



УНИВЕРЗИТЕТ У НИШУ
ЕЛЕКТРОНСКИ ФАКУЛТЕТ



PARKING FINDER

Извештај

Студијски програм:
Рачунарство и
информатика

Модул: Софтверско
инжењерство

Предмет: Информационе
технологије за развој
система е-Управе

Студент:

Ивана Петковић, бр. инд. 1787

Ментор:

Проф. др Леонид Стоименов

Ниш, 2024. година

Садржај

1. Увод.....	3
2. Идеја.....	4
3. Е-Управа	5
4. Истраживање и анализа постојећих решења	7
5. Инфраструктура и начини прикупљања података	12
6. Истраживање технологија	15
7. Реализација апликације	17
7.1. Подразумевана инфраструктура.....	17
7.2. Архитектура апликације.....	17
7.2.1. Google Maps Platform	18
7.2.2. Flask и .csv	20
7.3. Детекција објеката – пример реализације.....	23
8. Закључак	30
9. Литература.....	31

1. Увод

У оквиру овог рада бавићемо се проблемом тражења паркинг места на јавним паркинзима. Биће представљено решење у контексту **е-Управе** под називом **„Parking Finder“**. Циљ пројекта јесте да омогући проналажење слободног паркинг места на јавним паркинзима и тиме олакша људима овај јако познати проблем данашњице.

Паркирање је свакодневни изазов за многе возаче, посебно у урбаним срединама где је потражња за паркинг местима већа од понуде. Овај пројекат нуди иновативно решење за поменути проблем, користећи најновије технологије и концепте дигиталне управе, како би побољшао квалитет живота људи.

На почетку ћемо се упознати са самом идејом пројекта и како је она повезана са концептом е-Управе. Након тога ћемо имати преглед постојећих решења, као и анализу и увид у то која је инфраструктура потребна и уз помоћ којих технологија се слична решења реализују. Ово истраживање је од великог значаја ради имплементације конкретног пројекта „Parking Finder“. Напослетку, видећемо којим технологијама и уз помоћ које инфраструктуре је реализовано решење „Parking Finder“ што представља централни део рада.

2. Идеја

Пројекат носи назив „**Parking Finder**“. Главна идеја око које је осмишљен, како се може наслутити и из самог назива, јесте проналажење слободног паркинг места на јавним паркинзима. Циљ овог пројекта је да на једном месту пружи информације у реалном времену о слободним и заузетим паркинг местима на јавним паркинзима. Осим тога треба да пружи и услуге навигације до жељеног паркинг места.

Проблем је, дакле, проналажење слободних паркинг места за аутомобиле, нарочито у урбаним срединама. Погађа директно возаче аутомобила којима је потребно паркинг место на одређеној локацији. А индиректно погађа и све људе који живе у урбаним срединама, због проблема гужве и загађења. Последице овог проблема које се осећају на личном нивоу сваког појединца су трошење времена, горива и новца на проналажење паркинг места, повећање стреса и фрустрације, компромитовање личних планова, повећање ризика од добијања саобраћајне казне. На глобалном нивоу утиче на стварање гужве у саобраћају, што уз висок ниво стреса повећава ризик од саобраћајних незгода и прекршаја, ствара загушење саобраћаја и отежава промет, а такође и повећава загађење ваздуха због емисије горива. Свим овим факторима заједно негативно утиче на квалитет живота људи.

Успешно решење ће омогућити проналажење најближих паркинга у односу на тражену локацију заједно са информацијама о слободним паркинг местима, цени, удаљености и другим релевантним информацијама. Поред тога, пружиће и навигацију до жељеног паркинг места.

Производ је намењен свим возачима аутомобила, који ће уз помоћ њега на најбржи начин наћи слободно паркинг место свуда где им је слободно паркинг место на јавним паркинзима потребно. Производ може бити веб и/или мобилна апликација која ће брзо и ефикасно извршавати предвиђене акције.

За разлику од других, већ постојећих, овај производ ће пружити све важне информације о паркинзима и паркинг местима у једној апликацији(на једним „месту“) и навигацију до слободних места.

3. Е-Управа

Нове комуникационе и информационе технологије, као средство на путу ка информатичком друштву и економији заснованом на знању, намећу потребу за модернизацијом јавне управе и стварањем савремене електронске управе [1]. Појам **електронска управа (е-управа)** односи се на употребу информационих и комуникационих технологија у размени информација, пружању сервиса и пословању државних органа и институција са физичким и правним лицима [1]. Државне управе препознале су многобројне вредности увођења информационих и комуникационих технологија у све сегменте рада управе као и у сегменте односа управе са грађанима и привредом [1]. Земље света препознају потребу за развијањем е-управе као једног од чинилаца унапређивања својих економија и повећања улоге фактора знања у развоју економије [1].

Постоје различите дефиниције појма е-управа гледано из социолошког, технолошког, правног или економског аспекта [1]. Корисно је сагледати концепт е-управе из поменутих аспеката. Следе неке од дефиниција које се срећу у стручној литератури:

- е-управа означава употребу Интернета и „WWW“-а ради пружања информација и услуга грађанима [1].
- е-управа означава коришћење информационих и комуникационих технологија у јавној администрацији, у организационим процесима и процесима развоја нових вештина у администрацији са циљем побољшања демократских процеса, управљања и вођења политике [1].
- е-управа се односи на коришћење информационих технологија (укључујући широкопојасне мреже, Интернет, мобилно рачунарство) које имају моћ да промене односе са грађанима, привредом и међу институцијама јавне управе [1].

Дакле, е-управа је систем електронске управе који омогућава грађанима да обављају различите административне послове путем интернета. Основни циљ е-Управе је да поједностави, убрза и учини транспарентнијим процесе који иначе захтевају физичко присуство у различитим државним институцијама. Коришћењем е-управе смањује се потреба за физичким одласком у државне канцеларије, што штеди време и ресурсе корисницима. Има огромне користи за целокупно друштво, грађане и привреду, укључујући бржу обраду захтева, смањено време одзива на захтеве корисника, смањење трошкова, премошћавање дигиталног јаза, кројење услуга према потребама корисника и активно учешће грађана у процесима управе[1]. Свиме тим побољшава ефикасност рада државних служби. Е-управа је постала кључни фактор за трансформацију јавног сектора и увођење ефикасног управљања, транспарентности, одговорности и појачано учешће грађана у демократским процесима и креирању политике [1].

Развојни пут е-управе почео је у последњој деценији XX века, када се појавила идеја да се електронско пословање примени на сферу државне управе [1]. Временом је област е-управе постала јако популарна и важна област истраживања. Уопштено посматрано, скоро сваки проблем који се тиче грађана и отворених података повезан је са управом и може се пренети на систем е-управе. Тако се и проблем проналажења слободног паркинг места на јавним паркинзима везује за е-управу.

„Parking Finder“ апликација би могла бити интегрисана у постојеће системе е-управе, омогућавајући корисницима да приступе овој услузи преко званичних портала. Ова интеграција би омогућила и лакше управљање подацима о паркингу, као и бољу комуникацију између различитих државних и приватних субјеката који су укључени у управљање паркинг просторима.

Осим тога, систем би могао пружити статистичке податке и анализе које би помогле градским властима у планирању и управљању инфраструктуром. Ово би укључивало и могућност праћења коришћења паркинг простора, идентификовања најоптерећенијих подручја и предлагање управљања саобраћајем и паркингом.

4. Истраживање и анализа постојећих решења

Постоји више апликација које су већ развијене са циљем решавања проблема паркирања у урбаним срединама али и као помоћ приликом паркирања у виду информисања, резервације, могућности плаћања и слично. У наставку ћемо се осврнути на неке од њих које су међу најпознатијима.

Примери постојећих апликација за паркинг:

- **ParkWhiz:** Ова апликација омогућава корисницима да пронађу најближи паркинг, унапред резервишу паркинг место и добију информације о ценама. Такође омогућава упоређивање цена на различитим паркинзима, па корисник може да изабере паркинг који му највише одговара. Предност је уштеда времена и могућност планирања паркинга, али недостаје подршка за сваки град и праћење слободних паркинг места у реалном времену.

Више о самој апликацији може се видети на њеном званичном сајту - <https://www.parkwhiz.com/>.

- **SpotHero:** Слично као ParkWhiz, SpotHero омогућава унапред резервисање паркинг места. Има широк спектар паркинг гаража и приватних паркинг места, али је мање корисна за свакодневно паркирање на улици. Омогућава проналажење и резервисање паркинг места и на аеродромима. Недостатак ове апликације је ограничена подршка. Подржана је у преко 300 градова у оквиру Сједињених Америчких Држава и Канаде.

Званична веб локација ове апликације је <https://spothero.com/>.

- **Parkopedia:** Служи истој сврси као и до сада поменуте апликације. Дакле, помаже корисницима у проналажењу и резервисању паркинг места. Поред тога пружа детаљне информације о паркинг местима, укључујући цене, радно време, доступне услуге(као што су пуњење за електрична возила) и ограничења. Апликација може пружити навигацију до одабраног паркинг места. Ова апликација пружа информације о више од 60 милиона паркинг места у 75 земаља. Висока покривеност паркинга широм света чини је корисном за шири спектар људи укључујући путнике широм света и оне који често мењају локације. Међутим иако нуди обимне податке, фокусирана је више на статичне информације него на реално-временско праћење што је њен главни недостатак.

Више о њој је доступно на <https://www.parkopedia.com/>.

- **eParking:** У питању је домаћа апликација, односно апликација која се користи на територији Републике Србије, а која омогућава проналажење и плаћање паркинга. За коришћење услуга које она нуди потребна је регистрација и пријава. Апликација омогућава корисницима да пронађу доступне паркинг локације у свом граду или у другим градовима који подржавају овај систем. Корисници могу да плаћају паркинг путем апликације. Апликација шаље обавештења када се приближава истек плаћеног времена за паркинг, што омогућава корисницима да продуже време паркирања ако је потребно.

Корисници могу да прегледају своју историју плаћања и паркинг трансакција у апликацији. Апликација подржава плаћање паркинга у више градова Србије, што је чини погодном за оне који често путују. Недостаци овог система огледају се у томе што не пружа функционалност навигације до жељеног паркинга. Даље, има ограничену покривеност, eParking не обухвата све паркинге широм Србије. Подаци о паркинг местима и услуге плаћања доступни су само у одабраним градовима који су интегрисани у систем апликације. Такође, апликација не укључује информације о бесплатним паркинг местима. Фокусирана је искључиво на плаћене паркинге, што значи да корисници не могу добити информације о локацијама где паркинг није наплаћен.

Иста се може наћи на веб локацији <https://eparking.rs/login>.

На основу извршене анализе можемо укратко истаћи предности и мане ових решења:

Предности постојећих решења укључују:

- Могућност налажења слободног паркинг места.
- Могућност планирања унапред и резервисања паркинг места.
- Интеграција са навигационим системима.
- Могућност плаћања.
- Обимне базе података о паркинг местима широм света.

Мане укључују:

- Недостатак података у реалном времену о слободним паркинг местима.
- Ограничена подршка у различитим градовима и државама.
- Ограничена подршка у мањим или мање развијеним урбаним срединама.
- Високи трошкови резервисања и услуга.

Анализа корисничких искустава и рецензија

Корисничка искуства и рецензије постојећих апликација показују да је највећи изазов недостатак тачних и ажурних података о слободним паркинг местима. Корисници често пријављују да апликације не могу да прикажу тренутну доступност паркинг места или да су подаци о ценама и локацијама нетачни. Такође, постоји и велико ограничење у погледу покривености паркинга. Овај проблем може бити решен употребом напредних технологија за прикупљање и обраду података у реалном времену и покривеност свих типова паркинга као што је предвиђено у пројекту "Parking Finder".

Примери градова који су успешно имплементирали слична решења

Сан Франциско(SFPark)

SFPark је систем Сан Франциска за управљање доступношћу паркинга који је уведен 2011. године. Систем користи паметне паркинг уређаје који мењају цене паркинга у зависности од локације, доба дана и дана у недељи, са циљем да око 15% места остане празно у било ком блоку. Преко сензора постављених у асфалт врши се праћење

доступности паркинг места у реалном времену и подешавање цена паркинга у складу са потражњом. Поред паркинга на улицама укључене су и гараже у власништву града.

Сервис је грађанима доступан као мобилна и као веб апликација.

Резултати су показали да је дошло до смањења времена и горива потребног за проналажење паркинг места, самим тим и смањење саобраћајних гужви и загађења.

Принцип рада овог система базиран је на концепту тржишно заснованих варијабилних цена.

Студија из 2014. године показала је да је SFPark систем испунио свој циљ са 60-80% попуњености и да је кружење због паркинга смањено за 50%.

Систем је почетком 2018. проширен на 28 000 паркинг метара са почетних 7000 метара, након чега је наставио да се шири.

Општинска саобраћајна агенција Сан Франциска (СФМТА) проценила је пројекат SFPark како би видела колико је овај приступ управљању паркирањем донео користи. Евалуација је показала да након коришћења SFPark-а у Сан Франциску[6]:

- просечне цене паркирања су биле ниже;
- доступност паркинга је побољшана;
- лакше је проналажење паркинг места;
- лакше је платити и избећи наплату паркирања;
- емисије гасова стаклене баште су смањене;
- пређени километри возила су се смањили.

Чикаго

Са истим циљем је организован и паркинг систем у Чикагу. У оквиру јавно-приватног споразума са градом Чикагом, **Chicago Parking Meters (CPM)** је одговоран за рад, управљање и одржавање паркирања на улицама Чикага са „паркомати”¹ [7]. Од када је кренуо са радом 2009. године, CPM је модернизовао градску инфраструктуру за паркирање инсталирањем најсавременијих електронских станица за плаћање које прихватају кредитне и дебитне картице, као и увођењем апликације ParkChicago, која омогућава возачима да између осталог и плаћају паркирање путем својих паметних телефона [7]. Поред тога, ЦПМ је успоставио опцију рефундирања, 24-часовни центар за корисничку подршку и најсавременији процес одржавања и поправке [7].

IBM-ова Глобална анкета о паркирању назвала је Чикаго међународним лидером доделивши му признање за најбољи систем паркирања на улици којим се модернизовао застарели систем града, смањило време потребно за проналажење слободних паркинг места и побољшао проток саобраћаја [7].

¹ паркомати (parking meters) – уређај који се користи за прикупљање новца у замену за право паркирања возила на одређеном месту на ограничено време.

Возачи могу да плате паркинг преко својих паметних телефона од 2014. године када је СРМ лансирао **ParkChicago** апликацију. Њоме су олакшане и додатно проширене могућности дотадашњег система за паркирање. Поред лаког и практичног начина за плаћање паркинга, омогућено је и обавештавање о истеку времена за паркинг и продужење истог са удаљене локације.

Што се тиче надлежности, град Чикаго задржава контролу над стопама паркинга на платним станицама. СРМ не поставља и никада није одређивао стопе паркирања на платним станицама. Град задржава искључиву надлежност да одређује и утврђује тарифе, одређује радно време и поставља, додаје или уклања просторе са мерењем [7]. Почетни петогодишњи распоред стопа, који је завршен 2013. године, одобрило је Градско веће како би се ускладио са стопама упоредивим са другим великим градовима у САД-у [7]. Пре споразума, цене паркирања у Чикагу биле су много ниже од националног просека. Седамдесет процената бројила није забележило пораст у 20 година.

Функционалности апликације укључују следеће:

- Доступност информација и проналажење паркинг места: Апликација пружа корисницима информације о локацији зона за паркирање и њиховим тарифама што помаже корисницима да унапред планирају своје паркирање.
- Плаћање паркинга: Корисници могу да плаћају паркинге директно путем својих паметних телефона.
- Продужење времена паркирања: Корисници могу продужити време паркирања са било које локације, што им омогућава флексибилност и удобност.
- Подсетници: Апликација шаље подсетнике корисницима када се приближава крај њиховог времена за паркирање, како би могли на време да га продуже ако је то потребно.
- Праћење историје паркирања: Корисници могу прегледати и пратити историју својих трансакција у апликацији, што им омогућава бољи преглед и контролу трошкова.

Ове функционалности чине апликацију практичним алатом за возаче, олакшавајући свакодневно паркирање и омогућавајући им да уштеде време и избегну стрес приликом тражења паркинг места.

Барселона

Барселона је имплементирала паметне паркинг системе који користе различите технологије за праћење доступности паркинг места и апликације за пружање информација корисницима. Ово је допринело побољшању ефикасности паркинг система и смањењу загађења. Град је подељен на различите паркинг зоне, као што су зелена зона (за становнике) и плава зона (за посетиоце), са различитим ценама и максималним временима паркирања. Јавни паркомати су постављени широм града и прихватају готовину, кредитне и дебитне картице, као и плаћање путем мобилних апликација. Осим тога, бројне подземне гараже нуде дуготрајно паркирање на кључним локацијама у граду.

Технологије које се користе у систему паркирања у Барселони:

- **IoT сензори:** Сензори постављени на паркинг местима омогућавају праћење заузетости у реалном времену. Ови сензори шаљу податке централном систему, који их обрађује и пружа корисницима ажуриране информације о слободним местима.
- **Аутоматско Препознавање Регистарских Ознака (ANPR):** Камере са технологијом за аутоматско препознавање регистарских ознака користе се за идентификацију возила и проверу да ли су платили паркирање. Ова технологија такође помаже у откривању и спречавању прекршаја паркирања.
- **Паметни паркомати:** Модерни паркомати прихватају различите методе плаћања и повезани су са централизованим системом који управља паркирањем у граду. Ови паркомати су повезани са IoT сензорима и мобилним апликацијама, што омогућава интегрисано управљање паркингом.

Постоји више апликација за подршки паркирања које се могу користити у Барселони, међутим, званична апликација града Барселоне је **ApparkB**. Она омогућава корисницима да плаћају паркирање у различитим зонама и продужавају време паркирања. Апликација пружа информације о доступним паркинг местима у реалном времену. Апликација поред основних функционалности као што су плаћање паркинга, продужавање времена паркирања, подсетници у виду обавештења, информације о доступности паркинга, историје паркирања и различитих метода плаћања има и додатне карактеристике. Оне се огледају у интерактивној мапи, посебном режиму за становнике и подршка и кориснички сервис.

5. Инфраструктура и начини прикупљања података

Аспект који је од изузетног значаја за представљени проблем јесте инфраструктура која је потребна и начини за прикупљање потребних података. Циљ инфраструктуре јесте да обезбеди податке који ће се користити. Потребно је да подаци који бивају прикупљени буду релевантни и у реалном времену. Ове ствари су од кључног значаја за интегритет решења које се нуди.

Због тога је од великог значаја изучавање постојећих начина прикупљања података. Постоји више начина прикупљања података, као што су:

- **Традиционални усмени начин прикупљања података**

Овај начин прикупљања података о слободним и заузетим паркинг местима подразумева ангажовање људи који физички проверавају стање на терену. То су обично запослени у јавним службама или приватним компанијама који редовно обилазе паркинг зоне и бележе информације о доступности паркинг места.

Ова метода је временски захтевна, а у неким случајевима и финансијски. Тежа је за скалирање и не пружа увек податке у реалном времену. Проширење овог начина прикупљања података на веће подручје или већи број паркинг места захтева знатно повећање броја запослених и логистичке подршке, што није увек изводљиво или економично. Усмени начин ретко пружа податке у реалном времену, јер је потребно време да запослени обиђу паркинг зоне и обраде податке. Ово може довести до застарелих информација које нису корисне за кориснике који траже слободно паркинг место у датом тренутку.

Ипак, може бити корисна у мање развијеним подручјима или као додатна контрола уз друге технологије. У мањим или руралним заједницама где инвестиција у сензоре и камере можда није економски исплатива, људски надзор може бити одрживо и практично решење. Ова метода омогућава пружање основних информација без потребе за великом почетном инвестицијом у технологију. Запослени могу директно комуницирати са возачима и грађанима, прикупљајући не само податке о слободним местима већ и повратне информације о општем корисничком искуству, што може бити корисно за побољшање услуга. Људи могу да прикупљају различите типове података које машине можда не могу лако да идентификују. На пример, могу да бележе додатне информације о условима на терену, као што су оштећења на паркинг местима, препреке, или друге ванредне ситуације.

- **Сензори за детекцију присуства возила [8]**

Сензори су уређаји који се постављају директно на паркинг места или у њиховој близини како би детектовали присуство возила. Постоји неколико врста сензора који се користе у ову сврху:

- **Индуктивни сензори:** Ови сензори се уграђују у асфалт и мере промене у магнетном пољу када се возило паркира изнад њих. Имају висок степен прецизности у детекцији металних објеката. Отпорни су на већину спољашњих утицаја, укључујући промене температуре и временске услове. Имају дуг век трајања са минималним потребама за одржавањем. Међутим, имају високе трошкове уградње због потребе за постављањем испод асфалта. Постоји могућност оштећења током асфалтирања или реконструкције путева и нису

ефикасни за детекцију неметалних објеката.

- **Ултразвучни сензори:** Ови сензори се обично постављају на паркинг структуре (плафони, зидови) и мере растојање до најближе препреке, чиме утврђују присуство возила. Користе звучне таласе за мерење удаљености. Могу се лако инсталирати и на постојеће структуре. Није потребна интервенција на асфалу. Добри су за детекцију возила различитих величина и облика. Са друге стране, могу бити осетљиви на спољашње факторе као што су киша, снег или прашина. Могуће су грешке у детекцији у случају препрека или објеката у близини и потребно је редовно одржавање и калибрација.
- **Инфрацрвени сензори:** Користе инфрацрвене зраке да детектују присуство возила и могу се поставити на различите структуре око паркинг места. Добри су за детекцију возила у различитим условима осветљења и ефикасни су у детекцији покретних објеката. Нису захтевни за инсталацију. Мане ових сензора су што могу бити осетљиви на временске услове као што су магла и снег и потребно је редовно чишћење и одржавање како би се обезбедила прецизност.

Информације које пружају сензори су тачне и брзе. Могу се користити у различитим окружењима, од затворених паркинг гаража до отворених јавних паркинг места. Омогућавају аутоматизацију тако што смањују потребу за људским надзором и ручним прикупљањем података.

Међутим и они са собом носе одређене недостатке и ризике. Захтевају постављање и редовно одржавање што може бити скупо, посебно у већим системима ови трошкови инсталације и одржавања могу бити велики. Осим тога, могу бити осетљиви на спољашње факторе као што су временске прилике и механичка оштећења. Подаци прикупљени од сензора могу бити осетљиви на хакерске нападе и злоупотребу ако системи нису адекватно заштићени.

- **Камере [8]**

Камере, нарочито оне са напредним алгоритмима за обраду слика, могу бити веома ефикасне у детекцији слободних паркинг места. Камере високе резолуције се постављају на стратешким локацијама као што су улаз и излаз паркинг зона, стубови и слично.

- **Видео аналитика:** Са коришћењем алгоритама машинског учења и вештачке интелигенције за аутоматску анализу видео снимака и слика у реалном времену. У контексту управљања паркингом, ова технологија омогућава бројне функције које побољшавају ефикасност и корисничко искуство. Неке од њих су идентификација слободних и заузетих паркинг места, бројање возила, анализа понашања, детекција неправилног паркирања, интеграција са другим системима. Алгоритми могу да анализирају видео снимке са камера постављених на паркинг местима и аутоматски идентификују слободна и заузета паркинг места. Ови алгоритми препознају присуство возила на основу облика, боје и других визуелних карактеристика. Видео аналитика може прецизно бројати возила која улазе и излазе из паркинг зона. Ова функција је корисна за праћење попуњености паркинга и управљање протоком саобраћаја. Такође може се анализирати понашање возача и возила, укључујући време

паркирања, трајање боравка и фреквенцију коришћења одређених паркинг места. Ови подаци могу помоћи у оптимизацији распореда и политика паркинга. Даље може открити неправилно паркирање, као што је паркирање на забрањеним местима, блокирање излаза или паркирање на местима за инвалиде без одговарајуће дозволе. Системи могу аутоматски обавестити надлежне службе о прекршајима. Могућа је и интеграција са другим системима као што су системи за аутоматску наплату, контролу приступа и управљање саобраћајем, чиме се обезбеђује свеобухватно решење за управљање паркингом.

- **Аутоматско препознавање регистарских таблица (ALPR):** Камере са ALPR технологијом могу не само да детектују присуство возила већ и да препознају и евидентирају регистарске таблице, што може бити корисно за праћење корисника и аутоматско наплаћивање паркинга. Ова технологија функционише на следећи начин: Камере снимају слике возила и њихових регистарских таблица. Снимљене слике се обрађују у реалном времену како би се идентификовала регистарска таблица. Софтвер користи алгоритме за оптичко препознавање карактера(OCR) како би извукао текст са таблице. Издвојени текст се упоређује са базом података ради идентификације возила. Поред база података о регистрованим возилима вај процес може укључивати и проверу база украдених возила или других релевантних података. На крају, у зависности од саме апликације, систем може аутоматски предузети одређене акције као што су отварање рампе на паркингу, издавање казни за незаконито паркирање и слично.

Коришћење камера представља моћно средство за модерно управљање паркингом, које омогућава напредну аутоматизацију и анализу, побољшавајући ефикасност и корисничко искуство. Обезбеђује податке у реалном времену, што је од суштинског значаја за динамичко управљање паркингом и брзо реаговање на промене. Могу се лако проширити и прилагодити различитим величинама и типовима паркинг простора, од малих паркинг гаража до великих јавних паркинга. Поред управљања паркингом, покривеност паркинг простора камерама може допринети унапређењу безбедности, детекцијом сумњивих активности или кршења безбедносних протокола. Међутим, њихова примена захтева пажљиво планирање и управљање како би се максимизирале предности и ублажили потенцијални недостаци. Мане коришћења ове инфраструктуре и прикупљања података су: велики трошкови инсталације и одржавања система посебно у већим или комплекснијим окружењима, зависност од квалитета видео снимака - лоши услови осветљења или временске прилике могу утицати на тачност детекције, потреба за великим рачунарским ресурсима - обрада видео снимака у реалном времену захтева значајне рачунарске ресурсе, што може повећати трошкове и комплексност инфраструктуре.

- **Комбиновани начин**

Представља комбинацију било којих претходно наведених начина прикупљања података, било да имамо комбинацију рецимо сензора и камера или пак сва три поменута начина. Да ли ће и у којој мери бити коришћено комбиновано прикупљање података, зависи од више фактора. Између осталог, зависи од самог решења и степена модернизације који је предвиђен. На пример, у урбаним срединама са високим нивоом саобраћаја и комплексним паркинг системима, комбинација сензора и камера може пружити најтачније и најпоузданије резултате.

6. Истраживање технологија

Софтверски производ који представља решење описаног проблема може бити и мобилна и веб апликација. Шта више, ради доступности и флексибилности у великим паркинг системима корисницима су најчешће доступна оба типа апликација.

Интеграција са навигационим системима и мапама као што су Google мапе је кључна компонента модерних паркинг апликација. Ова интеграција омогућава корисницима да лако проналазе паркинг локације у близини и добију навигацију до њих. Google мапе обезбеђују детаљне картографске податке, сателитске снимке и информације о саобраћају, што корисницима омогућава да избегну гужве и брже пронађу паркинг место.

Python је један од најпопуларнијих програмских језика за развој софтвера у области машинског учења и анализе података. Због своје читљивости, једноставности и богатства библиотека, Python се често користи у развоју апликација за детекцију паркинг места. Библиотеке као што су OpenCV, TensorFlow и PyTorch омогућавају лаку имплементацију алгоритама за обраду слика и машинско учење, што је од велике важности за развој софтвера који користи неки алгоритам за детекцију објеката.

У софтверским решењима намењеним паркингу, алгоритми за детекцију објеката играју кључну улогу у аутоматској идентификацији слободних и заузетих паркинг места путем анализа видео снимка са камера. Ови алгоритми омогућавају системима да препознају различите врсте објеката, као што су возила, и на основу тога одреде статус паркинг места. Постоје бројни алгоритми за детекцију објеката, најпознатији међу њима су следећи:

YOLO (You Only Look Once)

YOLO (You Only Look Once) је један од најпопуларнијих и најефикаснијих алгоритама за детекцију објеката. Његове карактеристике укључују:

- **Брзина:** YOLO је дизајниран да обрађује видео у реалном времену. Овај алгоритам може да анализира видео снимке и детектује објекте у само једном пролазу кроз неуронску мрежу, што омогућава брзу и ефикасну обраду.
- **Прецизност:** YOLO је веома прецизан у детекцији различитих објеката, укључујући возила, што је од кључне важности за тачну идентификацију слободних и заузетих паркинг места.
- **Једноставност имплементације:** Python библиотеке као што су OpenCV и PyTorch омогућавају лаку интеграцију YOLO алгорита у софтверска решења за паркинг.

Алгоритми на бази конволуционих неуронских мрежа (CNN)

Конволуционе неуронске мреже (CNN) су основа за већину савремених алгоритама за детекцију објеката. Њихове карактеристике укључују:

- **Хијерархијско учење:** CNN алгоритми могу да уче различите нивое апстракције у сликама, што им омогућава да прецизно препознају сложене обрасце и објекте, као што су возила.
- **Флексибилност:** CNN могу бити обучене за препознавање широког спектра

објеката и ситуација, што их чини идеалним за употребу у различитим паркинг окружењима.

- **Интеграција са другим технологијама:** CNN се често користе у комбинацији са другим методама машинског учења и обраде слика, што омогућава развој напредних функција као што су праћење возила и предвиђање слободних места.

Регијске конволуционе неуронске мреже (R-CNN)

R-CNN (Region-based Convolutional Neural Networks) су напредна верзија CNN алгоритама која се користи за прецизнију детекцију објеката. Њихове карактеристике укључују:

- **Висока прецизност:** R-CNN алгоритми детектују објекте кроз идентификацију региона интересовања (ROI) у сликама, што омогућава прецизније препознавање возила.
- **Сложенија обрада:** R-CNN користе више фаза обраде, што може повећати време потребно за анализу, али истовремено омогућава већи ниво детаља и тачности.
- **Употреба у комбинацији са YOLO:** R-CNN се често комбинују са бржим алгоритмима као што је YOLO за постизање оптималног баланса између брзине и прецизности.

Алгоритми за детекцију објеката, играју важну улогу у аутоматској идентификацији слободних паркинг места путем анализа видео снимака. Ови алгоритми омогућавају брзу и прецизну детекцију возила, што је од кључне важности за ефикасно управљање паркинг системима. Комбинација ових технологија са Python библиотекама и навигационим системима, као што су Google мапе, омогућава стварање свеобухватних и кориснички прилагођених решења за проблем паркирања.

7. Реализација апликације

У оквиру овог поглавља видећемо како је реализована апликација „**Parking Finder**“. Након извршеног истраживања у претходним поглављима и анализе различитих могућности утврђене су технологије погодне за конкретно решење „Parking Finder“. У питању је **веб апликација**, за чију реализацију су коришћене следеће технологије. За имплементацију клијентске стране коришћени су **JavaScript, HTML, CSS** као и **Google Maps API** за интеграцију са интерактивним мапама. На серверској страни коришћен је **Python** који заједно са **Flask**-ом чини бекенд сервер. Обрада података на бекенду врши се уз помоћ **YOLOv8** модела, којим се детектују објекти са слика и видео записа. Као складиште података коришћен је **.csv** фајл.

Биће приказана детаљна архитектура апликације и опис технологија које су коришћене, а на практичном примеру ће бити демонстрирана детекција објеката, конкретно аутомобила и паркинг места, уз помоћ **YOLOv8** модела.

7.1. Подразумевана инфраструктура

Услов за реализацију овог пројекта представља инфраструктура у виду камера. Дакле, потребно је апликацији обезбедити податке над којима ће вршити обраду. Предвиђено је да се ти подаци повлаче са камера које су инсталиране на одређеним локацијама и да се у одређеном временском периоду стално ажурирају. Постављање и одржавање камера на предвиђеним паркинзима, улицама са паркинг зонама, јавним гаражама и слично је неопходно. Овај систем камера мора бити интегрисан са апликацијом како би јој пружио потребне податке. Како би подаци били ажурни и у реалном времену складиште података апликације треба да непрестано ажурира своје податке у што краћем временском периоду.

7.2. Архитектура апликације

С обзиром да је у питању веб апликација, у складу са тим можемо говорити о две целине од којих се она састоји. То су већ поменута клијентска(фронтенд) и серверска(бекенд) страна.

Клијентска страна(JavaScript, HTML, CSS) представља интерфејс који се приказује крајњем кориснику и она представља тачку интеракције са корисницима. Отвара се у претраживачу и интегрисана је са мапама које су неопходне за смисао самог проналажења паркинг места.

Када корисник отвори апликацију клијентска страна иницира приказ Google мапе са могућношћу претраживања локација. Корисник уноси локацију коју жели да претражи. Клијентска страна шаље захтев Google Maps API-ју да пронађе унету локацију и прикаже је на мапи. Maps JavaScript API омогућава приказ мапе у веб апликацији. Places API се користи за претраживање локација. Када корисник унесе локацију, овај API претражује и враћа резултате који одговарају унетом термину. У односу на локацију коју је унео кориснику се враћа приказ са обележеним најближим паркинг местима са информацијама о слободним, заузетим и другим релевантним подацима о паркинзима. За тачност података о слободним местима клијентска страна се ослања на сервер и она само приказује податке које је генерисао сервер. Али свакако интеграција

између ове две стране постоји и у погледу овог приказа. Geocoding API конвертује адресе (као што је унос корисника) у географске координате. Nearby Search функција из Places API се користи за претрагу објеката (у овом случају паркинг места) у близини одређене локације. Custom Markers су прилагођени маркери на мапи којима подаци о паркинг местима (као што су број слободних места) могу бити приказани. Корисник такође има могућност навигације до жељене локације. Directions API пружа услуге за израчунавање рута између две локације. Maps JavaScript API користи добијене руте и приказује их на мапи као путање са упутствима за навигацију. У следећем делу видећемо мало детаљнији опис Google Maps услуга.

7.2.1. Google Maps Platform

Како је циљ наше апликације да детектује локације паркинга са слободним паркинг местима на јавним паркинзима, интерфејс саме апликације мора бити оријентисан ка мапама. Мапа је основни приказ који корисник види када отвори апликацију. Због тога је неопходно њено интегрисање са одређеном технологијом која ће те могућности пружити.

У нашем случају имплементације „Parking Finder“ апликације користићемо конкретне **Google Maps API-је** из **Google Maps Platform-е**, као што су:

1. **Maps JavaScript API:** За приказивање интерактивне мапе у апликацији.
2. **Geocoding API:** Ако је потребно конвертовање адресе у координате или обрнуто.
3. **Places API:** За добијање информација о паркинг местима и другим објектима.
4. **Directions API:** За навигацију до паркинг места (у случајевима када је то потребно).
5. **Distance Matrix API:** За израчунавање удаљености и времена путовања до паркинг места.

Ради бољег разумевања направимо разлику између Google Maps Platform-е и Google Maps API-ја:

- **Google Maps Platform:** Општа платформа која обухвата све услуге и API-је које Google нуди за рад са мапама и геолокацијама.
- **Google Maps API:** Конкретни интерфејси који се користе за приступ услугама које нуди Google Maps Platform.

Google Maps Platform [9]

Google Maps Platform је општи назив за сет услуга и API-ја које Google пружа за интеграцију мапа, геолокацијских података и других географских информација у апликације. То укључује различите API-је за мапе, места, руте, итд. Користећи Google Maps Platform, развојни инжењери могу приступити низу алата:

1. **Maps:** Интеграција интерактивних мапа у апликације.
2. **Routes:** Планирање рута и добијање упутстава.

3. **Places:** Информације о местима, као што су ресторани, продавнице и друге ствари од интереса.

Google Maps API [9]

Google Maps API је конкретан интерфејс за програмирање апликација (API-Application Programming Interface) који омогућава приступ функционалностима које пружа Google Maps Platform. То су специфични алати и библиотеке које је могуће користити у својој апликацији за интеракцију са услугама Google мапа. Неки од најчешће коришћених API-ја укључују:

1. **Maps JavaScript API:** Омогућава уградњу интерактивних мапа у веб странице.
2. **Geocoding API:** Омогућава конвертовање адреса у географске координате и обрнуто.
3. **Places API:** Пружа информације о местима, укључујући детаље, фотографије и рецензије.
4. **Directions API:** Омогућава добијање упутстава за вожњу, ходање, или јавни превоз.
5. **Distance Matrix API:** Израчунава удаљеност и време путовања између више тачака.
6. **Roads API:** Омогућава корекцију путање возила на основу стварних путева.

Пример: Приказ тренутне локације корисника.

За приказивање тренутне локације корисника уз помоћ Google Maps API у JavaScript-у, потребна је дозвола коришћења локације од стране корисника.

У HTML датотеку потребно је укључити ознаку за Google Maps API са API кључем као што је то приказано на слици 1.

```
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY"></script>
```

Слика 1. API кључ за Google Maps API

Ово је неопходно да би се прочитао Google Maps JavaScript API, који омогућава коришћење свих функција и услуга које Google Maps нуди, укључујући и приказивање мапа и рад са геолокацијом. Без читавања овог API-а, JavaScript код неће имати приступ потребним методама и објектима за рад са Google Maps.

Код на слици 2 представља следеће. Иницијализација мапе функцијом `initMap()`, тако што креира нови `google.maps.Map` објекат са почетним центром и нивоом зумирања. Проверава да ли прегледач подржава геолокацију. Ако подржава, користи `navigator.geolocation.getCurrentPosition` да добије тренутну локацију корисника. Креира маркер на тренутној локацији и центара мапу на ту локацију.

```
function initMap() {
    var map = new google.maps.Map(document.getElementById('map'), {
        center: { lat: -34.397, lng: 150.644 },
        zoom: 6
    });

    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(function(position) {
            var pos = {
                lat: position.coords.latitude,
                lng: position.coords.longitude
            };

            var marker = new google.maps.Marker({
                position: pos,
                map: map,
                title: 'Ваша тренутна локација'
            });

            map.setCenter(pos);
        });
    }
}
```

Слика 2. Приказ тренутне локације

Серверски део генерише садржај који треба вратити клијентима када затраже неке информације. Овај део представља апликацију на којој се налази бизнис логика и обрада података да би се генерисале повратне информације. Ова апликација се налази на неком веб-серверу који је стално доступан прегледачима који му шаљу HTTP захтеве. У нашем пројекту бекенд је Flask у Python-у и складиште података у облику .csv фајла. Flask представља веб микрофрејмворк који омогућава креирање бекенд сервера. За имплементацију бекенд-а користимо Flask, микрофрејмворк у Python-у [11]. Flask омогућава креирање и хостовање веб сервера и дефинисање руте за различите захтеве. Служи као посредник између базе података и фронтенд-а. Омогућава сигурну и једноставну размену података у JSON формату [11]. Бекенд сервер је дизајниран да обрађује RESTful HTTP захтеве. Након избора локације, клијентска страна шаље захтев серверу за најближа паркинг места и за релевантне информације. Сервер обрађује захтев, преузима податке из складишта (.csv фајл) који се ажурирају сваких 20 секунди са камера. Овај фајл садржи информације о броју слободних паркинг места. YOLOv8 модел детектује објекте (возила) на сликама са камера и ажурира број слободних места. Резултати детекције се користе за ажурирање броја слободних паркинг места у .csv фајлу. Сервер враћа клијентској страни податке о најближим паркинг местима и броју слободних места.

7.2.2. Flask и .csv

У оквиру наше апликације имамо коришћење Flask-а као бекенд сервера. Flask је, као што је поменуто, микро веб оквир за Python који омогућава развој веб апликација једноставним и лако проширивим приступом. Његова главна улога је да обезбеди

основу за креирање веб сервера који ће управљати захтевима корисника и враћати одговарајуће одговоре. Flask је дизајниран да буде једноставан за употребу, али довољно моћан за изградњу сложених веб апликација. Тако да је идеалан за потребе наше апликације јер омогућава брз развој са флексибилношћу додавања функционалности по потреби. Flask је одговоран за посредовање између клијената и базе података(csv фајла), осигуравајући да подаци буду ажурни и доступни у реалном времену, и омогућавајући корисницима да ефикасно проналазе паркинг места.

Комуникација са клијентом

1. **Пријем захтева:** Клијент (као што је веб претраживач) шаље HTTP захтев Flask серверу. Захтеви могу бити различитих типова, као што су GET, POST, PUT, DELETE, у зависности од акције коју клијент жели да изврши.
2. **Обрада захтева:** Flask обрађује ове захтеве помоћу дефинисаних рута и функција [11]. Руте су путеви на које клијенти шаљу захтеве, а функције су одговорне за обраду тих захтева [11].
3. **Слање одговора:** Након обраде захтева, Flask враћа одговор клијенту. Одговор може бити HTML страница, JSON подаци, или било који други формат података [11]. Ово омогућава клијентима да добију потребне информације или потврду о успешно извршеним радњама.

Комуникација са системом камера

1. **Пријем видео стрима:** Flask сервер може примати видео стримове са камера преко HTTP захтева или користећи библиотеке као што су OpenCV за директно хватање видео стрима.
2. **Обрада видео стрима:** Када се видео подаци приме, они се могу обрађивати у реалном времену помоћу YOLOv8 модела за детекцију објеката. Ово обично подразумева употребу библиотека као што су OpenCV и PyTorch.

Комуникација са CSV фајлом

Читање података из CSV фајла: Flask може читати податке из CSV фајла користећи библиотеке као што су csv или pandas. Ови подаци се могу користити за иницијализацију базе података или за пружање информација у одговорима на захтеве.

csv (Comma-Separated Values) фајл је текстуални фајл који користи зарезе за раздвајање вредности. Ови фајлови се често користе за чување табеларних података. У нашој апликацији, подаци о паркинг местима се чувају у .csv фајлу. Python користи библиотеке као што су pandas или csv за читање и писање ових података.

У оквиру наше апликације .csv фајл служи као симулација базе података. Једна могућност је да фајл садржи информације о сликама и видео записима у виду url путања. Тада би бекенд сервер морао да прибави те податке у виду слика и видео записа и обрађује их како би генерисао резултате и поново их сместио у .csv фајл. Овај начин коришћења CSV фајла као симулације базе података за чување URL путања до слика и видео записа може бити прикладно за мање пројекте или током развоја апликације, али није оптимално за системе са великом количином података. Потребно је да наша апликација ради са јако великом количином података. Решење је да се

користи дистрибуирани систем. Тај дистрибуирани систем би имао део сервера који је повезан са камерама чува слике и видео записе, обрађује податке користећи YOLOv8 модел, и затим генерише CSV фајл са информацијама о обрађеним подацима. Он се може организовати на следећи начин:

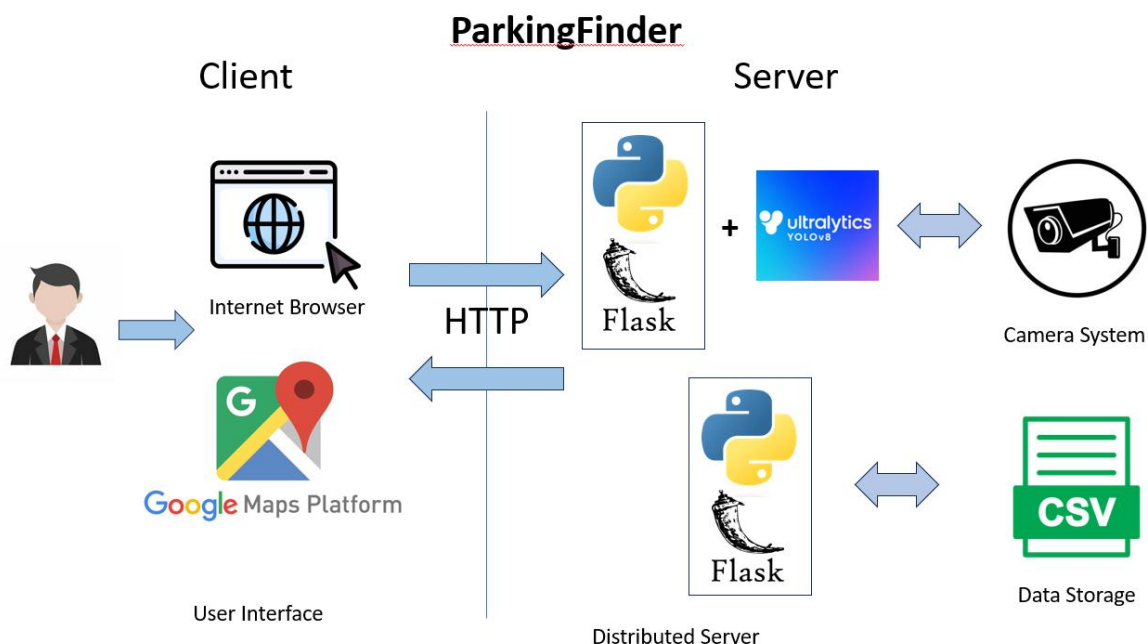
Архитектура дистрибуираног система

1. **Локални сервер код камера** – за обављање горе поменути комуникацију са системом камера
 - Локални сервер (или уређај) који је повезан са камерама преузима слике и видео записе.
 - Овај сервер користи YOLOv8 модел за обраду слика и видео записа.
 - Обрађује податке и генерише CSV фајл са релевантним информацијама (детекцијом објеката, временским печатом, броју слободних итд.).
2. **Централни сервер** – за такође поменути комуникацију са CSV фајлом
 - Централни сервер преузима CSV фајл са локалног сервера.
 - На основу информација из CSV фајла, централни сервер може даље анализирати податке или их приказивати корисницима путем веб апликације (Flask).

Могу се користити различите методе за пренос CSV фајла са локалног сервера код камера на централни сервер:

1. **FTP/SFTP:**
 - Користи FTP или SFTP за преузимање CSV фајла са локалног сервера на централни сервер.
2. **HTTP POST request:**
 - Локални сервер може послати CSV фајл централном серверу помоћу HTTP POST request-a.
3. **Cloud Storage:**
 - Локални сервер може упутовати CSV фајл на cloud storage (нпр. AWS S3, Google Cloud Storage), а централни сервер може преузети фајл са cloud-a.

Визуелна репрезентација архитектуре апликације Parking Finder се може видети на слици 3. Архитектура је приказана заједно са изгледом дистрибуираног сервера. Дистрибуирани систем омогућава ефикасну обраду података и бољу расподелу ресурса. Локални сервер код камера може обрадити слике и видео записе помоћу YOLOv8 модела и генерисати CSV фајл са резултатима, док централни сервер може преузети те податке за даљу анализу и приказивање корисницима путем веб апликације.



Слика 3. Архитектура апликације

Интеракција између компоненти:

- Корисник интерагује са веб апликацијом преко претраживача.
- Клијентска страна (JavaScript, HTML, CSS) шаље HTTP захтеве серверу.
- Сервер (Flask на Python-у) прима захтеве и прослеђује их YOLOv8 моделу за обраду.
- YOLOv8 модел врши детекцију објеката и враћа резултате серверу.
- Сервер складишти резултате у .csv фајл и враћа одговор клијентској страни.
- Клијентска страна ажурира приказ на мапи користећи Google Maps API.

7.3. Детекција објеката – пример реализације

Детекција објеката представља задатак рачунарског вида који укључује лоцирање и препознавање објеката од интереса на слици или видеу. Циљ је не само препознати присуство објеката у датом сценарију већ и одредити њихове тачне локације исцртавањем оквира око њих. Постоје бројни приступи за детекцију објеката, укључујући традиционалне технике рачунарског вида и савременије методе засноване на дубоком учењу, као што су R-CNN, YOLO, CNN, DETR и други. Ови модели се тренирају на великим скуповима података како би научили обрасце и карактеристике које им омогућавају прецизно детектовање и класификацију објеката у различитим сценаријима.

За наш конкретан проблем детекције аутомобила и слободних и заузетих паркинг места користили смо модел YOLO(You Only Look Once), односно његову најновију верзију YOLOv8. YOLO (You Only Look Once) модели су напредни модели за детекцију објеката, познати по својој брзини и прецизности. За разлику од традиционалних приступа који користе регионалне предлоге за идентификацију објеката у слици, YOLO приступ истовремено предвиђа и објекте и њихове локације у

једном пролазу кроз мрежу. Ова техника омогућава значајно бржу обраду слика, што је кључно за апликације у реалном времену као што су видео надзор, аутономна вожња и паметни градови.

Овај модел детекције објеката и сегментације слике, развили су Џозеф Редмон и Али Фархади са Универзитета у Вашингтону [10]. Лансиран 2015. године, YOLO је брзо стекао популарност због своје велике брзине и прецизности [10]. Развијан је кроз верзије YOLOv2, YOLOv3 до YOLOv8 верзије [10].

YOLOv8 је најновија верзија YOLO компаније Ultralytics [10]. Као врхунски, најсавременији (State-Of-The-Art-SOTA) модел, YOLOv8 се надограђивао на претходним верзијама, уводећи нове функције и побољшања за боље перформансе, флексибилност и ефикасност [10]. YOLOv8 подржава читав низ задатака вештачке интелигенције укључујући детекцију, сегментацију, процену позе, праћење и класификацију [10].

Предности:

1. **Брзина и ефикасност:** YOLOv8 је веома брз и може да обрађује видео записе у реалном времену, што га чини идеалним за апликације које захтевају брзу детекцију објеката.
2. **Висока прецизност:** Побољшана архитектура и методе тренинга повећавају прецизност детекције у поређењу са претходним верзијама.
3. **Једноставност коришћења:** YOLOv8 има једноставан интерфејс и добро документоване библиотеке, што олакшава имплементацију и интеграцију у различите апликације.
4. **Модерне технике тренинга:** Користи напредне технике као што су mixed precision training и data augmentation, што побољшава укупне перформансе и робусност модела.

Ограничења:

1. **Захтеви за хардвером:** Иако је ефикасан, за максималне перформансе захтева снажне GPU јединице, што може бити скупо и недоступно за све кориснике.
2. **Тешкоће са малим објектима:** Иако је прецизан, YOLOv8 може имати потешкоћа са детекцијом веома малих објеката у сложеним сценама.
3. **Тренинг и фино подешавање:** Тренинг и фино подешавање модела могу бити комплексни и захтевати много времена и ресурса.
4. **Осетљивост на осветљење и углове:** Промене у осветљењу и угловима могу утицати на перформансе модела, посебно у неким сценаријима.
5. **Потреба за великим скуповима података:** За најбоље резултате, потребни су велики и разноврсни скупови података за тренинг, што може представљати изазов у смислу прикупљања и анотације података.

Инсталирање и подешавање окружења

Пре инсталирања самог модела, потребно је, наравно, имати инсталиран Python. Даље је потребан pip који је Python-ов менаџер за пакете и PyTorch који YOLOv8 користи као свој основни оквир за дубоко учење.

Python је могуће преузети са званичног Python сајта. Верзија Python-а у којој је имплементиран пример који ће бити приказан је Python 3.12.4.

Pip обично долази уз Python инсталацију. Уколико то није случај може се инсталирати командом:

```
python -m ensurepip --upgrade
```

PyTorch није укључен у стандардну инсталацију Python-а, па га је потребно накнадно инсталирати. Може се инсталирати путем pip менаџера пакета након што је већ инсталиран Python. Команда којом се може инсталирати PyTorch је:

```
pip install torch torchvision torchaudio
```

Сада када су инсталиране све потребне зависности можемо инсталирати YOLOv8. Постоји више начина за инсталирање, укључујући pip, conda и Docker. Уз помоћ pip менаџера пакета могуће је добити најновију стабилну верзију клонирањем Ultralytics GitHub репозиторијума на следећи начин:

```
git clone https://github.com/ultralytics/yolov8.git
```

или уколико није потребан изворни код, само

```
pip install ultralytics
```

Docker се може користити за извршавање пакета у изолованом контејнеру, избегавајући локалну инсталацију.

Сада када је инсталиран може се користити и из Ultralytics командног интерфејса (CLI) као и из било ког Python окружења.

Ultralytics CLI омогућава једноставне једнолинијске команде и не захтева Python окружење. Све команде се у овом случају могу покренути из терминала уз помоћ yolo команде. Ultralytics yolo команда користи следећу синтаксу:

```
yolo TASK MODE ARGS
```

where TASK (optional) is one of [detect, segment, classify]
 MODE (required) is one of [train, val, predict, export, track]
 ARGS (optional) are any number of custom 'arg=value' pairs like 'imgsz=320' that override defaults.

TASK (опционо поље) је једна од [detect, segment, classify]. Ако није експлицитно прослеђен, YOLOv8 ће покушати да претпостави TASK на основу типа модела. Дакле, задатак може бити детекција објеката, сегментација објеката или њихова класификација.

MODE (обавезно поље) је једна од [train, val, predict, export, track] вредности. Оне се односе на тренинг модела, валидацију модела, прогнозу или инференцију на новим подацима, извоз модела у други формат и праћење објеката.

ARGS (опционо поље) - ARGS су додатни аргументи који се могу користити за прилагођавање параметара. Могу бити било који број arg=value парова, као што је imgsz=320 који би поставио величину слике на 320 пиксела. За комплетну листу

доступних аргумената и њихових подразумеваних вредности, потребно је погледати Configuration и датотеку defaults.yaml.

Нпр. следећа команда означава тренирање модела за детекцију објеката коришћењем слика величине 320 пиксела:

```
yolo task=detect mode=train imgsz=320
```

Остале зависности које су потребне, поред ultralytics, могу се сместити у један txt фајл, чијим извршењем ће све редом бити инсталиране. То је случај са нашим пројектом, код кога се све потребне зависности налазе у датотеци requirements.txt и могу се инсталирати на следећи начин:

```
pip install -r requirements.txt
```

Окружење у коме је реализован пример је Visual Studio Code. Команда за покретање Python скрипти је из овог окружења је:

```
python skripta.py
```

Детекција и пребројавање слободних паркинг места - пример

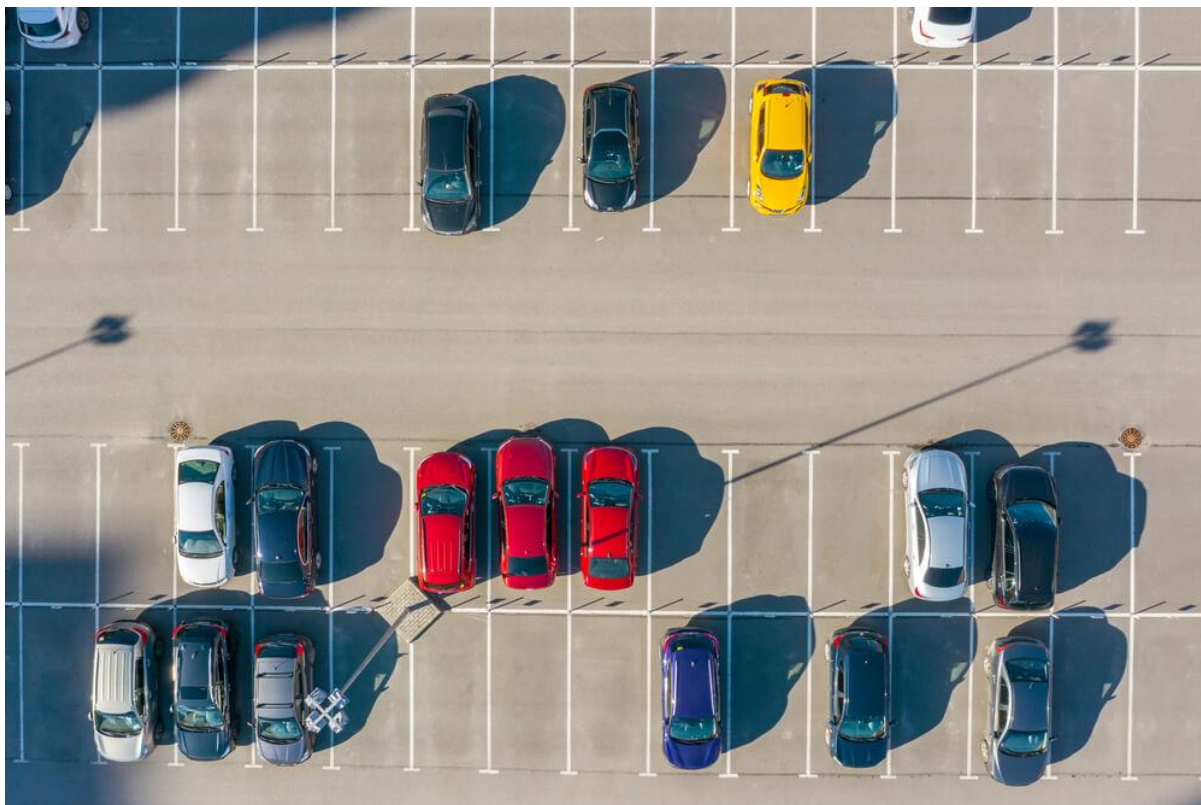
YOLOv8 је унапред претренирани модел. Један од најпопуларнијих скупова података на којима је YOLOv8 трениран је COCO (Common Objects in Context). Садржи преко 330.000 слика и 1,5 милиона инстанци објеката разврстаних у 80 категорија. Модел тренирани на COCO скупу података имају могућност препознавања и класификације широког спектра објеката, укључујући људе, возила, животиње, намештај, електронске уређаје и многе друге свакодневне предмете.

Анализираћемо Python скрипту, која се у оквиру пројекта на предатом gitHub репозиторијуму налази под називом parkingfinder.py.

Овај пример илуструје следеће аспекте рада YOLOv8 модела:

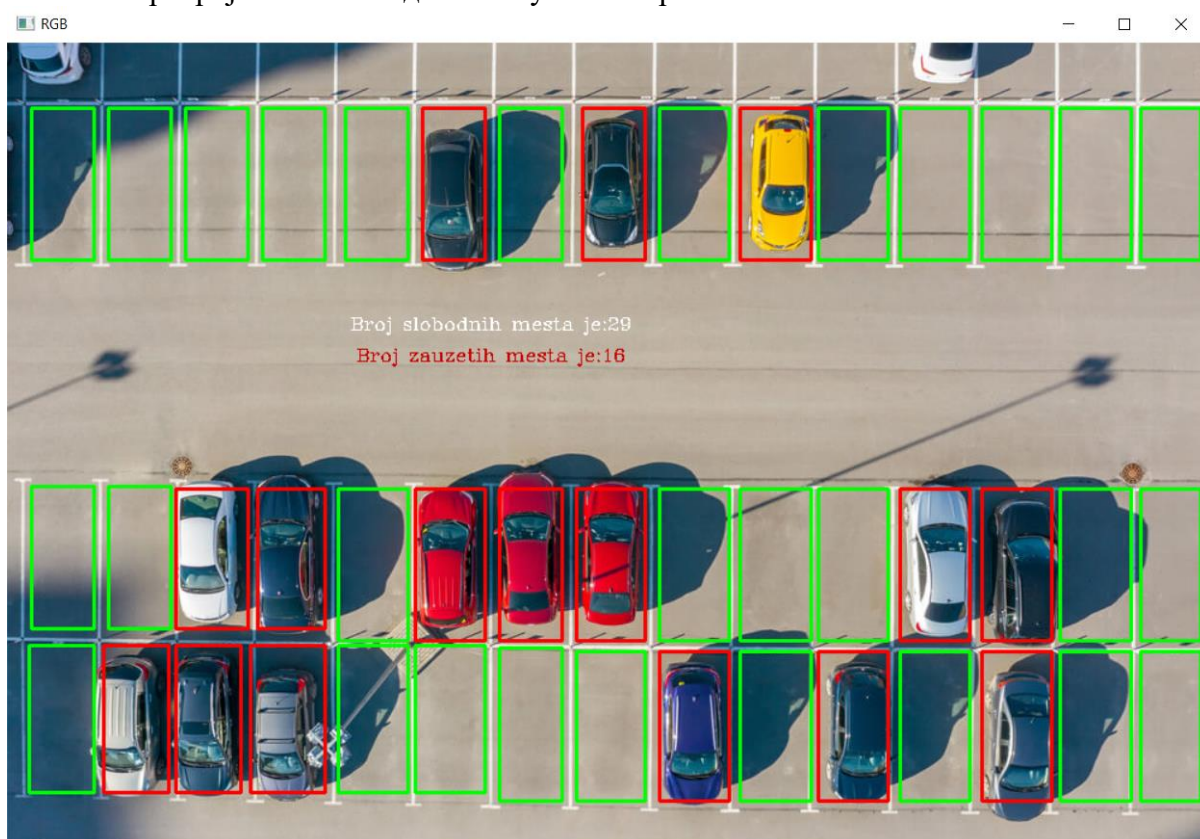
- **Детекција објеката:** Користи се претходно обучени YOLOv8 модел за детекцију возила на слици.
- **Бројање и класификација:** Детектовани објекти се класификују и броје унутар дефинисаних зона на слици.
- **Визуализација:** Резултати детекције се визуализују на слици, укључујући цртање правоугаоника око полигоналних зона.

На слици 4 видимо изглед пре детекције објеката.



Слика 4. Изглед слике пре обраде

Слика 5 је слика након што је одрађена детекција аутомобила и паркинг места и извршено пребројавање слободних и заузетих паркинг места.



Слика 5. Резултат након обраде слике

На сликама 6 и 7 дат је код којим је дефинисан начин функционисања.

На почетку је извршено учитавање YOLOv8 модела и класа објеката. Модел YOLOv8 се учитава из предтренираног фајла yolov8s.pt. о Класе објеката (као што су "car", "person", итд.) учитавају се из фајла coco.txt. Затим имамо функцију RGB која региструје координате миша када се помера преко слике. Ово може бити корисно за мануелно означавање или дебаговање. Учитавање слике паркинга врши се са путање дефинисане у image_path. Детекција објеката - Модел предвиђа објекте на слици и враћа bounding box координате за сваки детектовани објекат. Резултати детекције се конвертују у DataFrame ради лакше обраде. На слици се зоне са паркинг местима дефинишу као полигони (зоне) са координатама тачака које их описују. Пребројавање возила у свакој зони. Листе се користе за бројање возила у свакој зони. За сваки детектовани објекат, проверава се да ли је центар објекта унутар неке зоне користећи cv2.pointPolygonTest. Ако је центар објекта унутар зоне, црта се bounding box око објекта, ставља тачка на центар и додаје класа објекта у листу за ту зону.

Овај пример показује основне кораке у детекцији и пребројавању објеката на паркингу користећи YOLOv8. Модел је претрениран на COCO скупу података и може се додатно тренирати или прилагодити специфичним потребама пројекта.

```

1  import cv2
2  import pandas as pd
3  import numpy as np
4  from ultralytics import YOLO
5
6  model = YOLO('yolov8s.pt')
7
8  my_file = open("coco.txt", "r")
9  data = my_file.read()
10 class_list = data.split("\n")
11
12 def RGB(event, x, y, flags, param):
13     if event == cv2.EVENT_MOUSEMOVE:
14         colorsBGR = [x, y]
15         print(colorsBGR)
16
17 cv2.namedWindow('RGB')
18 cv2.setMouseCallback('RGB', RGB)
19
20 # Učitavanje slike
21 image_path = 'data/images/parking.jpeg'
22 frame = cv2.imread(image_path)
23
24 # Detekcija objekata
25 results = model.predict(frame)
26 boxes = results[0].boxes.data
27 px = pd.DataFrame(boxes).astype("float")

```

Слика 6. Део кода за демонстрацију YOLOv8 модела из parkingfinder.py скрипте

```

for index, row in px.iterrows():
    x1 = int(row[0])
    y1 = int(row[1])
    x2 = int(row[2])
    y2 = int(row[3])
    d = int(row[5])
    c = class_list[d]
    confidence = row[4]
    class_index = int(row[5])
    class_name = class_list[class_index]

    print(f"Detektovana klasa: {class_name}, Konfidencija: {confidence}")

    if 'car' in c:
        cx = int(x1 + x2) // 2
        cy = int(y1 + y2) // 2
        print(f"Centar automobila: ({cx}, {cy})")

        results1 = cv2.pointPolygonTest(np.array(area1, np.int32), ((cx, cy)), False)
        print(f"Rezultat za zonu 1: {results1}")
        if results1 >= 0:
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.circle(frame, (cx, cy), 3, (0, 0, 255), -1)
            list1.append(c)
            cv2.putText(frame, str(c), (x1, y1), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)

```

Слика 7. Део кода за демонстрацију YOLOv8 модела из parkingfinder.py скрипте

8. Закључак

Пројекат „Parking Finder“ представља значајан корак ка решавању проблема паркирања у урбаним срединама. Кроз примену напредних технологија и интеграцију са системима е-управе, овај пројекат има потенцијал да значајно побољша ефикасност коришћења паркинг простора, смањи гужве и загађење и побољша квалитет живота грађана.

Успех пројекта зависи од сарадње између приватног и јавног сектора, од постојања неопходне инфраструктуре и степена покривености паркинга предвиђеном инфраструктуром, као и од активног учешћа корисника.

Можемо још рећи да се проблем који смо анализирали тиче читаве људске заједнице. Негде се овом проблему успешније прилази и стаје на крај, док је у нашем случају заједница још увек негде на почетку пута његовог решавања. Извесне мере постоје али је потребно још много труда и рада како би остварено решење било на завидном нивоу.

Овај рад је само почетак истраживања и развоја поменутог проблема, и потребно је наставити са дубљим анализама и тестирањем како би се обезбедили оптимални и најбољи резултати у пракси. Уз посвећеност, иновације и сарадњу, проблем паркирања може постати ствар прошлости, а урбани живот бити значајно побољшан.

9. Литература

1. Наташа Вељковић, *Развој и имплементација оквира за евалуацију и праћење отворености е-управе*, Докторска дисертација, Ниш, 2015
2. Званична веб локација ParkWhiz апликације, доступна на <https://www.parkwhiz.com/>
3. Званична веб локација SpotHero апликације, доступна на <https://spothero.com/>
4. Званична веб локација Parkopedia апликације, доступна на <https://www.parkopedia.com/>
5. Званична веб локација eParking апликације, доступна на <https://eparking.rs/login>
6. SFpark Evaluation, SFMTA, доступан на <https://www.sfmta.com/getting-around/drive-park/demand-responsive-pricing/sfpark-evaluation>
7. ParkChicago, доступан на <https://parkchicago.com/>
8. Trista Lin, Hervé Rivano, and Frédéric Le Mouél, *A Survey of Smart Parking Solutions*, IEEE Transactions on Intelligent Transportation Systems, 2017.
9. Званична документација Google Maps Platform-е, доступна на <https://developers.google.com/maps/documentation>
10. Званична Ultralytics YOLO документација, доступна на <https://docs.ultralytics.com/>
11. Званична Flask документација, доступна на <https://flask.palletsprojects.com/en/3.0.x/>