

**Факултет инжењерских наука Универзитета у Крагујевцу**



**Назив студијског програма: Електротехника и рачунарство**

**Ниво студија: Мастер академске студије**

**Предмет: Статистичка класификација сигнала**

**Број индекса: 439/2019**

**Филип М. Секулић**

**Напредни алгоритми за препознавање  
регистарских ознака возила  
мастер рад**

**Комисија за преглед и одбрану:**

1. Др Маријана Гавриловић Божовић - ментор

2. \_\_\_\_\_

3. \_\_\_\_\_

Датум одбране: \_\_\_\_\_

Оцена: \_\_\_\_\_

У оквиру овог мастер рада кандидат треба да

Да преглед постојећих система за детекцију и препознавање регистарских ознака возила. Креира систем за аутоматску детекцију и препознавање регистарских ознака возила на основу доступних података. Резултат мастер рада је апликација која омогућава аутоматску детекцију и препознавање регистарских ознака возила.

#### Препоручена литература:

- [1] Anagnostopoulos, C.N., Anagnostopoulos, I., Psoroulas, I., Loumos, V., Kayafas, E.: License Plate Recognition From Still Images and Video Sequences: A Survey. IEEE Transactions on Intelligent Transportation Systems 9(3), 377–391 (sep 2008). <https://doi.org/10.1109/TITS.2008.922938>, <http://ieeexplore.ieee.org/document/4518951/>
- [2] Goncalves, G.R., da Silva, S.P.G., Menotti, D., Schwartz, W.R.: Benchmark for license plate character segmentation. Journal of Electronic Imaging 25(5), 1–5 (2016), <http://www.ssig.dcc.ufmg.br/wp-content/uploads/2016/11/JEI-2016-Benchmark.pdf>
- [3] Bulan, O., Kozitsky, V., Ramesh, P., Shreve, M.: Segmentation- and AnnotationFree License Plate Recognition With Deep Localization and Failure Identification. IEEE Transactions on Intelligent Transportation Systems 18(9), 2351–2363 (sep 2017). <https://doi.org/10.1109/TITS.2016.2639020>

Крагујевац, 30.04.2020.

Ментор:

Др Маријана Гавриловић Божовић – доцент

---

# Садржај

|                                                                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. Увод .....                                                                                                                                       | 6  |
| 2. Развој система за препознавање регистарских ознака таблица кроз историју .....                                                                   | 7  |
| 3. Технологије детекције и препознавања регистарских ознака возила .....                                                                            | 10 |
| 3.1 Алгоритми коришћени у реализацији система за препознавање регистарских ознака возила.....                                                       | 11 |
| 3.2 Потешкоће у реализацији алгоритама .....                                                                                                        | 12 |
| 3.3 Технологије коришћене за израду апликације система .....                                                                                        | 14 |
| 3.3.1 <i>Python</i> .....                                                                                                                           | 14 |
| 3.3.2 <i>OpenCV</i> .....                                                                                                                           | 14 |
| 3.3.3 <i>Tesseract (OCR)</i> .....                                                                                                                  | 14 |
| 3.3.4 <i>Tkinter</i> .....                                                                                                                          | 15 |
| 4. Апликација за препознавање регистарских ознака возила.....                                                                                       | 16 |
| 4.1 Основи слике .....                                                                                                                              | 16 |
| 4.2 Опис и примена алгорита за препознавање регистарских ознака возила имплементираног без утицаја алгорита к најближих суседа ( <i>KNN</i> ) ..... | 17 |
| 4.2.1 Конверзија слике у боји у сиву слику .....                                                                                                    | 18 |
| 4.2.2 Билатерални филтер .....                                                                                                                      | 19 |
| 4.2.3 <i>Canny edges</i> .....                                                                                                                      | 20 |
| 4.2.4 Проналажење и исцртавање контура на основу ивица .....                                                                                        | 22 |
| 4.2.5 Сортирање контура.....                                                                                                                        | 24 |
| 4.2.6 Пролазак кроз контуре и проналазак одговарајуће .....                                                                                         | 24 |
| 4.2.7 Конвертовање слике у текст.....                                                                                                               | 25 |
| 4.3 Опис и примена алгорита за препознавање регистарских ознака возила имплементираног уз утицај алгорита к најближих суседа ( <i>KNN</i> ) .....   | 27 |
| 5. Резултати рада апликације за препознавање регистарских ознака возила .....                                                                       | 30 |
| 5.1 Успешни резултати рада апликације применом оба начина .....                                                                                     | 32 |
| 5.1.1 Успешни резултати без употребе алгорита к најближих суседа .....                                                                              | 32 |
| 5.1.2 Успешни резултати уз употребу алгорита к најближих суседа.....                                                                                | 36 |

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| 5.2 Неуспешни резултати рада апликације применом оба начина.....          | 38 |
| 5.2.1 Неуспешни резултати без употребе алгоритма к најближих суседа ..... | 38 |
| 5.2.2 Неуспешни резултати уз употребу алгоритма к најближих суседа .....  | 42 |
| 5.3 Поређење резултата рада апликације применом оба начина.....           | 44 |
| 6. Изазови, контроверзе и проблеми .....                                  | 46 |
| 6.1 Обмане .....                                                          | 46 |
| 6.2 Контроверзе .....                                                     | 47 |
| 6.3 Неусклађеност регистарских таблица и разлике у надлежности .....      | 49 |
| 7. Закључак .....                                                         | 51 |
| 8. Литература.....                                                        | 53 |

## Summary

This master thesis presents an application for license plate recognition based on the use of algorithms for license plate detection and recognition as well as the Python programming language's libraries in which the application code for this system was written. A detailed description of the algorithms is given, as well as the experimental results aimed at examining the influence of individual parameters on the behavior of the system.

**Key words:** license plate recognition, edge detectors, license plate localization, optical character recognition, license plate segmentation.

## Резиме

Овим мастер радом је представљена апликација за препознавање регистарских ознака возила базирана на употреби алгоритама за детекцију и препознавање регистарских ознака као и библиотека програмског језика *Python* у коме је написан код апликације за овај систем. Дат је детаљан опис алгоритама, као и експериментални резултати са циљем испитивања утицаја појединих параметара на понашање система.

**Кључне речи:** препознавање регистарских таблица, детектори ивица, локализација регистарске табlice, оптичко препознавање карактера, сегментација регистарске табlice.

# 1. Увод

Једни од озбиљнијих проблема у области рачунарског вида су алгоритми за препознавање регистарских ознака возила на фотографијама као и директно преко видео извора. Одређивање где се тачно налази регистарска ознака на слици је главни предуслов за одвијање сложенијих задатака као што су препознавање регистарских ознака возила и прављење сложених 3Д анимација. На сликама се може наћи много објеката који нису регистарске ознаке (људи, дрвеће, животиње, саобраћајни знакови, итд.) и из тог разлога је неопходно прво детектовати регистарску ознаку возила, како би се касније та регистарска ознака обрадила у процесу препознавања.

Препознавање регистарских ознака возила је рачунарска технологија која се користи у разним апликацијама које идентификују регистарске ознаке возила на сликама. Такође се односи на психолошки процес којим људи проналазе регистарске ознаке возила у визуелном приказу. Алгоритми за препознавање регистарских ознака возила се фокусирају на детекцију фронталног дела регистарских ознака и обрадом одређених параметара дају одговарајуће резултате свога рада.

Препознавање регистарских ознака возила представља технологију која треба да идентификује или верификује регистарску ознаку са слике или видео оквира из видео извора. Постоји више метода у којима раде системи за препознавање регистарских ознака возила, али генерално, они раде упоређујући одабране карактеристике регистарских ознака са дате слике са регистарским ознакама возила унутар базе података. Технологија препознавања регистарских ознака возила је заснована на биометријској вештачкој интелигенцији која може јединствено идентификовати регистарску ознаку анализом образаца заснованих на текстури и облику регистарске ознаке.

У овом мастер раду приказан је једноставан и ефикасан поступак издвајања обележја регистарских таблица у циљу даље идентификације возила, који би се могао користити у оквиру система за аутоматско препознавање регистарских таблица (*ANPR – Automatic Number Plate Recognition*). Алгоритам обухвата локализацију и сегментацију таблице, као и издвајање прочитаних карактера, а реализован је у *Python* програмском језику. Алгоритам укључује и предобраду слике, уклањање шума, корекцију сенке и одсјаја, као и корекцију величине улазне слике, са циљем смањења њене величине због оптимизације трајања процеса обраде података. Након одговарајуће предобrade и примене предложеног поступка, са великом тачношћу се добијају сегментирани карактери, применом неког од стандардних поступака за оптичко препознавање карактера (*OCR – Optical Character Recognition*).

У другом поглављу је описан развој система за препознавање регистарских ознака возила кроз историју, док је већ у трећем поглављу то детаљније описано, за сваку технологију понаособ. Детаљнији опис алгоритама за препознавање регистарских ознака возила као и начин на који они функционишу у овом раду је описан у четвртном поглављу, а резултати рада апликације за препознавање регистарских ознака возила са експерименталним резултатима у петом поглављу.

## 2. Развој система за препознавање регистарских ознака таблица кроз историју

Први системи за препознавање регистарских ознака таблица јављају се 80-тих година за потребе идентификације возила у контроли саобраћаја, на царинама, наплатним рампама, у случају потраге за украденим колима, да би 90-тих година доживели своју комерцијалну употребу. Данас, широм света постоје системи за аутоматско препознавање регистарских ознака таблица који се свакодневно користе, као што су: *SeeCar* у Израелу, *VECON* у Хонгконгу, *LPR* у Америци, *IMPS* у Сингапуру, *CARINA* у Мађарској, итд. Сваки од наведених система састоји се од непокретне камере (у боји, или сива), хватача оквира (фрејма), рачунара и специјалног софтвера за обраду и анализу слике. Сам систем треба да ради у разноврсним условима и најчешће се састоји из три дела [1]:

1. локализација таблице,
2. сегментација карактера,
3. читавање карактера.

Систем за препознавање регистарских ознака таблица је изумљен 1976. године у огранку за научни развој полиције у Великој Британији. Прототипни системи су радили до 1979. године, а додељени су уговори за производњу индустријских система, прво у *EMI Electronics*, а затим у *Computer Recognition Systems (CRS)*, сада део *Jenoptik-a* у Вокингаму, у Великој Британији. Први системи постављени су на путу A1 и у тунелу Дартфорд. Прво хапшење откривањем украденог аутомобила извршено је 1981. године. Међутим, систем за препознавање регистарских ознака таблица се није широко користио све док нови развој јефтинијег и једноставнијег софтвера није уведен током 1990-их. Прикупљање података за будућу употребу, тј. у решавању тада неидентификованих злочина документовано је почетком 2000-их.

Први документовани случај коришћења система за препознавање регистарских ознака таблица за помоћ у решавању убиства догодио се у новембру 2005. године у Бредфорду, у Великој Британији, где је систем за препознавање регистарских ознака таблица играо виталну улогу у проналажењу и накнадној осуди убице Шерон Бешенивски.

Данашњи системи за препознавање регистарских ознака су модернизовани и могу се видети у многим градовима широм света, па и код нас у Србији (Око соколово – слика 1).



Слика 1. Око соколово (Србија) [13]



Слика 2. Опрема полиције у Дубаију [14]

Ова технологија данас је позаната и по другим интернационалним називима, као што су:

- *Automatic (or automated) license-plate recognition (ALPR)*
- *Automatic (or automated) license-plate reader (ALPR)*
- *Automatic vehicle identification (AVI)*
- *Automatisk nummerpladegenkendelse (ANPG)*
- *Car plate recognition (CPR)*
- *License plate recognition (LPR)*
- *Lecture automatique de plaques d'immatriculation (LAPI)*

- *Mobile license-plate reader (MLPR)*
- *Vehicle license-plate recognition (VLPR)*
- *Vehicle recognition identification (VRI)*

### 3. Технологије детекције и препознавања регистарских ознака возила

Алгоритми који се користе у овим системима укључују детекцију подручја регистарске таблице и сегментацију селектованог подручја, а у циљу издвајања и препознавања знакова на таблици велики значај свакако има и примена алгоритама за детекцију ивице у дигиталној слици. У циљу смањења утицаја услова осветљења, као и утицаја временских услова на поузданост ових алгоритама, пожељно је користити слике високог квалитета.

Свакако је избор адекватне камере важан фактор за правилно функционисање комплетног система, мада у оквиру овог рада нису коришћене професионалне висококвалитетне камере већ је рад базиран на унапред експортираним сликама. Улазне слике на које је примењен предложени алгоритам снимљене су у условима дневне светлости, као и ноћу, под претпоставком да се возило не креће, или се креће малом брзином.

Алгоритми приказани у овом раду прилагођени су идентификацији обележја регистарских таблица које се састоје из јасно уочљивих знакова. Алгоритам издвајања обележја регистарских таблица имплементиран у овом раду, обухвата локализацију и сегментацију таблице, као и издвајање прочитаних карактера, а реализован је у *Python* програмском језику. Након одговарајуће предобраде и обраде слике, добијени сегментирани карактери приказани су употребом бесплатних алата за препознавање текста са слике (*OCR – Optical Character Recognition*). Имплементација додатних могућности претраге видео архиве, као и приказ у реалном времену на неком стандардном клијентском модулу значајно би допринели квалитету апликативности предложеног алгоритма [2].

Како би алгоритам могао бити успешно имплементиран и реализован, неопходно је да карактери на регистарској таблици возила буду јасно уочљиви, а такође је понекад битан и сам фонт. На слици 3 је приказан пример фонта који се тренутно користи на холандским таблицама и који је морао бити промењен управо из разлога успешности примене алгоритма. Систем користи оптичко препознавање знакова (*OCR*) на сликама. Када су се холандске регистарске таблице 2002. године пребациле на другачији стил, једна од промена је извршена у фонту, уводећи мале празнине у неким словима (као што су *P* и *R*) како би их учинили јаснијим и стога читљивијим за такве системе. Неки аранжмани регистарских таблица користе варијације у величини фонта и позиционирању - системи морају бити у стању да се носе са таквим разликама да би били заиста ефикасни. Компликованији системи могу се носити са међународним варијантама, иако су многи програми прилагођени свакој земљи појединачно [3].



Слика 3. Фонт на возилима са холандским регистарским ознакама [15]

### **3.1 Алгоритми коришћени у реализацији система за препознавање регистарских ознака возила**

Постоји седам примарних алгоритама које систем захтева за идентификацију регистарске таблице [4]:

1. Локализација регистарске таблице - одговорна за проналажење и изоловање регистарске таблице на слици
2. Оријентација регистарске таблице и димензионисање - надокнађује искривљеност регистарске таблице и прилагођава димензије потребној величини
3. Нормализација - подешава осветљеност и контраст слике
4. Сегментација карактера - проналази појединачне карактере на регистарској таблици
5. Оптичко препознавање карактера
6. Синтаксичка/геометријска анализа - провера карактера и положаја
7. Апроксимација препознате вредности у више поља/слика да би се добио поузданији или сигурнији резултат, поготово што било која појединачна слика може садржати одбијену светлост, бити делимично заклоњена, или имати привремени ефекат
8. Алгоритам к најближих суседа

Сложеност сваког од ових подеделака програма одређује тачност система. Током треће фазе (нормализација), неки системи користе технике откривања ивица да би повећали разлику у слици између слова и подлоге регистарске таблице. Средњи филтер се такође може користити за смањење визуелних шума на слици [5].



Слика 4. Кораци 2, 3 и 4: регистарска таблица је нормализована за осветљеност и контраст, а затим се карактери сегментирају како би били спремни за OCR [16]

## 3.2 Потешкоће у реализацији алгоритама

Постоји низ могућих потешкоћа са којима софтвер мора да се носи. То укључује [6]:

- Лошу резолуцију датотеке, обично зато што је регистарска таблица предалеко, али понекад је последица употребе неквалитетне камере
- Мутне слике, посебно замућене у покрету
- Лоше осветљење и низак контраст због прекомерне експозиције, рефлексije, или сенки
- Предмет који заклања (део) регистарске таблице, често вучна кука или прљавштина на регистарској таблици
- Очитавање регистарских таблица које се разликују напред и позади због приколица, кампера, итд.
- Промена траке возила у углу гледања камере током очитавања регистарске таблице
- Другачији фонт карактера на регистарским таблицама
- Технике заобилажења

- Недостатак координације између држава - два возила из различитих земаља или држава могу имати исти број, али различит дизајн регистарске таблице



Слика 5. Изглед регистарске таблице у Великој Британији 1970-их година [17]

Рани системи нису могли да читају бела или сребрна слова на црној позадини, као што је било дозвољено на возилима у Великој Британији произведеним пре 1973. године

Иако се неки од ових проблема могу исправити у оквиру софтвера, пре свега је препуштено хардверској страни система да пронађе решења за ове потешкоће. Повећавањем висине камере може се избећи проблем са предметима (као што су друга возила) који заклањају регистарску таблицу, али уводи и повећава друге проблеме, као што је подешавање за повећану закривљеност регистарске таблице [7].

На неким аутомобилима шипке за вучу могу заклонити један или два знака регистарске таблице. Бицикли на носачима за бицикле такође могу да заклоне регистарску таблицу. Неки мали системи дозвољавају одређене грешке на регистарским таблицама. Када се систем користи за омогућавање приступа одређеним возилима у неприступачном подручју, може се донети одлука да прихватљива стопа грешке износи један знак. То је зато што је вероватноћа да неовлашћени аутомобил има такву сличну регистарску таблицу прилично мала. Међутим, овај ниво непрецизности не би био прихватљив у већини примена овог система.

## 3.3 Технологије коришћене за израду апликације система

### 3.3.1 *Python*

Пајтон (*Python*) је програмски језик високог нивоа опште намене. Подржава, у првом реду императивни, објектно-оријентисан и функционални стил програмирања.

Синтакса језика Пајтон омогућава писање веома прегледних програма. Језик се брзо и лако учи. Програми писани у Пајтон језику се најчешће интерпретирају. Уз интерпретатор се обично испоручује и веома развијена стандардна библиотека модула.

Аутор овог језика је Гвидо ван Росум са Универзитета Стичинг у Холандији.

### 3.3.2 *OpenCV*

*OpenCV (Open Source Computer Vision)* је библиотека програмских функција која садржи преко 2000 оптимизованих алгоритама, углавном у реалном времену, првобитно развијена у Интеловом истраживачком центру у Нижњем Новгороду (Русија), раније подржавана од стране *Willow Garage*, а сада подржана од стране *Itseez*. Библиотека је *cross-platform*, омогућено је коришћење библиотеке на различитим софтверима, бесплатна је за коришћење под *open-source BSD* лиценцом.

*OpenCV* је написан у језику *C++* и његов главни интерфејс је у *C++*, али и даље садржи мање разумљив, старији, проширенији *C* интерфејс. Постоје везивања у *Python*-у, *Java*-и и у *MATLAB*-у. *API* за ове интерфејсе се могу наћи на интернету.

Све нове технологије и алгоритми у *OpenCV*-у развијају се у интерфејсу *C++*.

### 3.3.3 *Tesseract (OCR)*

*Tesseract* је оптички механизам за препознавање знакова за различите оперативне системе. То је бесплатан софтвер, издат под лиценцом *Apache*. Првобитно га је развио *Hewlett-Packard* као власнички софтвер 1980-их, објављен је као отворен извор 2005. године, а *Google* га спонзорише од 2006.

2006. године, *Tesseract* се сматрао једном од најтачнијих *OCR* технологија отвореног кода које су тада биле доступне.

Ова технологија првобитно је развијена као власнички софтвер у лабораторијама *Hewlett-Packard* у Бристолу, у Енглеској и у Колораду, између 1985. и 1994. године, са још

неким променама направљеним 1996. године за пренос на *Windows*. Од тада је сав код претворен да се барем компајлира са *C++* компајлером. Потом су га 2005. године објавили као отворени извор *Hewlett-Packard* и Универзитет Невада у Лас Вегасу.

### **3.3.4 Tkinter**

*Tkinter* је библиотека у Пајтону која омогућава коришћење функција које пружају могућност креирања графичког корисничког интерфејса (*GUI*). Помоћу ове библиотеке је креиран графички кориснички интерфејс који омогућава кориснику да манипулише апликацијом.

## 4. Апликација за препознавање регистарских ознака возила

### 4.1 Основи слике

Апликација која је резултат овог мастер рада функционише тако што се у програм учита постојећа слика из рачунара, па се у следећем кораку детектује и препознаје регистарска ознака возила на фотографији. Све то је омогућено коришћењем *OpenCV* библиотеке, *Python* програмског језика и технологије за препознавање карактера на слици – *Tesseract (OCR)*.

С обзиром да се у овом програму користе слике, неопходно је прво објаснити шта чини слику и где се складиште информације о слици. Најмањи елемент слике се назива пиксел и представља једну тачку слике. Пиксели фотографије су распоређени у редове и колоне, а њихов број представља њену резолуцију. Пиксели слика у боји су представљењи RGB моделом (Red Green Blue). Сваки пиксел је комбинација те три боје и дефинисан је са три броја, који одговарају количини црвене, зелене и плаве боје садржане у тој тачки. Један пиксел црно-беле фотографије је представљен једним бројем (који представља вредност пиксела), а који изражава осветљеност и тиме нијансира опсег између црне и беле боје. Вредности количине боје присутне у пикселу крећу се од 0 до 255.

Свака слика садржи информације у форми одлика. Једна одлика може бити само један пиксел, али може бити објекат или ивица између два објекта на слици. Више одлика се може спојити у сложене одлике. Рад са сликом у ствари представља рад са одликама, а коришћење постојећих и изведених одлика се користи за разне примене у области обраде слике и препознавања објеката.

Сума пиксела одређеног региона слике се добија тако што се вредности пиксела горње-леве и доње-десне ивице региона саберу, а затим се од њих одузме збир вредности пиксела горње-десне и доње-леве ивице.

## 4.2 Опис и примена алгоритма за препознавање регистарских ознака возила имплементираних без утицаја алгоритма к најближих суседа (KNN)

Први корак ка успешно развијеном систему за препознавање регистарских ознака возила представља детекција таблица. На излазу алгоритма за детекцију регистарске таблице налази се најбољи „кандидат“ са листе неколико могућих „кандидата“ за регистарску таблицу, одређених претходном обрадом. Уколико су критеријуми за одређивање правог кандидата исправни и систем не уноси грешку, на крају се добија тражена таблица која се користи као улаз алгоритма за сегментацију карактера.

Препроцесирање има за задатак да побољша слику и олакша њену анализу. Од могућности и особина фотоапарата којим су начињене слике зависи квалитет. Коришћењем класичног дигиталног фотоапарата, слике које се добијају могу имати разне мане: одсјај, сенка, итд., па се стога у наменској употреби чешће користе тзв. РЕГ камере, које поседују специјалну оптику, сочива, инфрацрвене сензоре и процесоре за дате намене. У реализованом програму који је представљен у овом раду било је потребно смањити величину улазне слике, с обзиром на резолуцију добијених снимака, како би се убрзао сам процес обраде слике [8]. Узимајући у обзир стварне услове, постоји много сметњи у оригиналним сликама, попут величине слике, осветљења и квалитета слике, које озбиљно утичу на перформансе препознавања. Да би се регистарска таблица лоцирала брзо и тачно, потребно је извршити предобраду оригиналних слика. Оригинална слика возила приказана је на слици 6.



Слика 6. Оригинална слика возила [24]

### 4.2.1 Конверзија слике у боји у сиву слику

Први корак у процесу обраде оригиналне слике јесте промена димензије тако да ширина буде 500 пиксела.

Након тога, да би се слика лакше обрадила у даљем поступку, неопходно је да се конвертује из *RGB* (слике у боји) у сиву слику, што је и приказано на слици 7.

Функција за конверзију сиве слике користи се за претварање слике у боји у сиву слику у циљу лакше обраде. Затим, с обзиром на то да се ниво сиве површине регистарске таблице очигледно не разликује од осталих подручја на слици, примењује се обрада протезања у нивоу сиве да би се појачао контраст између два дела. Након тога се скуп регија кандидата екстрактује након детекције ивица [9].



Слика 7. Сива слика

### 4.2.2 Билатерални филтер

Углавном филтери имају тенденцију да замућују ивице. То није случај са билатералним филтером који је изузетно ефикасан у уклањању шума уз очување ивица. Али, његов рад је спорији у поређењу са другим филтерима. Гаусов филтер узима суседство око пиксела и проналази његов (Гаусов) пондерисани просек. Овај Гаусов филтер је функција само простора, односно приликом филтрирања узимају се у обзир пиксели у близини. Не узима у обзир да ли пиксели имају готово исту вредност интензитета и не узима у обзир да ли пиксел лежи на ивици или не. Резултат тога је да Гаусови филтери имају тенденцију да замућују ивице, што је непожељно.

Билатерални филтер такође користи Гаусов филтер у одређеном домену, али користи и још једну (мултипликативну) компоненту Гаусовог филтера која је функција разлике у интензитету пиксела. Гаусова функција простора осигурава да се за филтрирање узимају у обзир само пиксели као „просторни суседи“, док Гаусова компонента примењена у домену интензитета (Гаусова функција разлика интензитета) осигурава да су само они пиксели са интензитетима сличним интензитету централног пиксела („суседи интензитета“) укључени у израчунавање замућене вредности интензитета. Као резултат, овај метод чува ивице, јер за пикселе који леже близу ивица, суседни пиксели постављени на другој страни ивице неће бити укључени у замућење, зато што показују велике разлике у интензитету у поређењу са централним пикселем [10].



Слика 8. Слика након коришћења билатералног филтера

### 4.2.3 Canny edges

У наредном кораку је неопходно пронаћи одговарајуће ивице за даљу обраду (*Canny edges*).

*Canny Edge Detection* је популаран алгоритам за откривање ивица. Развио га је *John F. Canny* 1986. То је вишестепени алгоритам и састоји се из следећих фаза [11]:

#### 1. Смањење шума

Будући да је откривање ивица осетљиво на шум на слици, први корак је уклањање шума на слици помоћу Гаусовог филтера.

#### 2. Проналажење градијента интензитета слике

Углађена слика се затим филтрира Собеловим зрном у хоризонталном и вертикалном смеру да би се добио први дериват у хоризонталном смеру ( $G_x$ ) и вертикалном смеру ( $G_y$ ). Из ове две слике можемо пронаћи градијент ивице и смер за сваки пиксел на следећи начин:

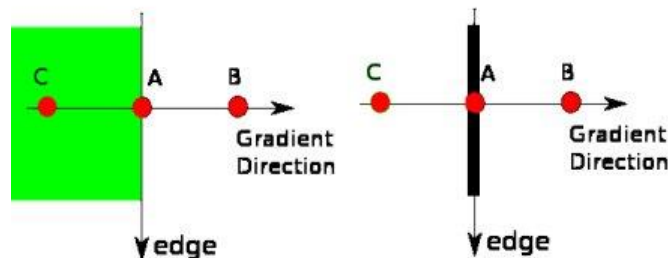
$$\text{Edge Gradient } (G) = \sqrt{G_x^2 + G_y^2}$$

$$\text{Angle } (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Смер градијента је увек окомит на ивице. Заокружен је на један од четири угла који представљају вертикални, хоризонтални и два дијагонална правца.

#### 3. Сузбијање без максимума

Након добијања величине и смера градијента, врши се потпуно скенирање слике како би се uklonили сви нежељени пиксели који можда не представљају ивицу. Зато се на сваком пикселу проверава да ли је локални максимум у његовом суседству у смеру градијента. То објашњава слика 9.

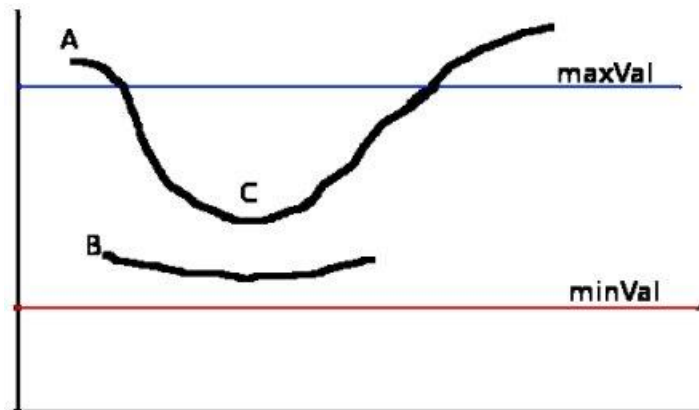


Слика 9. Провера смера градијента [18]

Тачка  $A$  је на ивици (у вертикалном смеру). Правац градијента је нормалан до ивице. Тачке  $B$  и  $C$  су у правцима градијента. Дакле, тачка  $A$  проверава са тачкама  $B$  и  $C$  да ли се формира локални максимум или не. Сузбијање без максимума има задатак да пронађе пиксел са максималном вредношћу на ивици. На горњој слици се то дешава када пиксел  $A$  има већи интензитет од  $B$  и  $C$ , где су пиксели  $B$  и  $C$  пиксели у смеру градијента  $A$ . Ако је овај услов тачан, задржава се пиксел, у супротном потискује се и пиксел се поставља на нулу (црни пиксел). Укратко, резултат који се добија је бинарна слика са „танким ивицама“.

#### 4. Праг хистерезе

Ова фаза одлучује које су све ивице заиста ивице, а које нису. За ово су потребне две граничне вредности,  $minVal$  и  $maxVal$ . Све ивице са градијентом интензитета већим од  $maxVal$  сигурно ће бити ивице, а оне испод  $minVal$  сигурно неће бити ивице, па се одбацују. Они пиксели који се налазе између ова два прага класификовани су према ивицама или не-ивицама на основу њихове повезаности. Ако су повезани са пикселима са „сигурном ивицом“, сматрају се делом ивице. У супротном, такође се одбацују. Ово је објашњено на слици 10.



Слика 10. Класификација ивица [19]

Ивица  $A$  је изнад  $maxVal$ , па се сматра „сигурном ивицом“. Иако је ивица  $C$  испод  $maxVal$ , она је повезана са ивицом  $A$ , тако да се и она сматра важећом ивицом и добијамо ту пуну криву. Али ивица  $B$ , иако је изнад  $minVal$  и налази се у истој области као и ивица  $C$ , није повезана ни са једном „сигурном ивицом“, па се то одбацује. Дакле, веома је важно да се изабере адекватне вредности  $minVal$  и  $maxVal$  да би се добио тачан резултат. Ова фаза такође уклања шум малих пиксела под претпоставком да су ивице дуге линије. Дакле, коначно се добијају јаке ивице као на слици 11.



Слика 11. *Canny edges*

#### 4.2.4 Проналажење и исцртавање контура на основу ивица

Контуре се могу објаснити једноставно као кривине која спајају све континуиране тачке (дуж границе), исте боје или интензитета. Контуре су користан алат за анализу облика и откривање и препознавање предмета.

За бољу тачност треба користити бинарне слике. Дакле, пре проналаска контура, треба применити праг или откривање ивица.

Функција *findContours* мења изворну слику. Овој функцији се као параметри прослеђују слика са *canny* ивицама, начин претраживања контуре и метода апроксимације контуре. Као резултат, ова функција враћа контуре и хијерархију контура. Дакле, да би се сачувала изворна слика, чак и након проналаска контура, треба је сачувати у некој другој променљивој.

У *OpenCV*-у проналажење контура је попут проналажења белог предмета са црне позадине. Дакле, треба запамтити предмет који треба пронаћи да буде бели, а позадина црна.

За цртање контура се користи функција *drawContours*. Такође се може користити за цртање било ког облика под условом да постоје његове граничне тачке. Први аргумент функције је изворна слика, други аргумент су контуре које треба проследити као Пајтон листу, трећи аргумент је индекс контуре (користан при цртању појединачне контуре – да

би се нацртале све контуре треба додати -1), а преостали аргументи су боја, дебљина линија, итд.

Контуре су границе облика истог интензитета. Ту се чувају (x, y) координате границе облика, али не чува све координате. То је одређено методом апроксимације контуре.

Ако се проследи *CHAIN\_APPROX\_NONE*, све граничне тачке се чувају, али заправо нису потребне све тачке. На пример, понађена је контура равне линије, али нису потребне све тачке на линији да би била представљена. Потребне су само две крајње тачке те линије. То чини *CHAIN\_APPROX\_SIMPLE*. Ово уклања све сувишне тачке и компресује контуру, чиме штеди меморију [12].



Слика 12. Проналажење и исцртавање контура

#### 4.2.5 Сортирање контура

Наредни корак представља сортирање контура на основу области, чувајући притом минималну неопходну површину која износи 30 пиксела (било шта друго што је мање неће бити разматрано).



Слика 13. Контуре чија је минимална површина 30 пиксела

#### 4.2.6 Пролазак кроз контуре и проналазак одговарајуће

Неопходно је проћи кроз све контуре чија је минимална површина 30 пиксела и одабрати ону која је одговарајућа. Пролазак кроз све контуре како би се пронашла најбоља, приближно одговарајућа контура регистарске ознаке возила, врши се проласком кроз једну *if* петљу. Ту се бира она контура која има јасне тачно четири ивице, координате те контуре се памте, затим се креира исечена слика и складишти у одговарајући фолдер.



Слика 14. Проналазак и исцртавање одговарајуће контуре на оригиналној слици

#### 4.2.7 Конвертовање слике у текст

Финални, а и најважнији корак, јесте приказ регистарских ознака возила са регистарске таблице.



Слика 15. Финални резултат рада апликације

OCR технологија помаже да се текст са слике заправо конвертује у прави текст како би апликација могла корисника да обавести о стварним резултатима, а све у евентуалној даљој обради података.

Исечена слика (слика 15) се прво претвара у сиву слику, а затим се проналази вредност прага интензитета сиве. Ако је вредност пиксела већа од вредности прага, додељује јој се једна вредност (може бити бела), у супротном јој се додељује друга вредност (може бити црна). Функција која се користи је *threshold*. Први аргумент је изворна слика, која би требало бити слика у сивим тоновима. Други аргумент је вредност прага која се користи за класификацију вредности пиксела. Трећи аргумент је *maxVal* који представља вредност

коју треба дати ако је вредност пиксела већа од (понекад и мања од) граничне вредности. *OpenCV* пружа различите стилове граничења и о њему одлучује четврти параметар функције, у овом случају је то збир два енумератора, а то су *THRESH\_BINARY\_INV* + *THRESH\_OTSU*. Први енумератор је бинарни, а други користи *Otsu* алгоритам како би се изабрала оптимална вредност прага.

У даљој обради, слика се замућује како би се успешно издвојио текст. То се ради уз помоћ Гаусовог филтрирања.

У овом приступу, уместо филтера кутије који се састоји од једнаких коефицијената филтера, користи се Гаусово језгро. То се ради са функцијом *GaussianBlur*. Треба одредити ширину и висину језгра који би требали бити позитивни и непарни. Такође, треба одредити стандардну девијацију у правцима  $X$  и  $Y$ ,  $\sigma_X$ , односно  $\sigma_Y$ . Ако је наведен само  $\sigma_X$ ,  $\sigma_Y$  се узима као једнак  $\sigma_X$ . Ако су обе дате као нуле, израчунавају се на основу величине језгра. Гаусово филтрирање је врло ефикасно у уклањању Гаусовог шума са слике.

### 4.3 Опис и примена алгоритма за препознавање регистарских ознака возила имплементираних уз утицај алгоритма к најближих суседа (*KNN*)

Примена алгоритма за препознавање регистарских ознака возила имплементираних уз утицај алгоритма к најближих суседа користи све технологије описане у тачки 4.2 уз наравно, као што је већ речено, алгоритам к најближих суседа.

Алгоритам к најближих суседа је алгоритам машинског учења који спада у технике надгледаног учења. Може се користити у класификацији и регресији, али су у индустрији чешћи класификациони проблеми предикције. Сваки алгоритам машинског учења треба да води рачуна о 3 аспекта:

- Једноставном тумачењу излазних података
- Времену извршавања алгоритма
- Колико је добра моћ предвиђања

Овај алгоритам у класификацији класификује тачку посматрања у односу на то како су суседи класификовани.

У препознавању образаца, *KNN* (*k Nearest Neighbors*) алгоритам представља непараметарски метод који се користи за класификацију и регресију. У оба случаја, улаз се састоји од  $k$  најближих тест примера у функцији простора. Излаз зависи од тога да ли се *KNN* алгоритам користио за класификацију, или регресију:

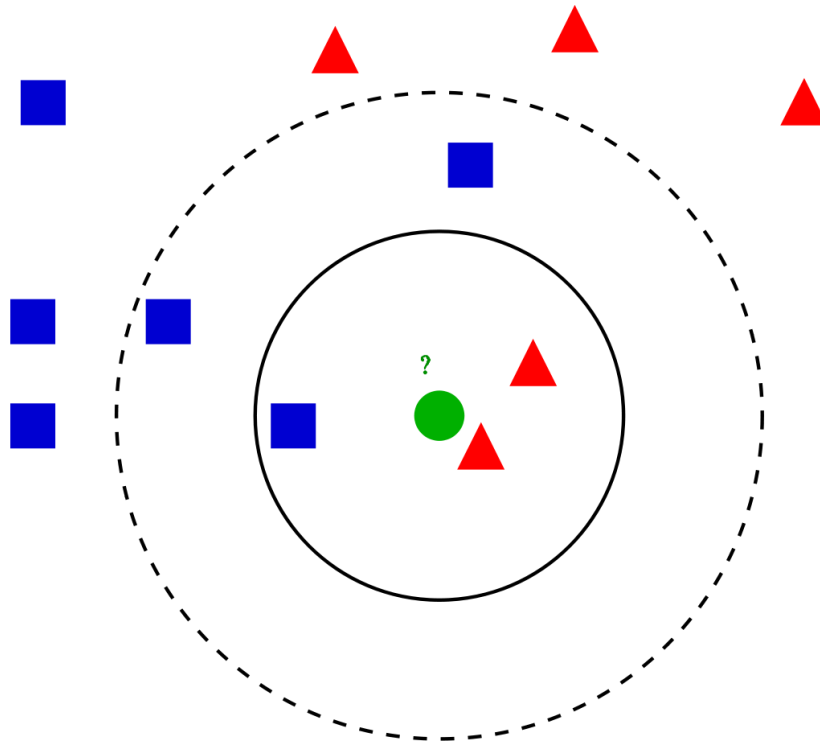
- У *KNN* класификацији, излаз је члан класе. Објект се класификује гласовима већине својих суседа, тако да објект буде распоређен у класу најчешћу међу својим  $k$  најближим суседима ( $k$  је позитиван цео број, најчешће мали број). Ако је  $k = 1$ , онда се објект једноставно додељује класи тог једног најближег суседа.
- У *KNN* регресији, излаз представља вредност објекта. Ова вредност представља просек вредности његових  $k$  најближих суседа.

*KNN* је тип учења које се заснива на примерима, или тзв. лењо учење, где је апроксимација функције локална и сва израчунавања се одлажу до класификације. *KNN* алгоритам је најједноставнији од свих алгоритама машинског учења.

Додела тежине доприносима суседа може бити од користи и за класификацију и за регресију, тако да ближи суседи доприносе просеку више у односу на оне удаљеније. На пример, честа шема доделе тежине је таква да се сваком од суседа додели тежина од  $1/d$ , где  $d$  представља растојање од суседа.

Суседи се узимају из скупа објеката за које је класа (за *KNN* класификацију) или вредност објекта (за *KNN* регресију) позната. Ово се може посматрати као тест скуп за алгоритам, иако експлицитни тест кораци нису неопходни.

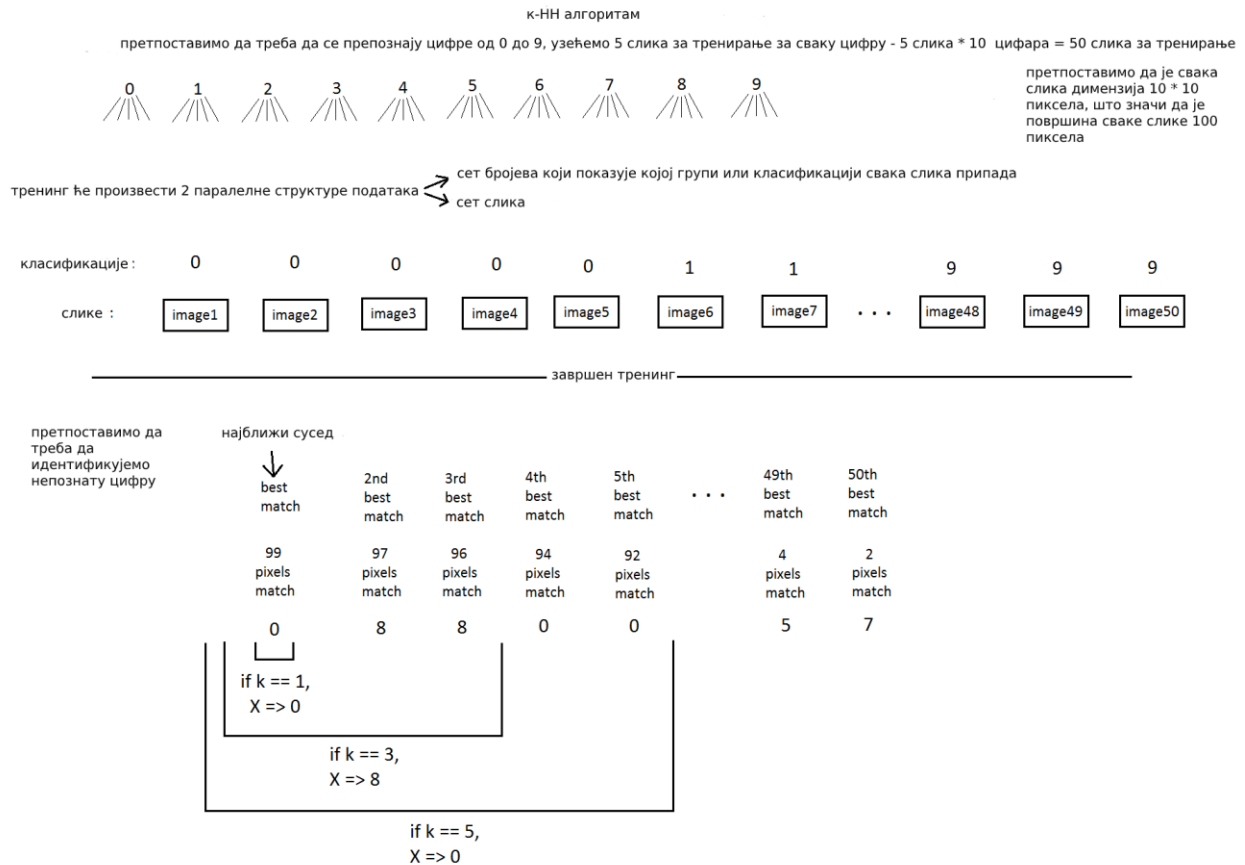
Мана *KNN* алгоритма је та што је осетљив на локалне структуре података. Алгоритам нема ништа заједничко и не треба га мешати са кластеризацијом методом *k* средњих вредности, што је друга популарна техника машинског учења.



Слика 16. Пример *KNN* (*k* Nearest Neighbors) класификације [23]

На слици 16 приказан је тест пример (зелени круг) који треба да буде класификован или као прва класа плавих квадрата, или као друга класа црвених троуглова. Ако је  $k = 3$  (пуна линија круга), онда је додељен другој класи јер се унутар унутрашњег круга налазе 2 троугла и само један квадрат. Ако је  $k = 5$  (испрекидана линија круга), онда је додељен првој класи (3 квадрата у односу на 2 троугла унутар спољашњег круга).

За потребе овог мастер рада, алгоритам је претходно истрениран на словима и бројевима, након чега је примењен за потребе препознавања карактера на регистарској таблици возила. То је описано и приказано на дијаграму (слика 17).

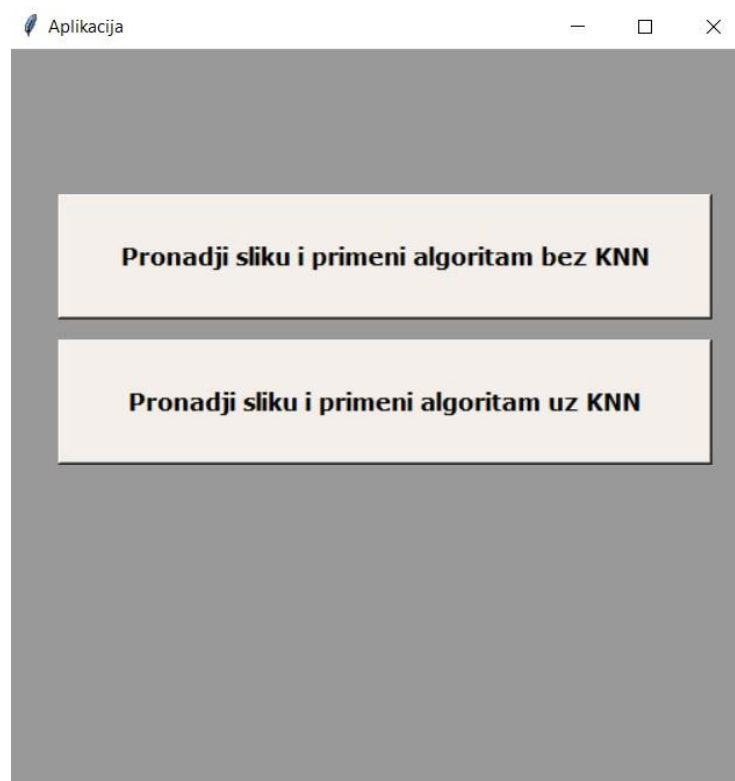


Слика 17. Преглед *KNN* алгоритма

## 5. Резултати рада апликације за препознавање регистарских ознака возила

Циљ овог мастер рада било је креирање апликације за препознавање регистарских ознака возила коришћењем библиотека програмског језика *Python* као и библиотека *OpenCV*, софтверског пакета који служи за рад са сликама приликом програмирања одређених софтверских решења.

Када се апликација покрене, приказује се почетни екран приказан на слици 18.



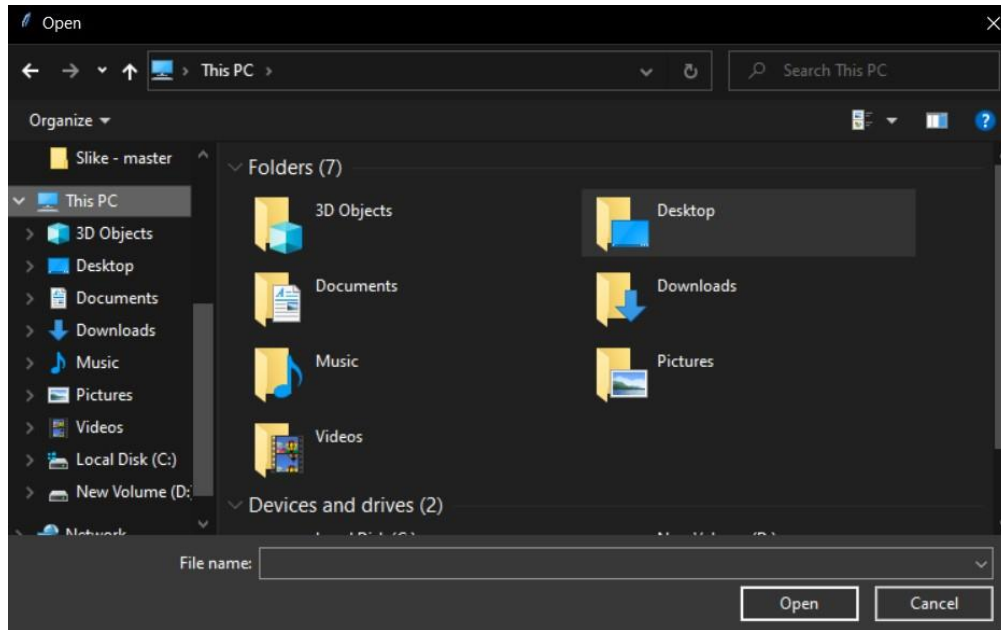
Слика 18. Почетни екран апликације

Треба напоменути да се у фолдеру под називом *LicPlateImages* који се налази у рачунару (на локацији *C:\Users\ficas\Desktop\app\LicPlateImages*) налазе слике које представљају базу података, а на којима се налазе одређена возила.

Кориснику апликације је омогућено да пронађе и учита слику која се налази у рачунару и да на тој слици препозна регистарску ознаку возила која се на њој налази. То је омогућено кликом и избором једног од два дугмета:

- *Pronadji sliku i primeni algoritam bez KNN*
- *Pronadji sliku i primeni algoritam uz KNN*

Тада се отвара прозор који кориснику омогућава да пронађе жељену слику (слика 19).



Слика 19. Прозор за тражење слике

Када се слика изабере, потребно је сачекати пар секунди да се програм изврши након чега ће бити приказана изабрана слика. Прво ће бити приказана оригинална слика, а након тога ће уследити приказ осталих слика, тј. изабрана слика измењена под утицајем различитих функција.

## 5.1 Успешни резултати рада апликације применом оба начина

### 5.1.1 Успешни резултати без употребе алгоритма к најближих суседа

Први корак при покретању апликације јесте избор и приказ слике возила на коме је потребно препознати регистарску ознаку.



Слика 20. Приказ оригиналне слике возила [25]

Следећи корак јесте приказ сиве слике. Оригинална, слика у боји, се конвертује у сиву слику ради лакше обраде.



Слика 21. Приказ сиве слике возила

Након дејста билатералног филтера, приказ је на слици 22.



Слика 22. Приказ слике возила након дејста билатералног филтера

*Canny* ивице су приказане на слици 23.



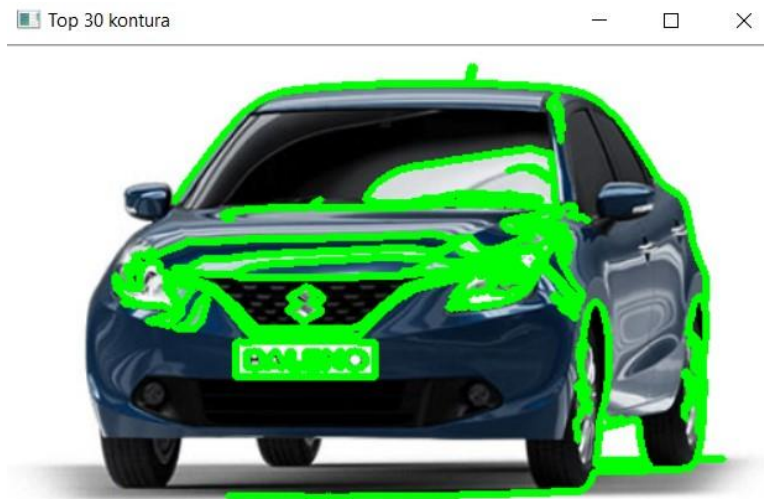
Слика 23. Приказ *canny* ивица возила

На слици 24. су приказане све детектоване контуре возила.



Слика 24. Приказ детектованих контура возила

Приказ првих 30 контура по површини је на слици 25.



Слика 25. Приказ првих 30 контура возила по површини

На самом крају, на слици 26. је приказана коначна слика са детектованом регистарском ознаком возила.



Слика 26. Приказ коначне слике са детектованом регистарском ознаком возила

Приказује се обавештење у виду прозора са препознатим ознакама на возилу.

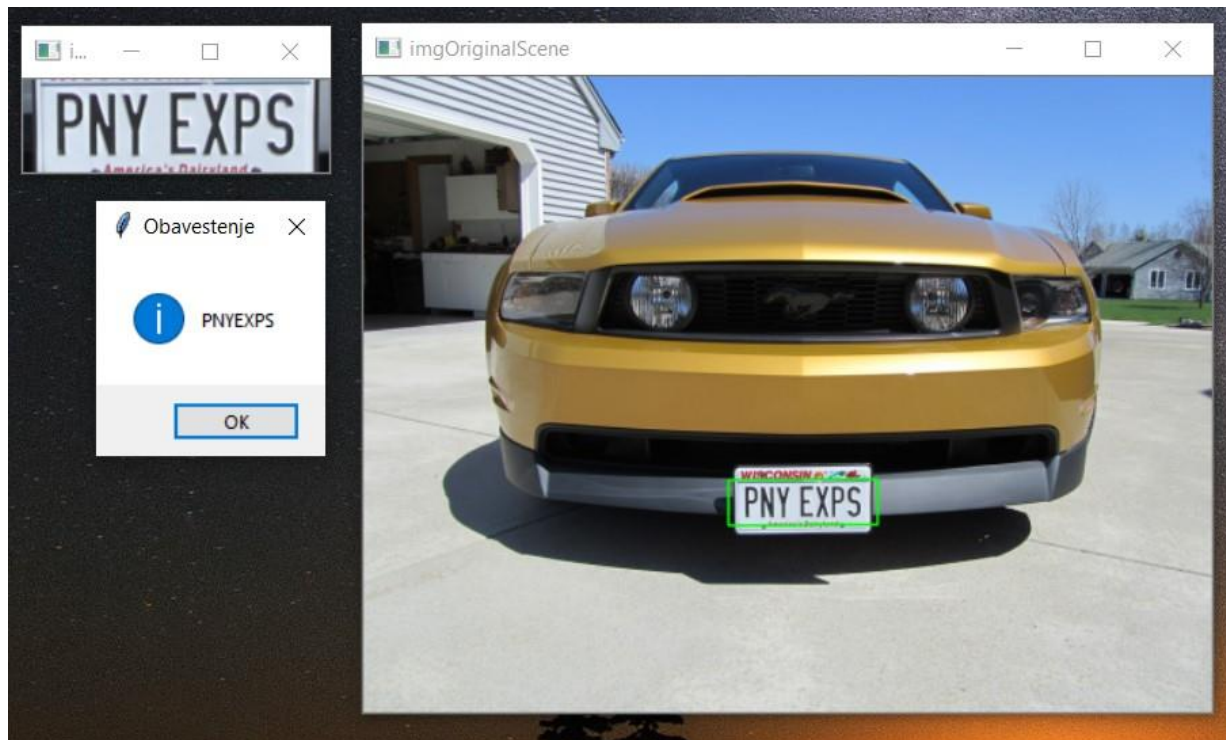


Слика 27. Обавештење са регистарском ознаком возила

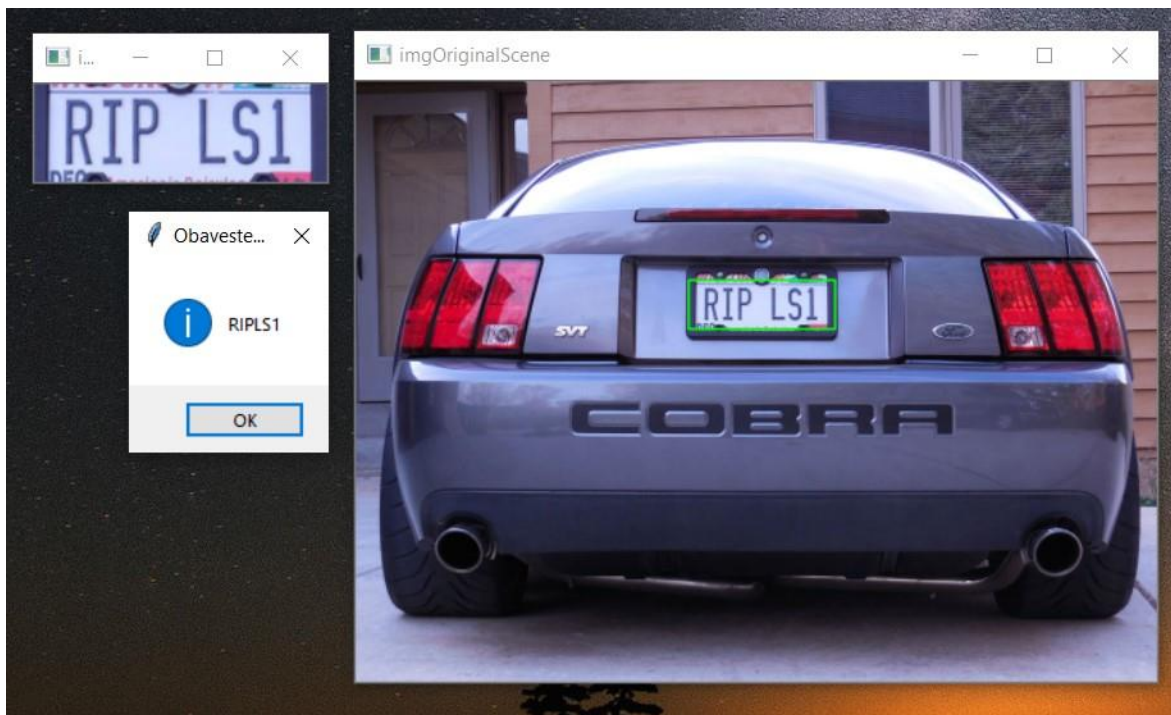


Слика 28. Приказ регистарске ознаке возила (исечена слика)

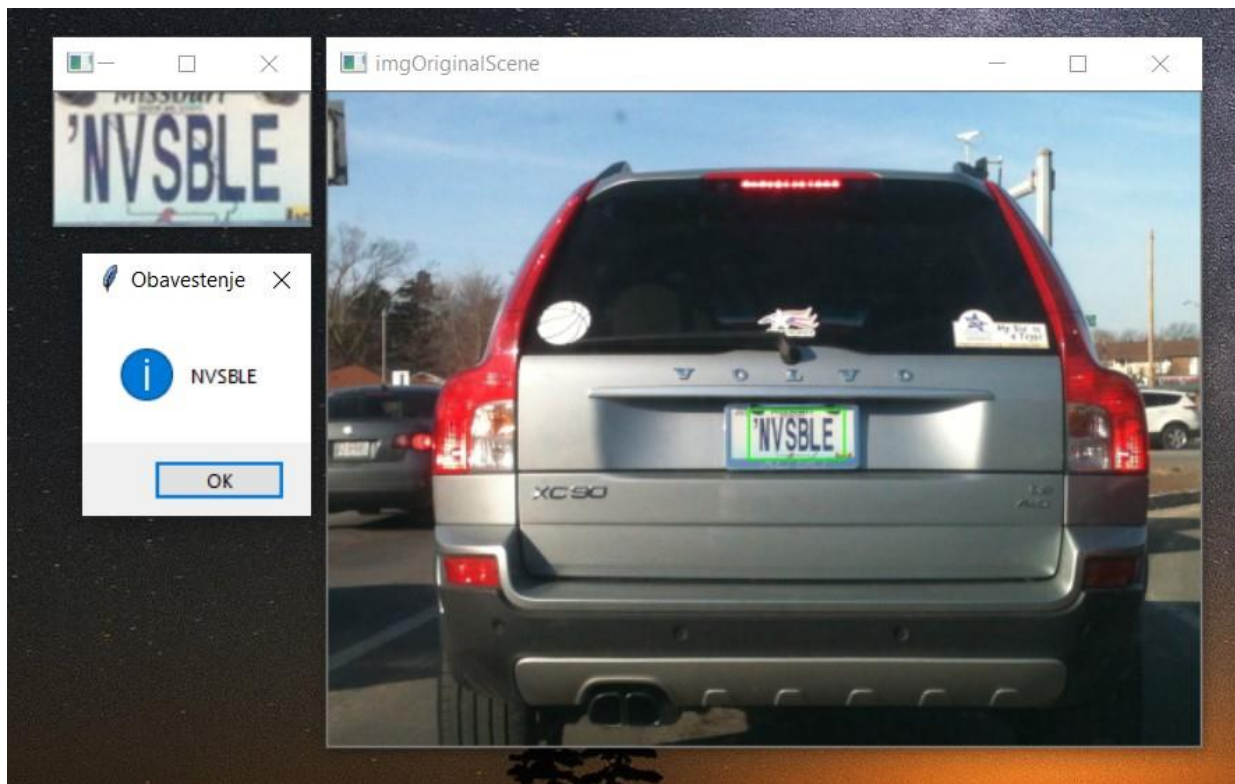
### 5.1.2 Успешни резултати уз употребу алгоритма к најближих суседа



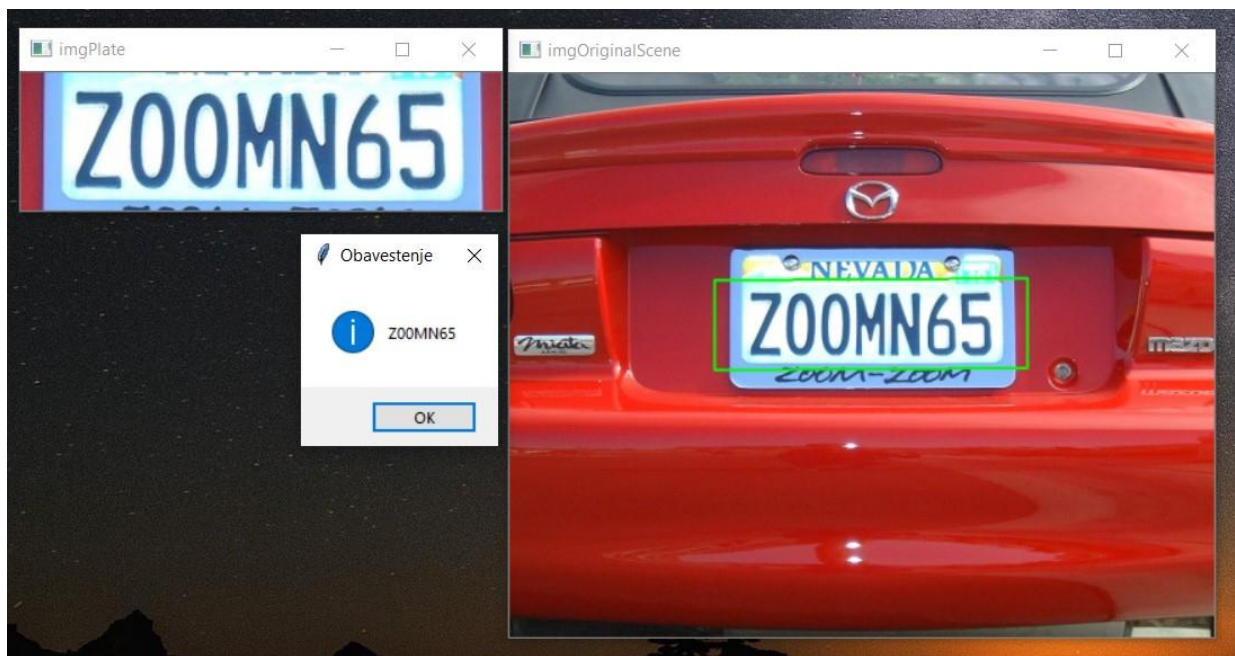
Слика 29. Успешан резултат рада апликације уз коришћење *KNN* алгоритма



Слика 30. Успешан резултат рада апликације уз коришћење *KNN* алгоритма



Слика 31. Успешан резултат рада апликације уз коришћење *KNN* алгоритма



Слика 32. Успешан резултат рада апликације уз коришћење *KNN* алгоритма

## 5.2 Неуспешни резултати рада апликације применом оба начина

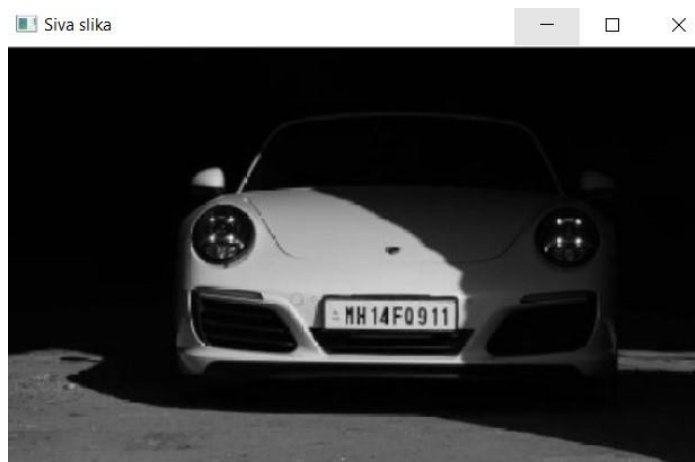
### 5.2.1 Неуспешни резултати без употребе алгоритма к најближих суседа

Као и свака друга апликација, ни ова апликација није савршена. Постоје резултати рада који показују да је дошло до грешке и на чему треба порадити у неким наредним верзијама. У наставку ће то бити демонстрирано примерима.

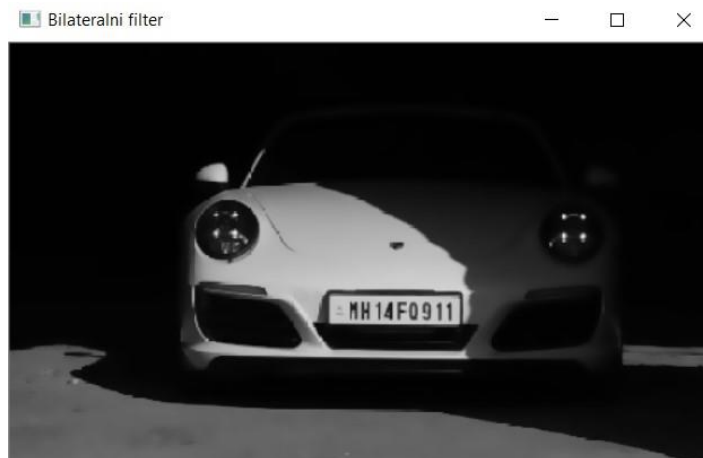
Наредни пример приказује возило чији се један део налази у осветљеном делу простора, а један део у неосветљеном.



Слика 33. Приказ оригиналне слике возила [26]



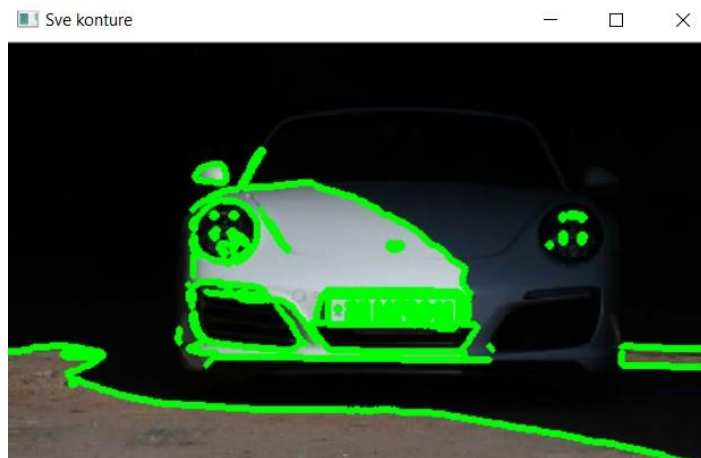
Слика 34. Приказ сиве слике возила



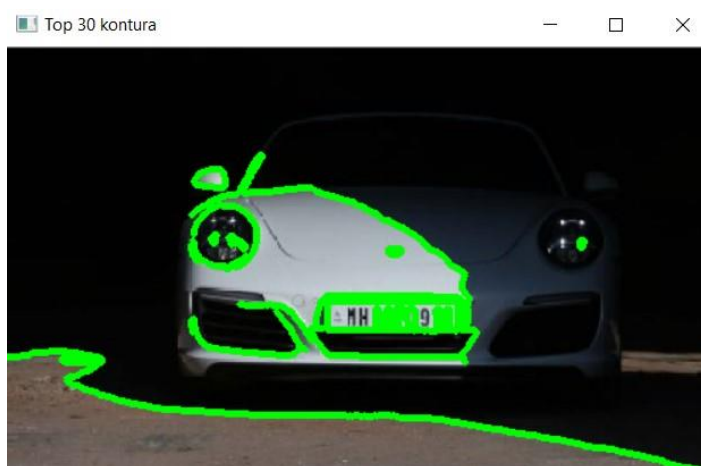
Слика 35. Приказ слике возила након дејста билатералног филтера



Слика 36. Приказ *canny* ивица возила



Слика 37. Приказ детектованих контура возила

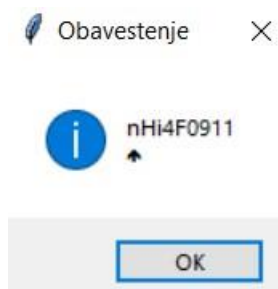


Слика 38. Приказ првих 30 контура возила по површини



Слика 39. Приказ коначне слике са детектованом регистарском ознаком возила

Када се појави обавештење може се видети да је дошло до грешке у препознавању и да нису сви карактери прецизно прочитани. До грешке у препознавању је дошло зато што слика није најбољег квалитета, па је систем први и трећи карактер погрешно прочитао зато што су слични са оним карактерима које је приказао као резултујуће. Први је уместо слова *М* заменио словом *n*, а трећи је број 1 заменио словом *i*.

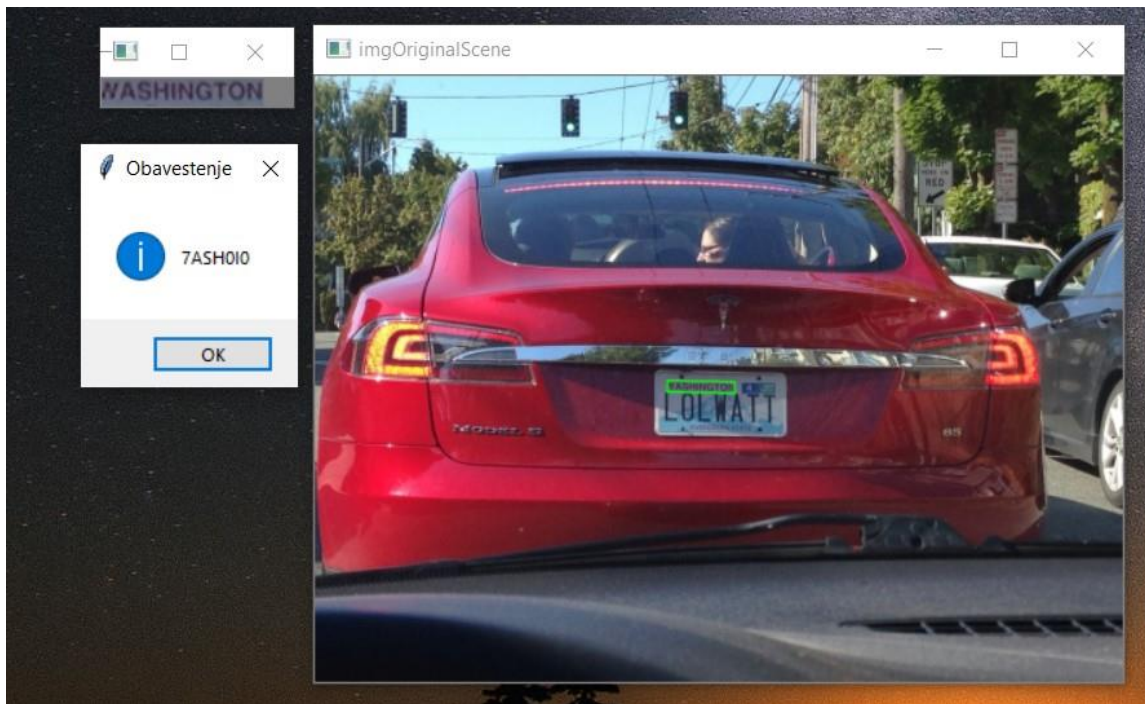


Слика 40. Обавештење са регистарском ознаком возила

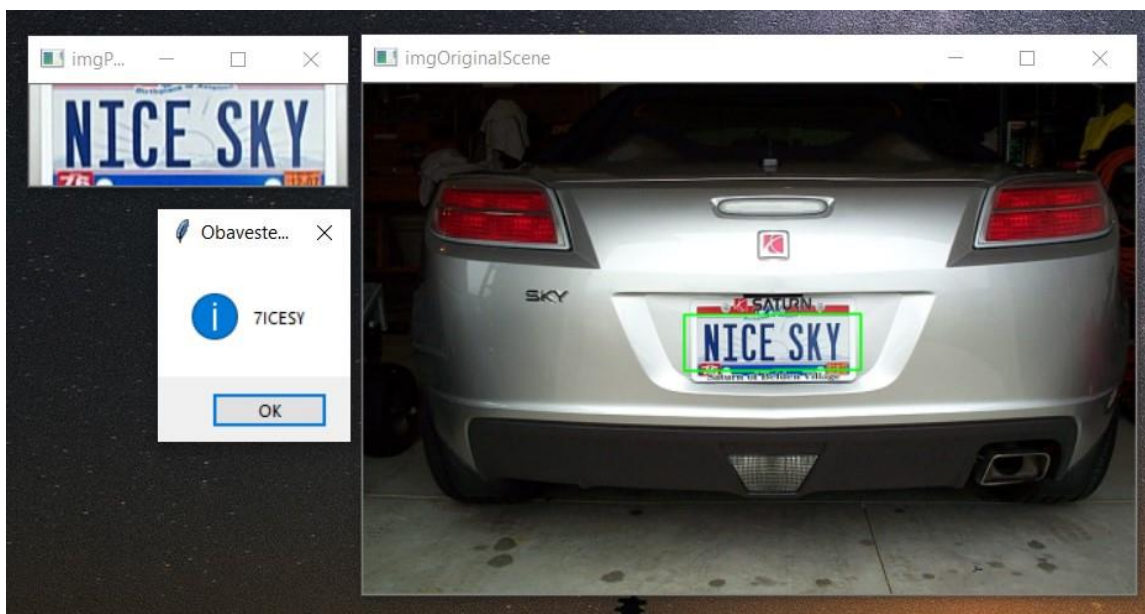


Слика 41. Приказ регистарске ознаке возила (исечена слика)

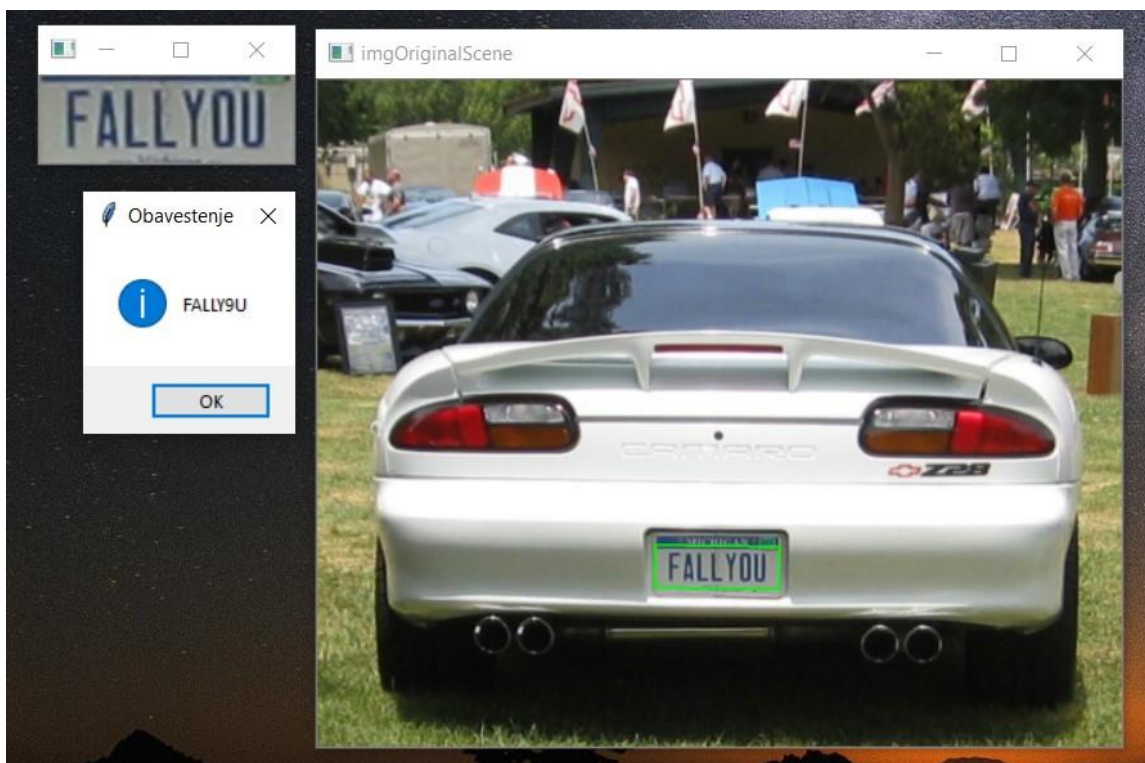
### 5.2.2 Неуспешни резултати уз употребу алгоритма к најближих суседа



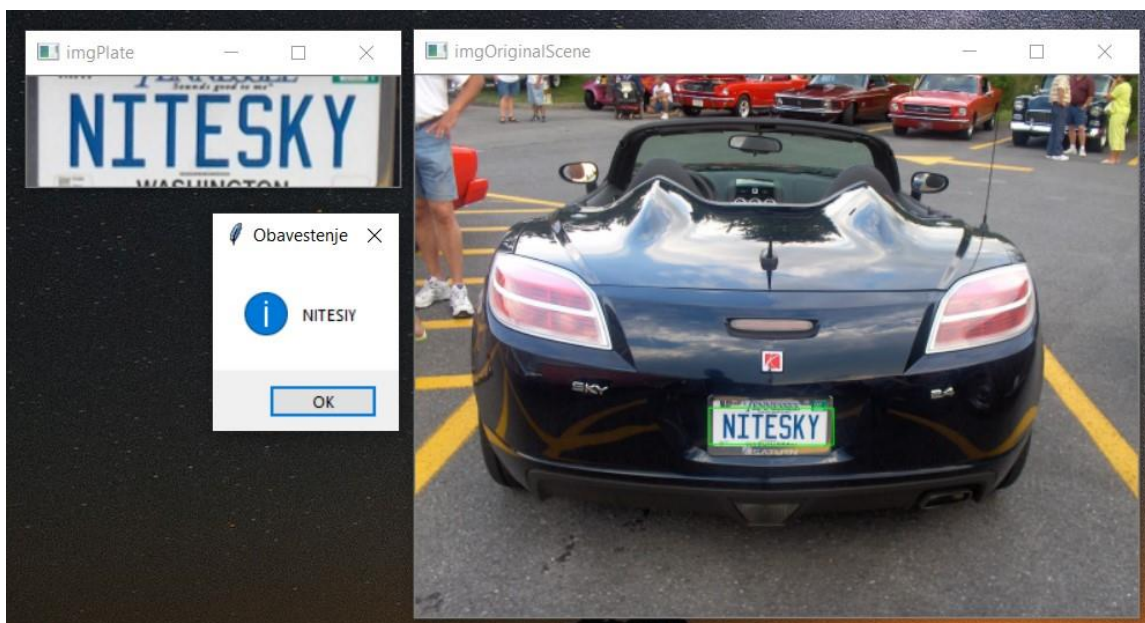
Слика 42. Неуспешан резултат рада апликације уз коришћење KNN алгоритма



Слика 43. Неуспешан резултат рада апликације уз коришћење KNN алгоритма



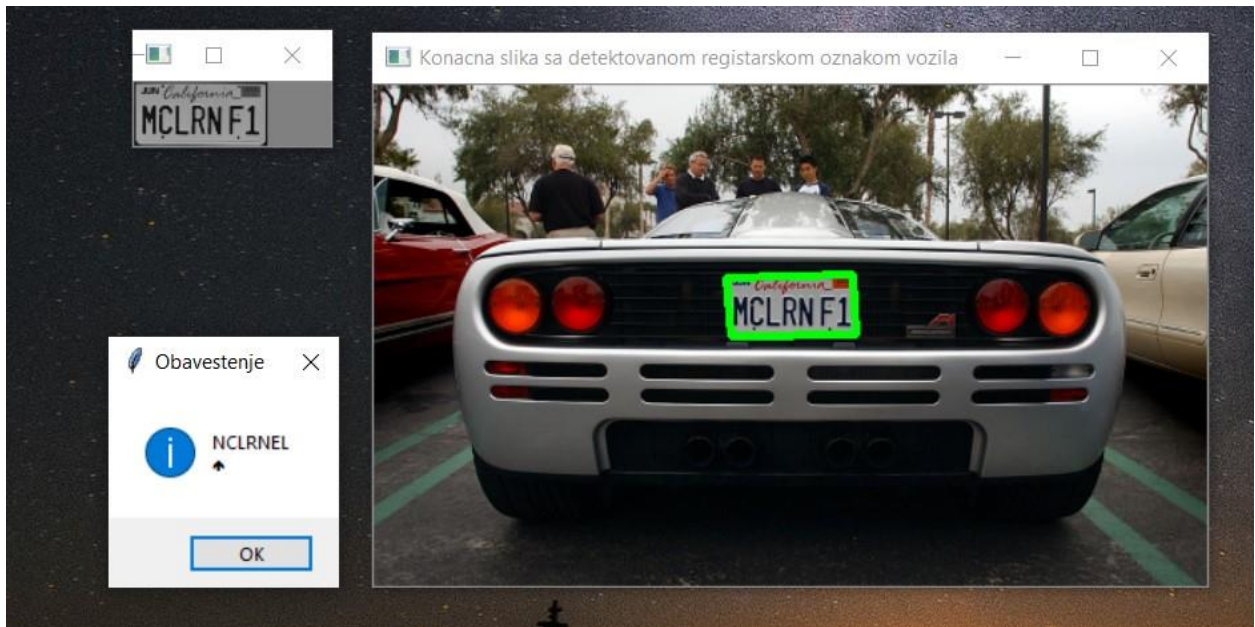
Слика 44. Неуспешан резултат рада апликације уз коришћење KNN алгоритма



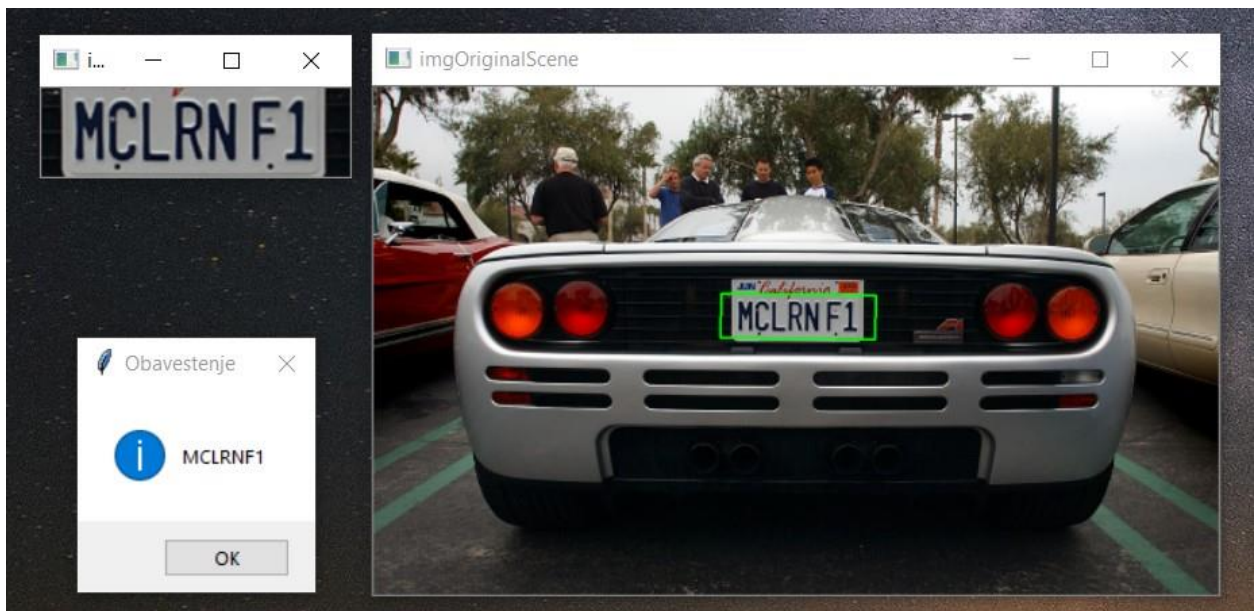
Слика 45. Неуспешан резултат рада апликације уз коришћење KNN алгоритма

### 5.3 Поређење резултата рада апликације применом оба начина

Као што се може и претпоставити, апликација ће дати боље резултате уз коришћење *KNN* алгоритма што ће бити демонстрирано у наставку. На слици 38 је приказан резултат рада апликације без коришћења *KNN* алгоритма што је довело до лоших резултата, а на слици 39 резултат рада апликације уз коришћење *KNN* алгоритма и то је успешно резултирало.

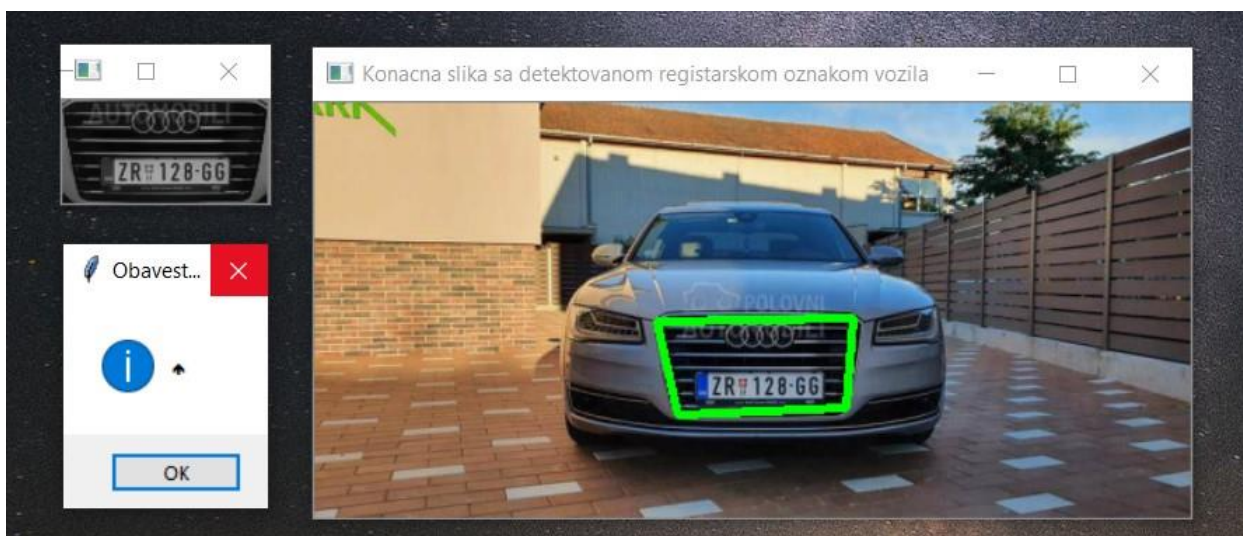


Слика 46. Резултат рада апликације без коришћења *KNN* алгоритма

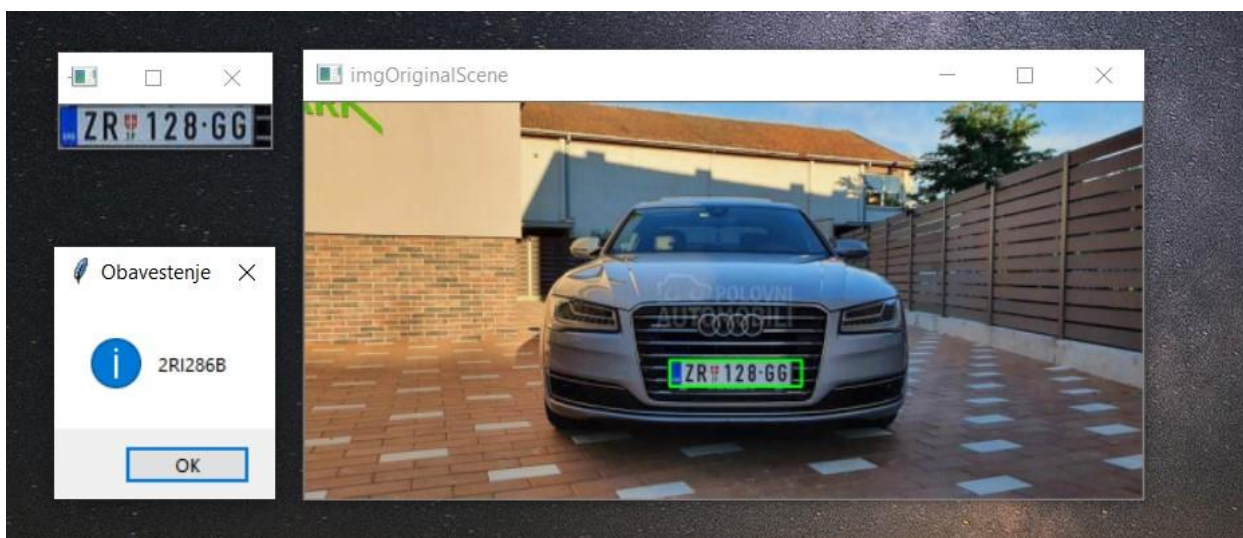


Слика 47. Резултат рада апликације уз коришћење *KNN* алгоритма

Оно што је значајно јесте то да оба алгоритма имају потешкоћа са регистарским ознакама које издаје МУП Републике Србије. Без употребе *KNN* алгоритма, резултата нема, док уз употребу *KNN* алгоритма, постоје резултати уз одређени степен грешке. Ралози за то могу бити фонт који се користи за регистарске ознаке, грб и други знакови који се ту налазе. У неким наредним верзијама апликације је неопходно радити на савладавању оваквих проблема.



Слика 48. Резултат рада апликације без коришћења *KNN* алгоритма



Слика 49. Резултат рада апликације уз коришћење *KNN* алгоритма

## 6. Изазови, контроверзе и проблеми

### 6.1 Обмане

Власници возила су користили разне технике у покушају да избегну системе и камере за примену правила о путевима уопште. Једна метода повећава рефлектујућа својства слова и повећава вероватноћу да систем неће моћи да лоцира регистарску таблицу, или да произведе довољно висок ниво контраста да би је могао прочитати. То се обично ради помоћу поклопца за регистарску таблицу, или спреја, мада се тврдње у вези са њиховом ефикасношћу оспоравају. У већини држава поклопци за регистарске таблице су илегални и покривени постојећим законима, док у већини земаља не постоји закон који забрањује употребу спрејева. Неки возачи су покушали да запрљају своју регистарску таблицу прљавштином како би прикрили регистарске ознаке.

Нови оквири око тексашких регистарских таблица су 1. септембра 2003. године постали недозвољени законом Сената Тексаса 439 јер су стварали проблеме са уређајима. Тај закон је учинио прекршај класе Ц (кажњив новчаном казном до 200 америчких долара), или класе Б (новчаном казном до 2.000 америчких долара и 180 дана затвора), ако се може доказати да је то учинио власник намерно прикривајући своју регистарску таблицу. Закон је касније разјашњен 2007. како би се дозволили мало другачији оквири.

Ако систем не може да прочита регистарску таблицу, може да услика возило, а оператери (људски фактор) да виде да ли су у стању да идентификују алфанумеричке знакове.

Да би се избегло надгледање или казнене пријаве, дошло је до пораста клонирања возила. То се обично постиже копирањем регистарских таблица из другог аутомобила сличног модела и старости. То може бити тешко за открити, посебно јер клонери могу променити регистарске таблице и понашање приликом путовања, да би ометали истраге.

2013. године истраживачи из компаније *Sunflex Zone Ltd* креирали су оквир за регистарске таблице за приватност који користи инфрацрвену светлост како би регистарску таблицу учинио нечитљивом за системе за препознавање регистарских таблица.

## 6.2 Контроверзе

Увођење система довело је до страха од погрешне идентификације и даљег надзора од 1984. године. У Сједињеним Америчким Државама, неки људи, као што је Грег Истербук, противе се ономе што називају „машинама које издају карте за пребрзу вожњу и карте за црвено светло“ као почетак клизавог нагиба ка аутоматизованом правосудном систему.

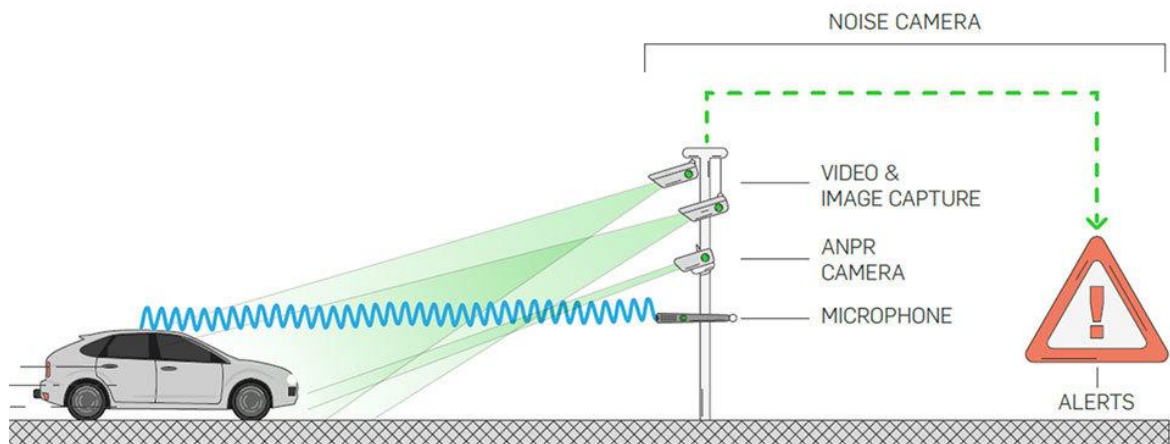
Сличне критике изнете су и у другим земљама. Истербук такође тврди да се ова технологија користи како би се максимизовао приход државе, уместо да би се промовисала безбедност. Систем електронског надзора издаје казне које у САД-у често премашују 100 долара, а грађанима је практично немогуће да се боре на суду без помоћи адвоката. Приход који генеришу ове машине издашно деле са приватним корпорацијама које их праве и управљају њима, стварајући снажан подстицај за подешавање система како би генерисао што више казни.

Старији системи су били приметно непоуздани. У Великој Британији је познато да ово доводи до нетачног подношења оптужнице, јер власник возила мора платити 10 фунти да би му се издао доказ (или не) о прекршају. Побољшања технологије драстично су смањила стопу грешака, али лажне оптужбе су и даље довољно честе да представљају проблем.

Можда најпознатији инцидент који укључује злоупотребу базе података у Северној Америци је случај извештача Кери Диота 2004. године. Диоте је написао чланак у којем критикује употребу полицијских камера у Едмонтоновој полицији за повећање прихода, а у знак одмазде додата је база података о „ризичним возачима“ у покушају да надгледа њихове навике и створи прилику за његово хапшење. Као резултат тога отпуштени су шеф полиције и неколико полицајаца, а канцеларија комесара за приватност Канаде изразила је забринутост јавности због „растуће полицијске употребе технологије за шпијунирање возача“.

Остали проблеми укључују складиштење информација које би се могле користити за идентификацију људи и чување детаља о њиховим возачким навикама и свакодневном животу, што је у супротности са Законом о заштити података заједно са сличним законодавством.

Такође, у Великој Британији постоји случај где је употреба специјализованих камера незаконита према Уредби о истражним овлашћењима из 2000. године. Кршење постоји, у чињеници да се систем користи за надгледање активности грађана који поштују закон и према свима се понаша као према осумњиченим криминалцима који су предвиђени да буду предмет истраге према закону. Познато је да и сама полиција систем назива „базом података о кретању у саобраћају 24 сата дневно, 7 дана у седмици“, што представља преусмеравање од предвиђене сврхе идентификовања возила умешаних у криминалне радње. Супротно гледиште је да ће тамо где су регистарске таблице клониране, „очитавање“ возила невиног возача омогућити уклањање тог возила из истраге визуелним прегледом сачуваних слика. Исто тако, украдена возила системи читају између времена крађе и пријављују полицији помажући у истрази.



Слика 50. Начин функционисања система за кажњавање возача [22]

### 6.3 Неусклађеност регистарских таблица и разлике у надлежности

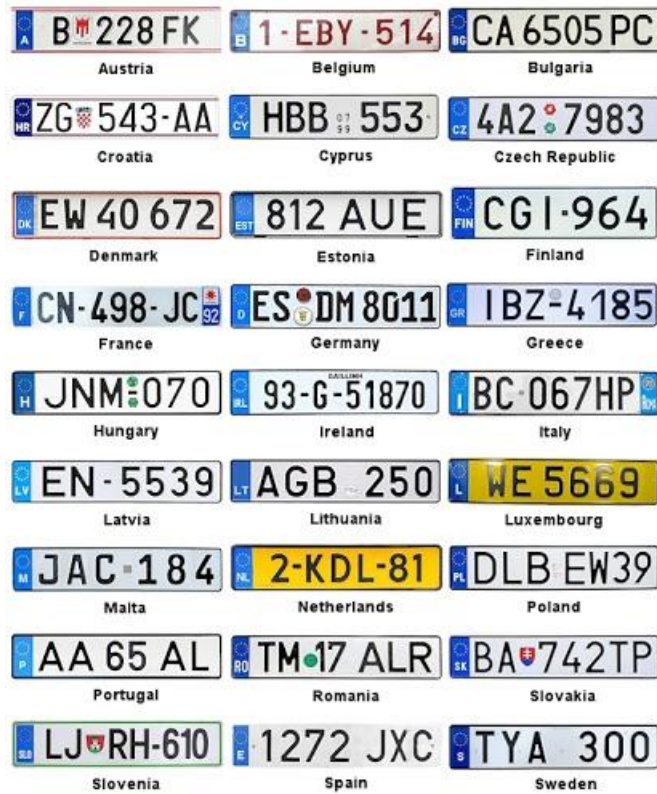
Многи системи захтевају тачност када су обучени за подударање плоча из једне државе или региона, али могу да пропадну када покушавају да препознају таблице из других држава због разлика у формату, фонту, боји, распореду и других карактеристика регистарских таблица. Неке државе нуде таблице са сличним карактерима (посебно у САД-у), што може створити много варијација унутар једне државе.



Слика 51. Регистарске таблице у САД-у [20]

Повремено америчке државе праве значајне промене у свом протоколу регистарских таблица које утичу на тачност OCR-а. Могу додати знак или додати нови дизајн регистарских таблица. Системи се морају брзо прилагодити овим променама да би били ефикасни. Још један изазов код ових система је тај што неке државе имају исти протокол регистарских таблица. На пример, више држава користи стандардна три слова, а затим четири броја. Дакле, сваки пут када се алармирају ови системи, на кориснику је одговорност да осигура да се таблица која је изазвала аларм поклапа са стањем придруженим регистарским таблицама наведеним на рачунару у аутомобилу. За максималну ефикасност, овај систем треба да буде у стању да препозна таблице из државе којој припада, али ове бројне променљиве чине такве задатке тешким.

Тренутно бар један амерички провајдер ових система (*PlateSmart*) тврди да је њихов систем независно прегледан као способан да тачно препозна америчке регистарске таблице, а једна европска компанија тврди да њихов систем може разликовати све регистарске таблице свих држава унутар ЕУ.



Слика 52. Регистарске таблице држава унутар ЕУ [21]

## 7. Закључак

У овом раду су представљени напредни алгоритми за препознавање регистарских ознака возила. Такође, приказани су резултати практичне имплементације једног једноставног алгоритма издвајања обележја регистарских таблица, који обухвата одговарајућу предобраду слике, локализацију и сегментацију таблице, као и издвајање прочитаних карактера, а реализован је у Пајтон програмском језику. Исто тако, приказани су и резултати имплементације претходно поменутог алгоритма уз примену *KNN* алгоритма, што је дало још боље резултате.

Када се говори о системима за препознавање регистарских ознака возила, може се рећи да је њихово постојање само по себи већ предност, јер такви системи налазе широку примену у многим делатностима и наукама. Питање безбедности и приватности је одувек било значајно и самим тим примена оваквог система је јако битна за тако осетљиве теме. Безбедносне службе могу користити овакав систем у циљу проналажења особа на потерницама или осумњичених за кривична дела, а такође и за проналажење несталих особа. Овакви системи налазе своју примену и у вештачкој интелигенцији и роботизици где се уз друге алгоритме могу применити приликом креирања сложенијих система. Ови системи се такође могу користити за:

- Контролу одређених деоница на путевима, за мерење просечне брзине возила на већим удаљеностима
- Граничне прелазе
- Поврат возила
- Бензинске станице које пријављују када возило оде без плаћања горива
- Анализе понашања на путу (избор руте, порекло-одредиште, итд.), у сврху планирања транспорта
- Препознавање возила сталних купаца, да би се аутоматски препознали купци на основу њихове регистарске таблице и да би им се понудили предмети које су наручили последњи пут када су користили услугу
- Помагање системима за управљање посетиоцима у препознавању возила гостију
- Полицију и комуналну полицију
- Компаније које се баве сервисима за паркирање возила
- За подизање или спуштање аутоматских стубова у паркинг зонама
- Хотеле
- Спровођење закона о померању када пролазе возила под интервенцијом
- Аутоматизовано испитивање емисије штетних издувних гасова

Како су овакви системи и алгоритми који се примењују у њима развијани одавно и још увек се развијају, простора за њихово унапређивање има на претек. Уз додатна и озбиљнија истраживања могуће је креирање нових модификација које би потенцијално побољшале перформансе оваквих система.

Постоји више начина за решавање проблема детекције и сегментације регистарских таблица. Програм који је написан и чији су резултати представљени у овом раду, даје

једно од могућих решења за овај проблем. Сваки од корака који су примењени, одговара проблематици и специфичности намене самог програма. Логичан след, након детекције и сегментације таблице, било би препознавање карактера на таблици. То је трећи и последњи корак у свим системима за аутоматско препознавање таблица, тако да се описани програм може користити као део софтвера ових система. У овом раду су коришћене слике лошијег квалитета од оних које би биле начињене специјалним камерама израђеним посебно у ове сврхе, што би директно утицало на тачност програма.

Правци будућих истраживања свакако ће се односити на анализу могућности интеграције са другим алгоритмима, што би сигурно допринело унапређењу описаних алгоритама, укључујући и употребу метода вештачке интелигенције и техника коришћених у постојећим *OCR* системима. Без обзира да ли би се имплементирани алгоритам за препознавање слова ослањао на поређење са шаблоном, статистичке методе, или идентификацију граница објеката, односно векторизацију слике, била би неопходна анализа могућности интеграције са алгоритмима изложеним у овом раду.

## 8. Литература

- [1] Anagnostopoulos, C.N., Anagnostopoulos, I., Psoroulas, I., Loumos, V., Kayafas, E.: License Plate Recognition From Still Images and Video Sequences: A Survey. *IEEE Transactions on Intelligent Transportation Systems* 9(3), 377–391 (sep 2008). <https://doi.org/10.1109/TITS.2008.922938>, <http://ieeexplore.ieee.org/document/4518951/>
- [2] Bulan, O., Kozitsky, V., Ramesh, P., Shreve, M.: Segmentation- and AnnotationFree License Plate Recognition With Deep Localization and Failure Identification. *IEEE Transactions on Intelligent Transportation Systems* 18(9), 2351–2363 (sep 2017). <https://doi.org/10.1109/TITS.2016.2639020>
- [3] Du, S., Ibrahim, M., Shehata, M., Badawy, W.: Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. *IEEE Transactions on Circuits and Systems for Video Technology* 23(2), 311–325 (feb 2013). <https://doi.org/10.1109/TCSVT.2012.2203741>, <http://ieeexplore.ieee.org/document/6213519/>
- [4] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2), 303–338 (Jun 2010)
- [5] Goncalves, G.R., da Silva, S.P.G., Menotti, D., Schwartz, W.R.: Benchmark for license plate character segmentation. *Journal of Electronic Imaging* 25(5), 1–5 (2016), <http://www.ssig.dcc.ufmg.br/wp-content/uploads/2016/11/ JEI-2016-Benchmark.pdf>
- [6] Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2016)
- [7] Hsu, G.S., Ambikapathi, A., Chung, S.L., Su, C.P.: Robust license plate detection in the wild. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. pp. 1–6. No. August, IEEE (aug 2017). <https://doi.org/10.1109/AVSS.2017.8078493>, <http://ieeexplore.ieee.org/document/8078493/>
- [8] Hsu, G.S., Chen, J.C., Chung, Y.Z.: Application-Oriented License Plate Recognition. *IEEE Transactions on Vehicular Technology* 62(2), 552–561 (feb 2013). <https://doi.org/10.1109/TVT.2012.2226218>
- [9] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. pp. 21–37 (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- [10] Wang, F., Zhao, L., Li, X., Wang, X., Tao, D.: Geometry-Aware Scene Text Detection with Instance Transformation Network. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1381–1389. Salt Lake City (2018)

- [11] Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Transactions on pattern analysis and machine intelligence* 31(10), 1733–1746 (2009)
- [12] Xie, L., Ahmad, T., Jin, L., Liu, Y., Zhang, S.: A New CNN-Based Method for Multi-Directional Car License Plate Detection. *IEEE Transactions on Intelligent Transportation Systems* 19(2), 507–517 (feb 2018).<https://doi.org/10.1109/TITS.2017.2784093>
- [13] [https://www.kurir.rs/data/images/2020/07/25/20/2285801\\_0802zorana-jevtic-2\\_ff.jpg](https://www.kurir.rs/data/images/2020/07/25/20/2285801_0802zorana-jevtic-2_ff.jpg)
- [14] <https://upload.wikimedia.org/wikipedia/commons/d/d6/Dubai-anpr-camera.png>
- [15] [https://upload.wikimedia.org/wikipedia/commons/3/3c/Dutch\\_license\\_plate\\_segment.jpg](https://upload.wikimedia.org/wikipedia/commons/3/3c/Dutch_license_plate_segment.jpg)
- [16] [https://upload.wikimedia.org/wikipedia/commons/9/9c/California\\_license\\_plate\\_ANPR.png](https://upload.wikimedia.org/wikipedia/commons/9/9c/California_license_plate_ANPR.png)
- [17] [https://upload.wikimedia.org/wikipedia/commons/3/33/United\\_Kingdom\\_license\\_plate\\_KAF\\_762L\\_1963.jpg](https://upload.wikimedia.org/wikipedia/commons/3/33/United_Kingdom_license_plate_KAF_762L_1963.jpg)
- [18] [https://opencv-python-tutroals.readthedocs.io/en/latest/\\_images/nms.jpg](https://opencv-python-tutroals.readthedocs.io/en/latest/_images/nms.jpg)
- [19] [https://opencv-python-tutroals.readthedocs.io/en/latest/\\_images/hysteresis.jpg](https://opencv-python-tutroals.readthedocs.io/en/latest/_images/hysteresis.jpg)
- [20] <https://cdn5.vectorstock.com/i/1000x1000/31/49/car-number-plates-license-registration-in-usa-vector-26003149.jpg>
- [21] <https://lh3.googleusercontent.com/proxy/StIVnZ1fHidUq8IEmmI542ZVvCLCgdkZV0elx61yTPRCxo9APRnZXI-PjskJoGcYPqLOedS2aHTNr-QS2J7qbbXWDVY4SEP0KzwJo-7oIQ>
- [22] [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/88199/s960\\_noise-cameras.jpg](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/88199/s960_noise-cameras.jpg)
- [23] <https://upload.wikimedia.org/wikipedia/commons/thumb/e/e7/KnnClassification.svg/1024px-KnnClassification.svg.png>
- [24] <https://vroom.ro/wp-content/uploads/2016/03/audi-q2-6.jpg>
- [25] <https://autolife.com.np/wp-content/uploads/2016/04/cbb473957c2d70948e236a8213bdcf97.jpg>
- [26] <https://gtlux.com/wp-content/uploads/2020/02/DSC0009-min.jpg>