|  |  |  |
| --- | --- | --- |
|  | UNIVERZITET U NOVOM SADU  **FAKULTET TEHNIČKIH NAUKA U NOVOM SADU** |  |

Иван Мршуља

**Праћење и препознавање геста шаке комбинацијом неуронских и традиционалних приступа**

Дипломски рад

- Основне академске студије -

Нови Сад, 2022.

|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**  21000 НОВИ САД, Трг Доситеја Обрадовића 6 | Датум: |
|  |
| **ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА** | Лист: |
| 1/1 |

*(Податке уноси предметни наставник - ментор)*

| Врста студија: | **Основне академске студије** |
| --- | --- |
| Студијски програм: | **Софтверско инжењерство и информационе технологије** |
| Руководилац студијског програма: | **проф. др Мирослав Зарић** |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент: | **Иван Мршуља** | Број индекса: | **SW 65/2018** |
| Област: | **Електротехничко и рачунарско инжењерство** | | |
| Ментор: | **Др Јелена Сливка, редовни професор** | | |
| НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:   * проблем – тема рада; * начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; * литература | | | |

**НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:**

|  |
| --- |
| **Праћење и препознавање геста шаке комбинацијом неуронских и традиционалних приступа** |

**ТЕКСТ ЗАДАТКА:**

|  |
| --- |
| 1. Анализирати стање у области.  2. Израдити спецификацију захтева софтверског решења.  3. Израдити спецификацију дизајна софтверског решења.  4. Имплементирати софтверско решење према израђеној спецификацији.  5. Тестирати имплементирано софтверско решење.  6. Документовати (1), (2), (3), (4) и (5). |

|  |  |
| --- | --- |
| Руководилац студијског програма: | Ментор рада: |
|  |  |

|  |
| --- |
| Примерак за:  - Студента;  - Ментора |

# КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

|  |  |
| --- | --- |
| Редни број, **РБР**: |  |
| Идентификациони број, **ИБР**: |  |
| Тип документације, **ТД**: | монографска публикација |
| Тип записа, **ТЗ**: | текстуални штампани документ |
| Врста рада, **ВР**: | дипломски рад |
| Аутор, **АУ**: | Иван Мршуља |
| Ментор, **МН**: | Др Јелена Сливка, редовни професор |
| Наслов рада, **НР**: | Наслов рада |
| Језик публикације, **ЈП**: | српски |
| Језик извода, **ЈИ**: | српски / енглески |
| Земља публиковања, **ЗП**: | Србија |
| Уже географско подручје, **УГП**: | Војводина |
| Година, **ГО**: | 2022 |
| Издавач, **ИЗ**: | ауторски репринт |
| Место и адреса, **МА**: | Нови Сад, Факултет техничких наука, Трг Доситеја Обрадовића 6 |
| Физички опис рада, **ФО**: | бр. поглавља / страница / цитата / табела / слика / графикона / прилога |
| Научна област, **НО**: | Софтверско инжењерство и информационе технологије |
| Научна дисциплина, **НД**: | Софтверско инжењерство |
| Предметна одредница /  кључне речи, **ПО**: | Праћење шаке, препознавање геста шаке |
| **УДК** |  |
| Чува се, **ЧУ**: | Библиотека Факултета техничких наука, Трг Доситеја Обрадовића 6, Нови Сад |
| Важна напомена, **ВН**: |  |
| Извод, **ИЗ**: | апстракт – један пасус који добро описује суштину рада – проблем, мотивацију, назнаку решења и резултат. |
| Датум прихватања теме, **ДП**: |  |
| Датум одбране, **ДО**: |  |
| Чланови комисије, **КО**: |  |
| председник | др Име Презиме, звање |
| члан | др Име Презиме, звање |
| ментор | др Име Презиме, звање |
| Потпис ментора | |

# KEY WORDS DOCUMENTATION

|  |  |
| --- | --- |
| Accession number, **ANO**: |  |
| Identification number, **INO**: |  |
| Document type, **DT**: | monographic publication |
| Type of record, **TR**: | textual material |
| Contents code, **CC**: | bachelor thesis |
| Author, **AU**: | Ivan Mršulja |
| Mentor, **MN**: | Jelena Slivka, full professor, PhD |
| Title, **TI**: | Hand tracking and gesture recognition using neural and traditional model pipeline |
| Language of text, **LT**: | Serbian |
| Language of abstract, **LA**: | Serbian / English |
| Country of publication, **CP**: | Serbia |
| Locality of publication, **LP**: | Vojvodina |
| Publication year, **PY**: | 2022 |
| Publisher, **PB**: | author’s reprint |
| Publication place, **PP**: | Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6 |
| Physical description, **PD**: | br. poglavlja / stranica / citata / tabela / slika / grafikona / priloga |
| Scientific field, **SF**: | Software Engineering and Information Technologies |
| Scientific discipline, **SD**: | Software Engineering |
| Subject / Keywords, **S/KW**: | Hand tracking, gesture recognition |
| **UDC** |  |
| Holding data, **HD**: | Library of the Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad |
| Note, **N**: |  |
| Abstract, **AB**: | Prevod apstrakta na engleski |
| Accepted by sci. Board on, **ASB**: |  |
| Defended on, **DE**: |  |
| Defense board, **DB**: |  |
| president | Ime i prezime, zvanje na eng., PhD |
| member | Ime i prezime, zvanje na eng., PhD |
| mentor | Ime i prezime, zvanje na eng., PhD |
| Mentor's signature | |

**SADRŽAJ**

[КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА 4](#_Toc94596412)

[KEY WORDS DOCUMENTATION 5](#_Toc94596413)

[1. UVOD 9](#_Toc94596414)

[2. PREGLED STANJA U OBLASTI 11](#_Toc94596415)

[3. TEORIJSKI POJMOVI I DEFINICIJE 17](#_Toc94596416)

[3.1 Opis algoritma na koji se vaš rad oslanja 17](#_Toc94596417)

[3.2 Opis algoritma na koji se vaš rad oslanja 17](#_Toc94596418)

[4. METODOLOGIJA 19](#_Toc94596419)

[4.1 Prvi modul sistema (izmeniti naslov da bude specifičan za vaše rešenje) 23](#_Toc94596420)

[4.2 Drugi modul sistema 23](#_Toc94596421)

[4.3 Korišćeni alati 23](#_Toc94596422)

[5. EKSPERIMENTI 25](#_Toc94596423)

[5.1 Skup(ovi) podataka 25](#_Toc94596424)

[5.2 Eksperiment 1 26](#_Toc94596425)

[5.3 Evaluacija 27](#_Toc94596426)

[6. REZULTATI (I DISKUSIJA) 29](#_Toc94596427)

[7. DISKUSIJA 31](#_Toc94596428)

[8. ZAKLJUČAK 33](#_Toc94596429)

[9. LITERATURA 35](#_Toc94596430)

[10. BIOGRAFIJA 37](#_Toc94596431)

# UVOD

Могућност рачунара да перципира облик и кретање шаке може бити једна од виталних компоненти која побољшава *user experience-*ау широком спектру технолошких домена и платформи. Праћење шаке и препознавање гестикулације исте представљају један од основних проблема у овом домену истраживања.

На пример, решење горе поменутог проблема, може да представља основу за разумијевање знаковног језика и контролу покрета рукама, а такође може да омогући преклапање дигиталног садржаја и информација из физичког свијета у системима проширене стварности (augmented reality). Иако је људима природна, робусна перцепција руку у реалном времену је изразито изазован задатак у домену компјутерске визије, јер руке често заклањају саме себе или једна другу (нпр. оклузије прста/длана и дрхтање руку) и немају регионе високог контраста.

У овом раду, рађено је праћење и препознавање геста шаке у домену разумијевања знаковног језика, конкретно, разумијевања алфабета немачког језика.

За рјешавање овог проблема, имплементиран је јеноставан pipeline koji ima dva koraka. Sistem kao ulazne podatke dobija frejmove real-time снимка са веб камере које даје горе поменутом pipeline-у на обрађивање. Prvi korak u obradи слике јесте одређивање 21 кључног oбележја (landmark-а) шаке док други корак представља класификациони модел. Класификациони модел као улаз прима координате сваког од 21 кључног обележја и као излаз даје коначну предикцију геста шаке. Од технологија коришћене су MediaPipe радни оквир (framework) за детекцију кључних обележја шаке као и *scikit-learn* и *Keras* библиотеке за имплементацију класификационих модела.

Евалуација модела вршена је евалуацијом квалитета коначне предикције истог. Квалитет одређивања кључних обележја није евалуиран посебно. Разлог за ово је тај што први корак у проточној обради (детекција и локализација кључних обележја) служи као својеврсни feature extractor те се као сваки други вид претпроцесирања улазне слике не евалуира сам за себе већ у склопу читавог модела проточне обраде. С обзиром да је улазни скуп података релативно балансиран, као мјере перформансе користе се тачност, прецизност, одзив као и микро Ф мјера. Такође, пошто је веома битно да овај модел ради што ближе реалном времену, као мјеру перформансе квалитета рада у реалним условима користи се однос фрејмова које модел може да обради у секунди (Frames Per Second) и микро Ф мјере. Главни закључци који су изведени из евалуације показују да овај приступ има најбољи однос између брзине обраде и квалитета предикције у односу на потпуно трдиционлне и потпуно неуронске приступе (поготово на слбијем хрдверу). Такође, имплементирани приступ се показао супериорно у односу на потпуно неуронске приступе и у домену обучавања јер му је потребно доста мање података за обучавање. На крају, треба напоменути да је мој приступ до одређене мјере отпоран код примјене на различитим нијансама коже као и заклањања шаке што није случај код традиционалних приступа.

Додати пасус са организацијом остатка рада по поглављима.

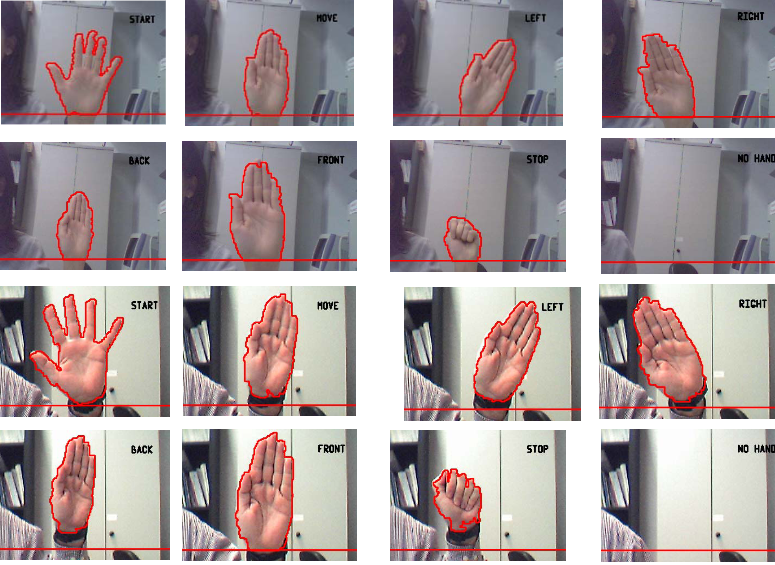
# PREGLED STANJA U OBLASTI

У овом поглављу је наведен историјат развоја приступа за праћење и препознаваље геста шаке као и тренутни state-of-the-art приступи код рјешавања овог проблемског домена. Код избора релевантних радова вођено је рачуна да су представљене методологије имале довољно добре перформансе приликом класификације како би се могли користити у пракси као и да су употребљиви из аспекта интеракције човека-рачунара тј. да се резултат могао добити у реалном времену или близу реалног времена као и да је погодан за реалну примјену. Такође, узети су у обзир само они радови који класификацију раде искључиво коришћењем фрејмова видеа као улаза те су сви Skeleton-Based, Depth-Based i 3D-Based modeli otpisani u startu.

Први приступ за праћење и препознавање геста шаке кој је дао задовољавајућ резултат за реалну употребу био је заснован на pipeline-у који се састојао из три фазе:

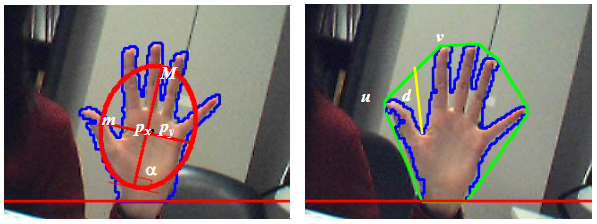
* традиционалне технике сегментације помоћу пробабилистичког модела
* ручног одређивања релевантних обележја
* систем заснован на правилима (rule-based) који је служио за одређивање геста

Аутори су развили овај систем за потребе контролисања кретања аватара у видео игрици те је коначна лабела имала 8 класа: start, move, stop, left, right, front, back, no hand. Прије почетка коришћења апликације (приликом сваког покретања) корисник треба да у исцртани квадрат на екрану услика свој длан како би сегментациони алгоритам могао да се обучи. Сви усликани пиксели длана пребацују се из RGB (Red Green Blue) у HSV (Hue Saturation Value) репрезентацију јер је она доста погоднија за даљу обраду. Функција густине вјероватноће боје коже моделована је гаусовом расподјелом а параметри се одређују стандардном методом максималне вјеродостојности (maximum likelihood estimation). Након процеса обучавања модел је спреман за употребу. За сваки пиксел улазне слике одређује се вјероватноћа да ли припада длану, а затим се користи CCA (Connected Components Аlgorithm) преко којег се пиксели длана групишу у један blob. Ивични пиксели овог blob-a представљају контуре шаке.



Примјер 2.1. Приказ одређивања контура шаке употребом пробабилистичког модела

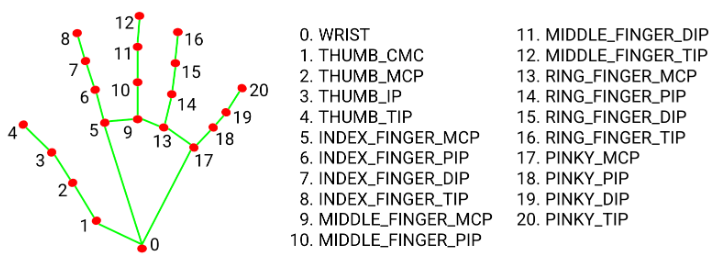
Када сегментациони алгоритам одреди контуре шаке, користи се једноставан хеуристички алгоритам који одређује центар длана. Циљ ове фазе је да се длан представи као елипса и да се одреде мала и велика оса (m, M респективно). С обзиром да као излаз из предходне фазе имамо ивичне пикселе региона, одређивање ових параметара је тривијално и своди се на одређивање линије која представља највеће растојање два пиксела унутар региона (велика оса) и линије која је нормална на њу и спаја два пиксела ивице који се налазе на супротним странама шаке (мала оса). Додатно, стандардним техникама компјутерске визије, рачуна се и конвексни овотач региона. На основу конвексног омотача и репрезентације путем елипсе могуће је одредити низ контурних тачака између два узастопна конвексна врха прста/длана. Ова секвенца формира такозвани дефект конвексности (Convexity Defect) и могуће је израчунати дубину и-тог дефекта конвексности di.



Примјер 2.2 Приказ екстракције обележја са детектованог региона од интереса

Просјек дужина свих дефеката конвексности зједно са дужином мале и велике осе, координатама центра шаке као и орјентација шаке се узимају као обележја у финалној фази која се ослања на неколико ручно написаних if-then-else правила која, на крају, дају коначан резултат процеса класификације. Модел је евалуиран на ручно прикупљеном скупу података на коме су аутори успјели да постигну веома добре перформансе. За метрику перформансе коришћена је тачност која је израчуната за сваку класу те је након тога упросјечена. Просјечна тачност на свих 8 класа износила је одличних 98%. Главна мана овог приступа је та што се приликом сваког покретања мора поново обучавти пробабилистички модел који ради сегментацију слике. Истина је да се овим постигла робусност јер ће модел радити на различитим типовима коже корисника али остаје проблем коришћења у лоше освијетљеној просторији. Наиме, модел се мора користити у соби гдје је освијетљење константно јер промјеном освјетљења добијамо потпуно различите вриједности пиксела те се пробабилистички модел мора поново обучити. Такође, у сцени гдје је позадина сличне боје као и кожа корисника (поготово на камерама лошег квалитета) аутори наводе да се модел понаша непредвидиво. Још једна од великих мана овог приступа, коју сами аутори наводе, је та што је немогуће користити овај модел код гестова приликом којих нису виљиви сви прсти.

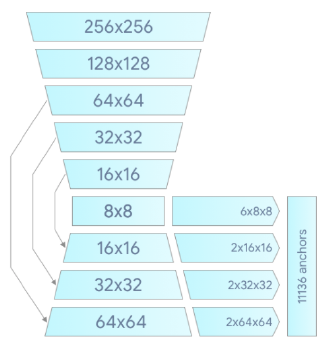
У раду [] проблем са поновним обучавањем пробабилистичком модела код употребе на различитим нијансама коже као и проблем позадине са бојом сличној боји коже превазиђен је на начин што је као додатна опрема коришћена рукавица која је офарбана бојом која се јако ријетко може наћи у позадини (и природи уопште). Самим тим, једноставном примјеном сегментације помоћу прага је могуће извући регију од интереса. Поново се ручно имплементираним техникма извлче обележја која се у овом приступу убацују у k-nearest neighbors (KNN) класификатор. Аутори овог рада су добили скоро идентичне резултате као аутори рада [] на јако сличном скупу података с тиме што су превазишли један од проблема увођењем додатне опреме. Са друге стране, идаље је остао велики проблем употребе модела у сцени која је неједнако освијетљена и генерално је остао проблем детекције геста код којег нису видљиви сви прсти. Рјешење за ове проблеме изложено је у [] гдје T. Maung предлаже да, за разлику од свих претходних модела, немамо одвојене фазе за екстракцију обележја и класификацију већ је све било енкапсулирано у једну потпуно повезану неуронску мрежу са hard-limit активационом функцијом. Овај приступ дао је око 90% тачности на ручно прикуплјеном скупу података али је имао велику ману да ради само уколико се шака налази непосредно испред камере, стога није ни адресирао проблем праћења шаке. У раду [] предложено је коришћење SSD () модела за локализацију шаке и класификацију геста (конкретно модела из YOLO породице). Иако доста тежак за обучавање, овај модел ријешио је све проблеме претходно набројаних приступа и дуго се сматрао као state-of-the-art рјешење све до појаве MediapipeHands модела. Аутори овог рада су увидјели да је главни проблем у овом проблемском домену управо екстракција обележја и ријешили су да направе pipeline модел који ће за улазну слику у RGB репрезентацији моћи да одреди 21 кључно обележје шаке које би касније било који модел машинског учења могао искористити као улаз за процес класификације геста.



Примјер 2.3 Визуализација сваког кључног обележја заједно са његовом ознаком

Њихов предложени модел има двије главне фазе. У првој фази користи се SSD модел за детекцију длана (BlazePalm detector). Разлог за ово је намјера да се ријеши проблем парцијалног или комплетног заклањања прстију шаке, такође, у њиховим експериментима се показало да је много мање података потребно за обучавање овог модела у односу на друге приступе који користе SSD модел за детекцију цијеле шаке. У другој фази, користи се Hand Landmark модел који врши прецизну локализацију сваког од 21 кључна обележја шаке путем регресије. Идеја је да модел научи конзистентну интерну структуру шаке како би могао да предвиди позиције прстију и самим тим био робустан на парцијално или потпуно заклањање истих.

Архитектура BlazePalm модела је јако слична архитектури YOLOv3 модела те се улазна слика прво скалира на 256х256 резолуцију и провлачи се кроз 5 конволутивних блокова, 3 upscale слоја и 4 блока за детекцију. Поједини конволутивни блокови имају стандардну skip конекцију са појединим upscale слојевима у циљу очувања својстава (енгл. feature) слике већег нивоа и подизања квалитета детекције. Излаз из детекционих блокова чине 11136 anchor box-a који на крају пролазе кроз стандардни non maximum suppression алгоритам у циљу добијања коначних граничних оквира дланова. Битно је нагласити да се овај модел може користити за детекцију произвољног броја дланова на снимку у исто вријеме од којих ће сваки бити обрађен. Истина је да детекција већег броја објеката може наштетити перформансе овог модела али јако се ријетко јавља потреба за више од 4 истовремено детектоване шаке те ово и не представља велики проблем.



Примјер 2.4 Архитектура *BlazePalm* модела

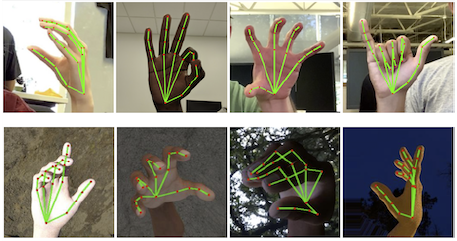
Архитехтура HandLandmark модела је специфична за овај проблемски домен. У питању је multi-head модел који има 3 излазна слоја:

1. 21 кључно обележје гдје за свако обележје имамо триплет (*x* координата, *y* координата, релативна дубина у односу на камеру)
2. Вјероватноћа да је на слици детектована шака
3. Бинарна класификација типа шаке (да ли је шака лијева или десна)

Сви излазни слојеви дијеле зједнички feature extractor. Занимљива чињеница је то, да је свака од глава (излазних слојева) тренирана на различитим скуповима података (о томе више у наставку). Потребно је нагласити да сама имплементација овог модела јако варира од верзије до верзије радног оквира а постоји и „лакша“ верзија која се користи за праћење шаке на слабијем хардверу (телефону, таблету...).

Аутори овог рада наводе да је најизазовнији дио овог пројекта представлјао управо креирање погодног скупа података који би омогућио квалитетно обучавање горе поменутих модела. Прикупљени подаци били су организовани у 3 различита скупа података који су адресирали различите аспекте проблемског домена:

1. In-the-wild скуп података који је садржао око 6000 слика са веома великом варијансом (географски диверзитет, различито освјетљење, различите величине и боје шаке)
2. In-house скуп података који је садржао око 10000 слика, прикупљан у лабораторијским условима који је покрио све познате гестове које је могуће направити шаком из различитих углова. Једна од мана скупа података је што је прикупљен на свега 30 лјуди у скоро истом окружењу те нема превелике варијације у позадини и тону коже, међутим, ово није направило превелики проблем приликом обучавања.
3. Синтетички скуп података од 100000 генерисаних слика компјутерски анотираног са 21 3D кључним обележјем шаке. Пажљиво је вођено рачуна да се у скупу података нађу различити тонови коже као и различите позадине.



Примјер 2.5 Приказ неколико анотираних инстанци скупова података, слике из реалног свијета (горњи ред) и синтетичке слике (доњи ред)

Код обучавања BlazePalm модела коришћен је само in-the-wild скуп података док су се за обучавање HandLandmark модела користили сви скупови података и то:

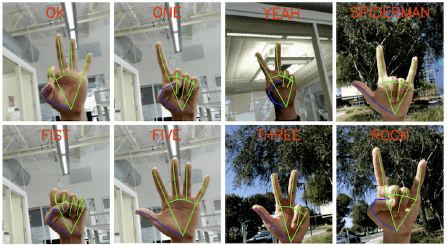
* Ручно анотирани in-the-wild скуп заједно са синтетичким скупом за обучавање главе за одређивање 21 кључног обележја
* Подскуп in-the-wild скупа заједно са In-house скупом за обучавање главе за детекцију присуства шаке као и главе за одређивање да ли је шака лијева или десна

Такође је битно напоменути да су се за евалуацију модела (и валидацију и тест) користили само подаци прикупљени из реалног свијета како би се дала што реалнија слика о перформансама модела. Приликом евалуације коришћена је average precision (AP) мјера перформансе за детекцију длана i mean squared error (MSE) нормализован према величини шаке за евалуацију квалитета локализације кључних обележја. Код детекције длана, аутори су пробали различите оптимизације са и без употребе декодера као и употребе различитих loss функција. Добијене резултате можете погледати у табели 2.1.

|  |  |
| --- | --- |
| **Варијација модела** | **Average Precision (AP)** |
| Без декодера + cross entropy loss | 86.22% |
| Декодер + cross entropy loss | 94.07% |
| Декодер + focal loss | 95.7% |

Табела 2.1 Добијени резултати детекције длана са употребом различитих оптимизација

Што се тиче евалуације квалитета локализације кључних обележја шаке, резултати су били веома добри гдје је нормализована MSE износила свега 16.1%. Резултати евалуације предикције да ли је шака на слици као и класификације шке на лијеву и десну нису дискутовани у овом раду. Исти аутори у раду [] демонстрирају како се овај модел може користити као основа за класификацију геста шаке. Предложено је да се користи стандардна потпуно повезана неуронска мрежа са 5 потпуно повезаних слојева са 64, 128, 512, 64, 32 неурона респективно, заједно са ReLu активационом функцијом и softmax loss-ом као функцијом грешке. Резултати овог модела нису изнијети у раду али је дата слика која показује да овај приступ заправо ради што ме је и заинтересовало да детаљно експериментишем у овом пољу.



Примјер 2.6 Демонстрација рада комплетног модела за праћење шаке и класификацију геста

# TEORIJSKI POJMOVI I DEFINICIJE

**Cilj poglavlja**: predstaviti teorijske osnove neophodne za razumevanje rešenja koje ste primenili u svom diplomskom radu.

**Važno:** Napravite jasno razgraničenje između pojašnjenja algoritama koje upotrebljavate (ovo poglavlje) i načina na koje upotrebljavate ove algoritme zarad konstrukcije vašeg rešenja (poglavlje 1).

Ako delite ovo poglavlje na više celina (koje odgovaraju algoritmima koje opisujete), ovde u uvodu poglavlja opišite šta je cilj poglavlja i najavite podelu na potpoglavlja (u potpoglavlju 4.1 opisan je..., a u potpoglavlju 4.2...).

|  |
| --- |
| Metod mašinskog učenja korišćen za detekciju i prepoznavanje saobraćajnih znakova u ovom radu je konvoluciona neuronska mreža. U ovom poglavlju će prvo biti objašnjene klasične veštačke neuronske mreže (poglavlje 3.1) koje predstavljaju osnovu za razumevanje konvolucionih neuronskih mreža, predstavljenih u poglavlju 3.2.   * 1. Veštačke neuronske mreže   2. Konvolucione neuronske mreže |

Primer 4.1 Primer podele poglavlja „Teorijski pojmovi i definicije“

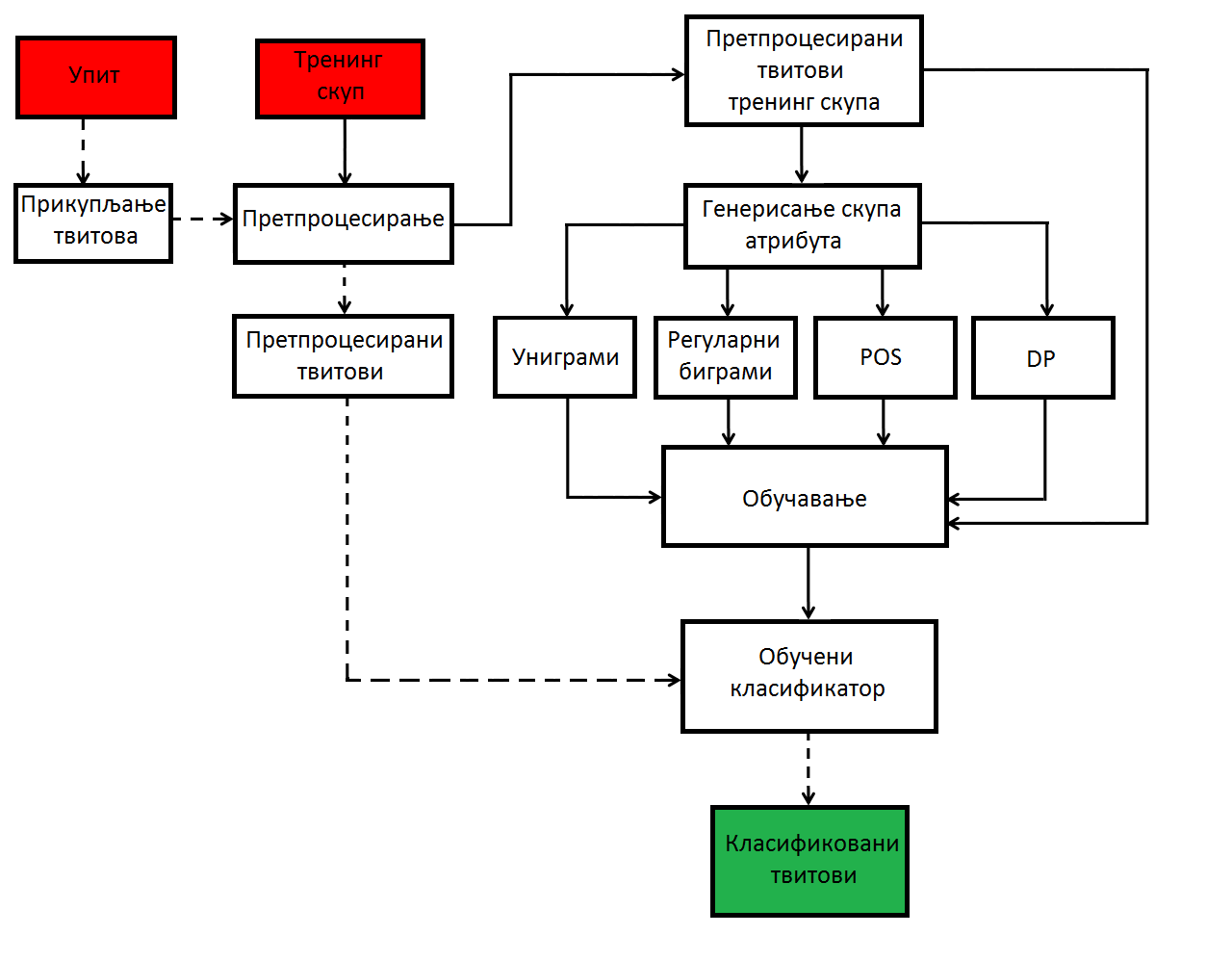
## Opis algoritma na koji se vaš rad oslanja

## Opis algoritma na koji se vaš rad oslanja

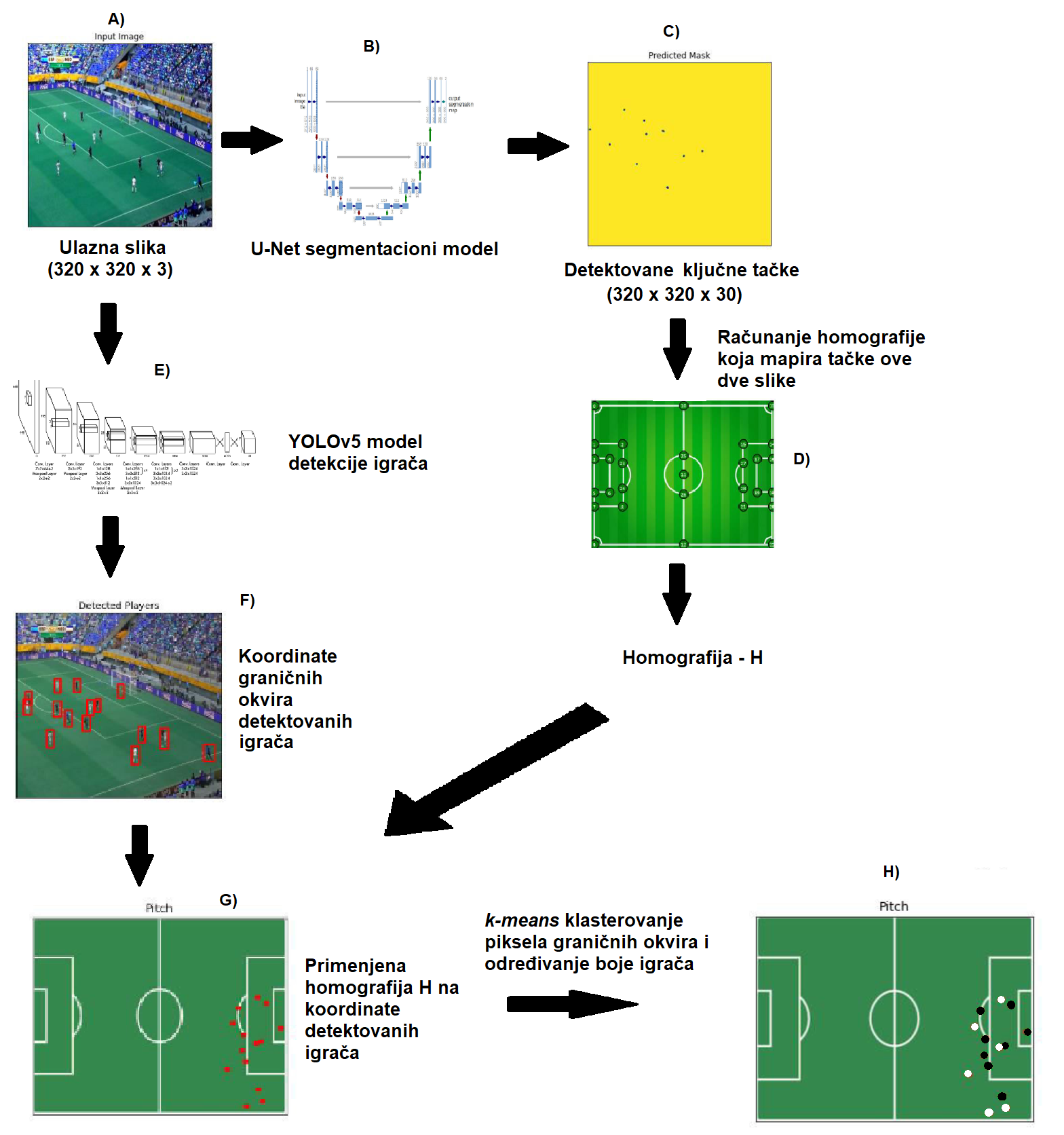
# 

# METODOLOGIJA

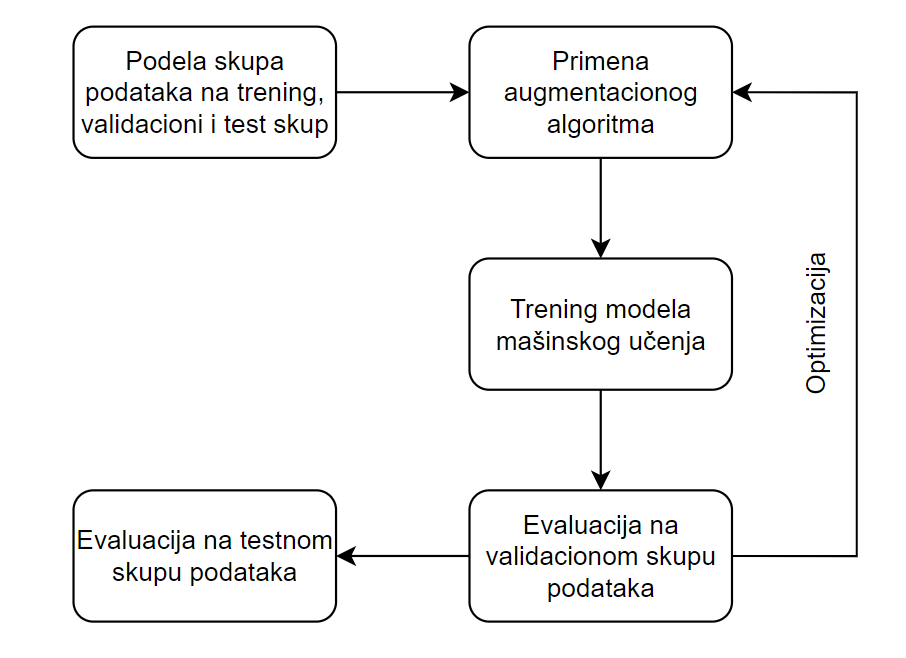
* Krenite sa pasusom koji navodi šta je prikazano u ovom poglavlju. Na primer, „U ovom poglavlju je predstavljena implementacija sistema za...“. Jasno istaknite šta je ulaz u sistem, a šta očekivani izlaz iz sistema.
* U narednom pasusu predstavite arhitekturu rešenja na visokom nivou apstrakcije (npr. dijagram toka podataka u sistemu). Primeri 5.1 - 5.4 daju ideje za predstavljanje arhitekture rešenja. Ovaj dijagram toka podataka podelite na module. Svaki modul razradite u tekstu i, ako je potrebno, prikažite i njegove delove grafički.
* Pojedinačni moduli zahtevaju opsežna objašnjenja, poglavlje „Metodologija“ možete izdeliti na potpoglavlja koja odgovaraju datim modulima. Ako ovo radite, ne zaboravite da dodate rečenicu koja najavljuje ta potpoglavlja.
* Prilikom opisa rešenja ne zaboravite da opišete sledeće stvari (ako ste ih primenili):
  + Kako ste pretprocesirali podatke
  + Kako ste vršili augmentaciju podataka
  + Arhitekturu neuronske mreže koju ste koristili uz obrazloženje zašto ste je odabrali
  + Pomoću koje funkcije gubitka se optimizuje model
  + Koje ste sve kombinacije model/pretprocesiranje isprobali
  + Na koji način ste u svoje rešenje uključili tehnologije predstavljene u poglavlju 4



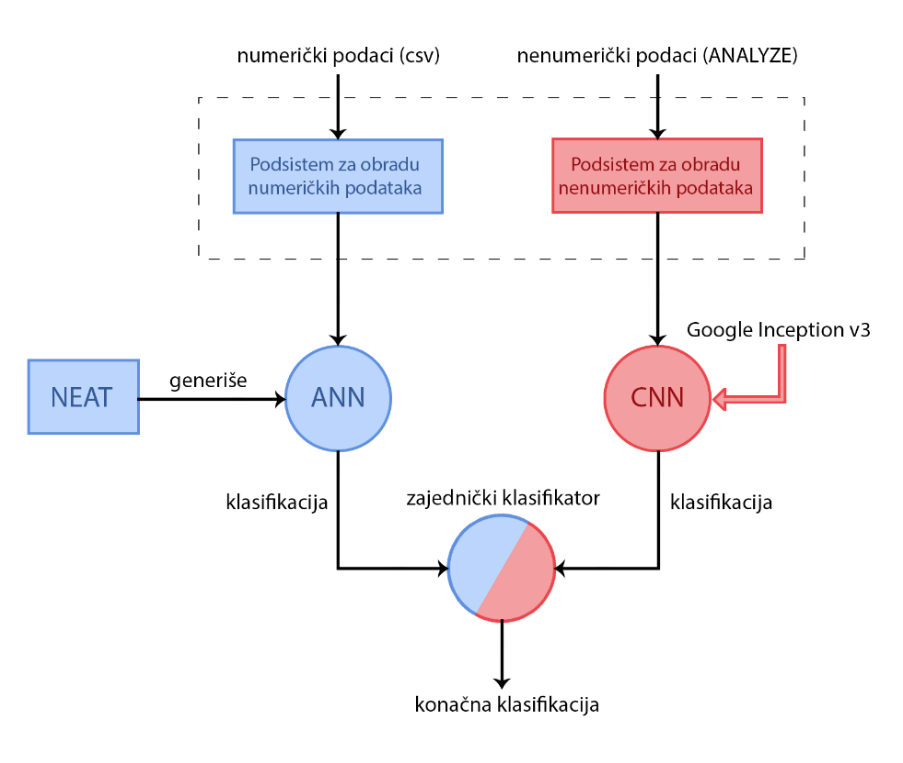
Primer 5.1 Šematski prikaz postupka klasifikacije tvitova.



Primer 5.2 Ulazna slika (A) se prosleđuje U-Net modelu (B) koji za izlaz daje matricu koja predstavlja detektovane tačke koje su vizualizovane na slici (C). Mapiranjem tačaka na njihove ciljne koordinate sa slike (D), dobijamo homografiju H. Na sliku (A) se primenjuje YOLO (E) model za detekciju igrača (F). Primenom homografije na igrače dobijamo sliku (G). K-*means* klasterovanjem piksela dobijamo boje igrača za konačni prikaz (H).



Primer 5.3 Dijagram toka eksperimenta.



5.4 Klasifikacija pacijenata na osnovu: (1) kliničkih testova kojima se detektuje demencija (numerički podaci) i (2) MRI snimaka pacijenta (nenumerički podaci).

## Prvi modul sistema (izmeniti naslov da bude specifičan za vaše rešenje)

* Krenite od toga šta je cilj modula, koji su očekivani izlazi i ulazi.
* Nakon toga specificirajte kako je ovaj cilj implementiran.

## Drugi modul sistema

## Korišćeni alati

Ovo je opciono potpoglavlje koje specificira hardver, biblioteke, alate i programske jezike na koje ste se oslanjali prilikom implementacije rešenja. Alternativno (ako nema puno sadržaja), detalje implementacije možete integrisati u odgovarajuća poglavlja koja opisuju module sistema.

# EKSPERIMENTI

**Cilj poglavlja**: U ovom poglavlju potrebno je opisati sve eksperimente koji su izvršeni u okviru diplomskog rada.

**Važno:** Postavka svakog eksperimenta trebalo bi da sadrži sve detalje koji su potrebni da bi neko drugi mogao da reprodukuje eksperiment.

**Razlika između poglavlja “Eksperimenti” i „Metodologija“**: u metodologiji opisujete arhitekturu sistema koji predstavljate u radu i detalje svakog od njegovih delova modula. Nasuprot toga, u ovom poglavlju opisujete na koji ste način(e) evaluirali taj sistem i/ili svaki od njegovih delova. Na primer, sistem ste možda evaluirali na više različitih skupova podataka ili sa više različitih kombinacija parametara ili ste hteli da vidite uticaj pred ili post-procesiranja pa ste evaluaciju radili sa i bez tih komponenti, itd.

**Organizacija:** U glavnom delu iskažite cilj poglavlja „Eksperimenti“ i pojasnite njegovu podelu na potpoglavlja koja je, tipično, ovakva:

1. Obično, eksperimenti u AI oblasti uključuju jedan ili više obučavajućih skupova. Predstavite ih u posebnom potpoglavlju „Skupovi podataka“.
2. Tipičan sledeći deo eksperimentalne postavke je opis jednog ili više modela mašinskog učenja. Napravite posebno potpoglavlje za svaki eksperiment koji ste izvršili, npr. „Eksperiment 1“, „Eksperiment 2“ (predložite specifičnija imena ako imaju smisla).
3. Opis postupka evaluacije.

**Napomena:** trudite se da ne ponavljate informacije – prvo utvrdite šta je zajedničko za sve vaše eksperimente. Tipično su to skupovi podataka i evaluacija, pa onda možete izdvojiti u dva potpoglavlja, dok ostale detalje pišete za svaki eksperiment posebno u potpoglavljima „Eksperiment 1“, „Eksperiment 2“, itd.

## Skup(ovi) podataka

**Cilj potpoglavlja:** opis skupova podataka korišćenih u svim eksperimentima.

**Organizacija:** Ako imate puno skupova podataka, možete ovo poglavlje podeliti na potpoglavlja koja im odgovaraju. Ako skupovi podataka imaju duža imena, u poglavlju 6.1 uvedite skraćenice. Na primer, K1 (korpus 1).

**Tipični podaci koje opis skupa podataka treba da sadrži:**

1. količina podataka,
2. distribucija klasnog obeležaja,
3. detalji vezani za prikupljanje i anotaciju (labeliranje),
4. da li ste skup podataka preuzeli (da li je javno dostupan, odakle ste ga preuzeli) ili sami kreirali za potrebe diplomskog rada,
5. detalji specifični za vaš skup podataka:
   * Na primer, ako su u pitanju slike onda: veličina, rezolucija, format, kvalitet slika, itd.
   * Na primer, ako je u pitanju tekst onda: prosečna dužina rečenica, dokumenata, entiteta, itd.
6. detalji skupa podataka značajni za problem koji rešavate:
   * Razmišljajte o tome koji detalj skupa vam je potreban da biste u diskusiji objasnili neki rezultat eksperimenta.
   * Na primer, za slike, rezultati mogu biti loši jer je rezolucija slika niska (i onda kod tih rezultata referencirate ovo poglavlje).
   * Na primer, za tekst, rezultati mogu biti loši jer je određena vrste entiteta slabo zastupljena u obučavajućem skupu, rečenice su previše dugačke, itd.

## Eksperiment 1

**Cilj:** prikazujete detalje vezane za ML model(e). Sam model je opisan u prethodnim poglavljima („Teorijske osnove“ i/ili „Metodologija“), dok je ovde cilj da opišete detalje koje ste podesili baš za eksperiment sa ciljem da eksperiment može da se kompletno reprodukuje.

Na primer, ovde prikazujete:

* Vrednosti hiper-parametara. Ako ih ima puno, navedite ih u tabeli.
* Način na koji ste odredili vrednosti hiper-parametara.
* Ako u vašem eksperimentu poredite više modela, onda ovde opisujete konkretne postavke za svaki takav model.
* Ako je cilj vašeg eksperimenta da utvrdite optimalne hiper-parametre ili skup atributa, onda je fokus na opisu načina na koji to radite.
* Hardver ili *cloud* servis koji je korišćen, itd.

## Evaluacija

**Cilj:** prikaz načina evaluacije rešenja. Tipično, ovo obuhvata:

1. Kako ste formirali skup podataka za evaluaciju:

* kako je formiran test skup,
* odnos udela podataka u *test/train* podeli,
* koja vrsta uzorkovanja (*sampling*) je upotrebljena pri podeli na *train*/*test* (npr. *random* ili *stratified*)
* ako je upotrebljena unakrsna validacija, opišite detalje vezane za taj postupak.

1. Mere performansi koje koristite za evaluaciju. Ako su mere neuobičajene i kompleksne onda bi bilo dobro da ih detaljnije objasnite na primeru. Jedan primer takve mere je *Intersection over Union* (IoU) koja se koristi za evaluaciju sistema za detekciju objekata.

# REZULTATI (I DISKUSIJA)

**Cilj:** prikazujete rezultate eksperimenata opisanih u prethodnom poglavlju.

**Napomena oko naslova poglavlja**: ponekad eksperimenti i rezultati mogu biti obimni i/ili zanimljivi pa se o njima može napisati duža diskusija. U tom slučaju je potrebno diskusiju odvojiti u posebno poglavlje „Diskusija“ koje ide nakon rezultata. U nastavku ovog teksta opisani su detalji vezani za poglavlje „Rezultati“, pa onda za poglavlje „Diskusija“, a na vama je da odlučite da li ćete ih pisati u jednom ili u dva odvojena poglavlja.

**Sadržaj poglavlja:**

1. Ovo poglavlje tipično sadrži jednu ili više tabela sa rezultatima eksperimenata. Najbolje je da tabele prikazujete redom po eksperimentima i onda na kraju da date jednu sumarnu tabelu ako to ima smisla. Na primer, ako eksperimenti uključuju evaluaciju nekoliko različitih ML modela, onda u sumarnoj tabeli pokazujete sve njih zajedno i komentarišete kako se porede.
2. Nakon prikaza tabele napišete jedan ili više pasusa u kojima ukratko saopštite i komentarišite rezultate.
   1. Ne ponavljate celu tabelu u tekstu, već istaknite nabolje, najlošije rezultate.
   2. Komentarišite da li su ovi rezultati očekivani (u skladu sa dosadašnjom literaturom) ili iznenađujući.
   3. Na primer, model X se pokazao kao najbolji, što je i očekivano jer je to SOTA model za ovaj problem ili rezultati za klasu X su najbolji, što je iznenađujuće jer je to klasa koje jako slabo zastupljena u obučavajućem skupu.
3. Važno je da donesete odluku o tome da li ćete diskusiju rezultata uključiti u ovo poglavlje ili ćete ih predstaviti u posebnom poglavlju „Diskusija“.
   1. Ako diskusije nema puno, onda ovde dajte svoja objašnjenja za rezultate. U tom slučaju, u nastavku kao posebne pasuse možete dodati delove poglavlja „Diskusija“ (sadržaj poglavlja „Diskusija“ je objašnjen u poglavlju 8).
   2. Razloge zašto su rezultati takvi kakvi su često nije lako utvrditi. Zato, pri diskusiji pišite „pretpostavljamo da su rezultati takvi zato što...“ umesto da čvrsto „tvrdimo da su rezultati...“.

Preporuke oko predstavljanja sadržaja:

* Koristite tabele, a ne grafikone jer se iz tabela vide tačne vrednosti mera performansi. Pored tabela možete imati i grafikon koji je vizuelno predstavlja. Razmislite detaljno o tome šta je to što će on moći da prikaže, a tabela neće.
* Tabele ne bi trebalo da budu previše velike. Uvek je bolje da prikažete više manjih preglednih tabela nego jednu veliku nepreglednu.
* Podebljajte (*bold*) najbolje rezultatе u tabeli.
* Nivo detalja prikaza rezultata bi trebalo da bude što veći. Na primer, ako radite više-klasnu klasifikaciju trebalo bi da pored sumarnih mera (npr. makro F-mera) date i preciznost, odziv i F-meru za svaku klasu posebno.
* Ako imate preveliki broj klasa da biste rezultate prikazivali za svaku klasu posebno (preko 10 ili 15), onda rezultate po klasama možete opisati tako što ćete ih grupisati. Na primer, klase 1,2 i 8 imaju F-mere u rasponu od X do Y, dok klase 3,5,9 imaju....itd.
* Trudite se da sumarne mere što realnije sumiraju rezultate vašeg sistema. Na primer, za više-klasnu klasifikaciju makro F-mera je dobar izbor sumarne mere. Dakle, pored ili u okviru tabele sa rezultatima po klasama možete dati i makro F-meru kao sumarnu.

# DISKUSIJA

**Cilj:** prikazujete svoje rezultate iz svih uglova važnih za cilj vašeg rada.

* Na primer, ako je cilj vašeg rada bio da kreirate novi sistem za neki problem onda je važno da diskutujete kako se vaši rezultati porede sa srodnim rešenjima, pogotovo onim najboljim.
* Ako ne postoje srodna rešenja koja baš rešavaju isti problem kao i vi, sigurno postoje rešenja koja rešavaju sličan problem sa kojima se možete uporediti.
* Na primer, razvijate određen NLP sistem za srpski jezik i niko do sada nije radio to baš za srpski jezik. U tom slučaju, poredite se sa sistemima koji to rada za engleski.

**Sadržaj poglavlja**:

* Komentarišite poređenja sa drugim sistemima.
  + Na primer, rezultati prikazani u radu [x] su bolji jer je korišćen bolji hardver ili veći skup podataka, što ovde nije moguće uraditi usled nedostatka *big data* infrastrukture.
  + Ako je teško porediti performanse usled korišćenja različitih skupova podataka i meri performanse, istaknite tu činjenicu.
* Komentarišite realnu upotrebljivost sistema.
  + Da li su performanse na nivou onih koje postižu ljudi?
  + Da li se rezultati direktno mogu koristiti u produkciji ili su tek prvi korak ka nekom boljem sistemu?
  + Ovde je takođe važno da komentarišete razloge za tvrdnje koje ste izneli.
  + Napomena: ponekad je samim anotatorima skupa podataka teško da se dogovore oko anotacije konkretnog primera[[1]](#footnote-1). Ovo može biti uzrok loših performansi modela. Često u srodnim rešenjima postoje poglavlja koja govore o anotaciji skupa podataka gde su date mere (tipično F-mera) koje pokazuju koliko je teško da se više ljudi složi oko anotacije. U tom slučaju se možete porediti sa tim merama.
* Diskutujte prednosti i ograničenja svog sistema.
  + Na primer, prednost je da je sistem lako proširiv, ali je trenutno ograničenje da radi samo za srpski jezik.
  + Važno je da budete realni, nemojte sakrivati nedostatke. Svrha ovog dela je upravo to da pokažete da ste svesni ograničenja svog sistema.
  + Dajte predloge kako bi se ograničenja mogla prevazići. Na primer, bolji hardver, veći skup podataka, itd.
* (Opciono) u ovo poglavlje uključite i analizu grešaka sistema[[2]](#footnote-2). Za detalje vezane za analizu grešaka pogledajte prateći materijal.

# 

# ZAKLJUČAK

**Cilj**: (1) sumiranje glavnih poenti u radu i diskusija implikacija rešenja i (2) opis mogućih daljih pravaca za unapređenje iznetog rešenja.

**Sadržaj**:

* Sumiranje glavnih poenti u radu:
  + Rešavani problem i motivacija za njegovo rešavanje
  + Opis rešenja na visokom nivou apstrakcije
  + Opis evaluacije rešenja na visokom nivou apstrakcije
  + Podsetiti na sve izvučene zaključke i komentarišite njihovu važnost (glavne poente iz diskusije)
* Iznesite moguće pravce daljeg proširivanja/unapređenja/otklanjanje identifikovanih nedostataka rešenja

|  |
| --- |
| **(Rekapitulacija problema i motivacije za njegovo rešavanje:)** U ovom radu predstavljen je sistem za automatsku detekciju i prepoznavanje saobraćajnih znakova sa slike. Motivacija je bila što bi takav sistem mogao da se koristi prilikom vožnje i da koristi vozaču putem obaveštavanja o važećim saobraćajnim znakovima. Sistem je implementiram u vidu dva odvojena dela: modula za detekciju i modula za prepoznavanje saobraćajnih znakova. Oba modula oslanjaju se na modele konvolucionih neuronskih mreža.  **(Rezime rešenja, načina evaluacije i postignutih rezultata:)**  Modul za detekciju koristi *Faster* R-CNN [4] metod sa *AlexNet* [5] arhitekturom konvolucione mreže. Rezultat evaluacije istreniranog modela je 0.6756 mAP. Za treniranje i testiranje korišćen je skup belgijskih saobraćajnih znakova (engl. BTSD – *Belgium Traffic Sign Dataset*) [6], uz izbacivanje kategorija „nedefinisano“ i „ostalo“, i smanjenje broja negativnih uzoraka.  Modul za prepoznavanje saobraćajnih znakova vrši klasifikaciju znakova dobijenih iz faze detekcije. Za klasifikaciju treniran je model konvolucione neuronske mreže sa *ResNet* [8] arhitekturom od devet rezidualnih blokova. Korišćen je skup nemačkih saobraćajnih znakova za prepoznavanje (engl. GTSRB – *German Traffic Sign Recognition Benchmark*) [9], a preciznost modela nad test skupom iznosi 96.2%.  **(Predlozi za unapređenje rešenja:)** Preciznost konvolucione mreže za prepoznavanje mogla bi se dalje povećati boljim izborom hiper-parametara modela, što bi zahtevalo ponavljanje procesa treniranja. Što se tiče *Faster* R-CNN metoda korišćenog za detekciju, značajno bolje performanse mogu se postići prelaskom na *ResNet* arhitekturu konvolucione mreže. S obzirom na to da *Faster* R-CNN implementacija u korišćenom *Microsoft Cognitive Toolkit* (CNTK) [10] alatu ovo ne podržava, to bi zahtevalo izmenu implementacije ili korišćenje drugog alata. Skup podataka bi takođe mogao da se proširi spajanjem sa nekim drugim, kao što je na primer skup nemačkih saobraćajnih znakova za detekciju (engl. GTSDB – *German Traffic Sign Detection Benchmark*) [32]. |

Primer 9.1 Primer celokupnog zaključka uz komentare

# LITERATURA

|  |
| --- |
|  |
| 1. Manresa, C., Varona, J., Mas, R. and Perales, F.J., 2005. Hand tracking and gesture recognition for human-computer interaction. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, *5*(3), pp.96-104. 2. Wang, R.Y. and Popović, J., 2009. Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, *28*(3), pp.1-8. 3. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y. and Freedman, D., 2015, April. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems* (pp. 3633-3642). 4. Melax, S., Keselman, L. and Orsten, S., 2013, March. Dynamics based 3D skeletal hand tracking. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (pp. 184-184). 5. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L. and Grundmann, M., 2020. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*. 6. Yeo, H.S., Lee, B.G. and Lim, H., 2015. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications*, *74*(8), pp.2687-2715. 7. Maung, T.H.H., 2009. Real-time hand tracking and gesture recognition system using neural networks. *International Journal of Computer and Information Engineering*, *3*(2), pp.315-319. 8. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J. and Chang, W.T., 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*. 9. Halder, A. and Tayade, A., 2021. Real-time vernacular sign language recognition using mediapipe and machine learning. *Journal homepage: www. ijrpr. com ISSN*, *2582*, p.7421. 10. Duy Khuat, B., Thai Phung, D., Thi Thu Pham, H., Ngoc Bui, A. and Tung Ngo, S., 2021, February. Vietnamese sign language detection using Mediapipe. In *2021 10th International Conference on Software and Computer Applications* (pp. 162-165). 11. Le, T.H., Jaw, D.W., Lin, I.C., Liu, H.B. and Huang, S.C., 2018, May. An efficient hand detection method based on convolutional neural network. In *2018 7th International Symposium on Next Generation Electronics (ISNE)* (pp. 1-2). IEEE. |
|  |  |

# BIOGRAFIJA

Ovde navedite osnovne informacije o sebi koje će predsednik komisije pročitati prilikom otvaranja odbrane završnog/diplomskog rada (primer 11.1 ).

|  |
| --- |
| Petar Petrović je rođen dd.mm.gggg. u Novom Sadu, gde je stekao svoje osnovno i srednje obrazovanje. Školske xxxx/yy godine se upisuje na Fakultet tehničkih nauka na studijski program X. Položio je sve ispite predviđene planom i programom i stekao uslov za odbranu završnog rada. |

Primer 11.1 Primer biografije

1. Na primer, cilj nam je da razvijemo AI model čiji je cilj da detektuje da li je određena Java klasa previše kompleksna (pati od tzv. *God Class code smell*-a). Samim programerima (anotatorima) je teško da usaglase stavove oko toga koje klase su „previše kompleksne“. Stoga skup podataka može imati nekonzistentne anotacije, čineći da obučeni ML model ima lošije performanse. [↑](#footnote-ref-1)
2. Analizu grešaka sistema možete opciono staviti i u poglavlje „Rezultati“. U tom slučaju, u diskusiji kratko prodiskutujte rezultate te analize. [↑](#footnote-ref-2)