovarajuće senzore obrađujući njihove ulazne podatke, a potom iste šalje na udaljenu platformu. U okviru udaljene IoT platforme čuvaju se podaci ovog sistema sa ciljem mogućnosti udaljenog nadzora rada platforme.

Baza podataka sa stanarima je locirana na samom uređaju zbog čega je neophodno fizički pristupiti RPi sistemu kako bi se ova baza podataka uredila. Pozitivna stvar je što je program u stanju da radi neprekidno uprkos ažuriranju baze podataka.

To znači da sistem nije nefunkcionalan u slučaju prekida internet veze. Sistem će i dalje prepoznavati registracione tablice i omogućiti pristup parkingu samo neće slati obaveštenja na platformu sve dok ne bude uspostavljena internet konekcija.

Ako izuzmemo obaveznu alternativu za pristup usled potencijalnih kvarova ključnih komponenti osnovno ograničenje ovog sistema u pogledu daljeg razvoja sa većim brojem korisnika jest ograničena memorija sistema koji još treba unaprediti sa dodatnim podacima. S toga zaključujemo da je neophodno proširenje memorijskog dela.

Predlozi za dalje unapređenje ovog sistema su:

Trenutna verzija programa zahteva apsolutno poklapanje pronađenih registracionih oznaka sa slike sa onima unesenim u bazi podataka. Razlog za to leži u činjenici da ne postoji način da stanar da povratnu informaciju da se radi baš o njegovom vozilu, u slučaju parcijalnog pronalaska tablice.

Moguće je na osnovu slike pronaći marku i model vozila. Baza podataka već sadrži polja sa informacijama o kom proizvođaču i modelu automobila je reč. Pri tome već postoji kod za klasifikaciju vozila u okviru ove verzije programa, ali se ne upotrebljava u ovoj verziji. Za unapređenje ove tačke, potrebno je u instalacionu skriptu ubaciti instalaciju za *vehicleclassifier* biblioteku, a potom modifikovati kod u zavisnosti od potrebe. Glavno unapređenje, se ogleda u činjenici da ukoliko je program našao parcijalno poklapanje tablica, stanaru neće stići upit da li se radi o njegovom vozilu ukoliko iznađeni proizvođač i/ili model vozila ne odgovaraju onome što je upisano u bazi podataka. Na ovaj način se može sprečiti neželjeno slanje upitnika ukoliko neki nasumični vozač sa sličnim registracionim oznakama se nađe ispred privatnog parkinga stanara

ZAHVALNICA

Ovo istraživanje je podržano od strane Ministarstva prosvete, nauke I tehnološkog razvoja kroz projekat broj 451-03-68/2022-14/ 200156 "Inovativna naučna i umetnička istraživanja iz domena delatnosti FTN-a"

LITERATURA

- [1] Alexander S. Gillis, <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> – 1.12.2021
- [2] Hardik Shah, Enable IoT technology security with IoT software engineering <https://internetofthingsagenda.techtarget.com/post/Enable-IoT-technology-security-with-IoT-software-engineering> 1.12.2021.
- [3] Rahul Guhathakurta, <https://trak.in/tags/business/2016/08/30/internet-of-things-iot-changing-fundamentals-of-retailing/> – 1.12.2021.
- [4] Uputstvo za rad sa platformom Wolkabout <https://wolkabout.com/user-guides/>
- [5] Uputstvo za rad sa Raspberry PI Rpi4 dostupno na adresi : <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/> – 2.12.2021.
- [6] Uputstvo za rad sa Raspberry PI Rpi4 dostupno na adresi <https://www.raspberrypi.org/about/> – 2.12.2021.
- [7] Uputstvo za rad sa Raspberry PI Rpi4 dostupno na adresi <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/> – 2.12.2021.
- [8] Uputstvo za rad sa Raspberry PI kamerom dostupno na adresi <https://www.tdgulf.com/product/raspberry-pi-camera-v1-3-5mp-1080p/> – 2.12.2021.
- [9] Uputstvo za upotrebu senzora KY038 <https://datasheetspdf.com/pdf-file/1402048/Joy-IT/KY-038/1> – 2.2.2021.
- [10] Uputstvo za rad sa bibliotekom *Openalpr* <http://doc.openalpr.com/> – 5.12.2021.
- [11] Uputstvo za rad sa OCR softverom <https://guides.library.illinois.edu/OCR> – 5.2.2021.
- [12] Uputstvo za rad sa OCR softverom <https://searchcontentmanagement.techtarget.com/definition/OCR-optical-character-recognition> – 5.12.2021

ABSTRACT

With the development of cities, finding a free space for parking vehicles is a growing problem. Due to that, the possibility of buying a parking space is increasingly offered within the newly built residential buildings. In order to protect the investment, private parking has a limited and reserved number of vehicles. This paper describes a system that uses image processing to identify vehicles that have been approved for parking. The system is based on the concept of machine learning, and the RPi4 microcomputer and cameras and IoT platform are used for the realization.

THE CONCEPT OF REALIZATION OF A SMART PARKING SYSTEM BASED ON MACHINE LEARNING

Tatjana Šestić¹, Kovač Miloš², Tomović Nemanja³, Miloš Arbanas⁴, Vladimir Rajs⁵, Milica Kisić⁶