

Проектна задача по Дигитално процесирање на слика

Марија Трпчевска
196002

Детекција на регистарски таблички и препознавање на карактери

Апстракт

Полето на компјутерската визија во изминатите децении бележи значителни напредоци кои овозможуваат да се размислува за имплементирање на концепти во код кои дотогаш имаат само теориска основа. Еден од нив е детекцијата на регистарски таблички за кој во прилог на натамошниот труд се нуди можно решение користејќи ги конвенционалните методи на процесирање на слика. Се дава краток осврт на проблемите со кои би се соочил систем кој работи робусно и во реално време, се објаснуваат одлуките зад сопствениот обид, се врши анализа на резултатите и се даваат идеи кон негово подобрување.

I. Вовед

Потребата за систем за автоматска детекција на регистарски таблички е очигледна и се наметнува природно со експоненцијалниот раст на моторни возила низ патиштата и сразмерното зголемување на сообраќајни прекршоци асоцирани со нив, од тривијални намени за пронаоѓање на автомобили кои се движат над препорачаната брзина или не ги почитуваат сообраќајните знаци и за автоматско наплатување на патарини или паркинг места до прилично сериозните во кои е украден автомобилот и е издадена активна потерница за негово запирање, вршење на мониторинг на возило на лице кое се смета дека е потенцијален терорист итн.

Сепак, на успешната имплементација и се закануваат повеќе фактори. Реалната средина како таква е непредвидлива, па детекцијата ќе биде афектирана од тоа колку е јасна сликата (прекршок за брзината со кое се движи возилото често значи работење со заматени слики), атмосферските прилики (сончевиот одблесок доведува до преекспонирана слика, наталожен снег на издадениот дел на браникот, магла која додава ниво на нејаснотија), аголот на кој е направена сликата со што страничен поглед ја вади од центарот на фокус табличката, а премногу висока или ниска поставеност ги воведува сенките како можен предизвик, низок квалитет на камерата значи зрнеста фотографија, справување со шум од околината кој може да претставува возило со своја регистарска табличка, стикери залепени на браникот (често со текст на нив), знаци и банери со правоаголен облик и текст или едноставно премногу дополнителни контури кои ќе го зголемат времето за нивно процесирање. Ако се претпостави дека табличката е успешно изолирана, самите карактери се избени со времето, може да бидат деформирани од удари, а секогаш има прилика досетливиот сопственик на возилото да стави спреј за маскирање на табличката или пластична покривка со леќи за изобличување на буквите и бројките. Естетичниот сопственик пак може да ја стилизира табличката за свое и уживање на совозачот зад него, но е непожелна козметика за коректна сегментација.

Не е сè безнадежно; еден од бенефитите при анализа на регистарски таблички специфично е униформноста на нивниот изглед на национално ниво со прецизно дефинирана ширина и

висина на карактерите кои се истакнуваат зад неутрална позадина. Во ноември 1998, воведен е заеднички формат од страна на Европската Унија за сите земји-членки согласно Виенската Конвенција за патен сообраќај^[1] која налага државните обележја да бидат гравирани на левата страна од табличката или во форма на налепница кои би имале сина позадина, кратенка за државата и препознатливиот круг на ѕвезди на ЕУ над неа; оддесно е регистрацијата со бела или жолта позадина и црни букви со малку изменет стил на фонт за да се избегне повторување. Клучно, одредено е ширината на табличката да биде поголема од нејзината висина. Овој факт ќе биде користен нашироко за да се пресмета очекуван сооднос на сите контури кои можат да содржат табличка во пракса. Затоа, постојат голем број на решенија кои специјализираат во детекција на таблички само за одредена држава користејќи го специфичниот однос, површина и големина на знаци. За решение кое треба да диференцира коректно таблички од различни делови на светот нештата се посложени бидејќи постои голем број на пристапи кои се имаат преземено во поглед на претходно наведените карактеристики. Дополнително, различни таблички се користат за различни видови на возила, дури и во склоп на една држава. Согласно кажаното, позитивен развој на нештата би била глобална стандардизација што во овој момент не изгледа доста веројатно.



Слика 1. Дијаграм на активности преземени во алгоритмите за детекција на регистарски таблички

II. Генерални задачи преземени при детекција на регистарски таблички

Под претпоставка дека на еден или друг начин се добиени слики од возила (преку апарат, инфрацрвена или дигитална камера со висок квалитет, база на слики со возила- непристапна опција за тестирање на програма ако истата не е побарана од државни службеници бидејќи таквите информации се сметаат за сензитивни и можат да нарушат приватност, па отворен репозиториум со таблички засега не постои), следниот логичен чекор е да се направи одреден степен на претпроцесирање на сликата со цел да се минимизира ефектот на споменатите негативни варијабилни (работа со црно-бела слика, заматување, поставување на прагова вредност за пикселите, зголемување на контраст итн.).

Следи сегментирањето на табличката од околината кое може да се изведе преку повеќе пристапи. Доста користена е детекцијата на рабови бидејќи табличките имаат препознатлива правоаголна форма, па може да се бараат контури така што ќе се одредат четирите рабни точки и соодветно даваат одлични резултати. Она што може да биде пречка се рабови кои се испрекинати по примена на филтер или слики кои имаат премногу непотребни контури, па потребно е да се направи деликатен баланс меѓу тоа колку контури можат да се еродираат пред да се компромитира контурата на табличката. За филтрација се применуваат алгоритам за вертикална детекција на рабови (VEDA), генерализирана симетрична трансформација (GST), блок-базиран метод, Canny и убедливо најуспешниот во поглед на мемориска инвестиција спрема добиени резултати- Sobel филтер^[2].

Глобалните карактеристики на сликата пак се користат за пронаоѓање на поврзани објекти чија димензија е слична на табличка и согласно се инваријантни на локацијата на истата во сликата, но не се справуваат најефикасно со лош квалитет и одземаат време. Анализата на поврзани компоненти (CCA) во комбинација со длабоко учење е сепак често избрана опција.

Впечатливата транзиција на боја меѓу карактерите и позадината е причина да се пробаат текстурни карактеристики кои покажуваат значителна робусност во пракса со успешна детекција дури и кога границите се деформирани. Пресметковната комплексност ја извинуваат со најточно однесување од традиционалните методи без комбинирана невронска мрежа преку Gabor филтерот^[3], дискретни Фуриеви трансформации и лизгачки концентрични прозорци (SCW).

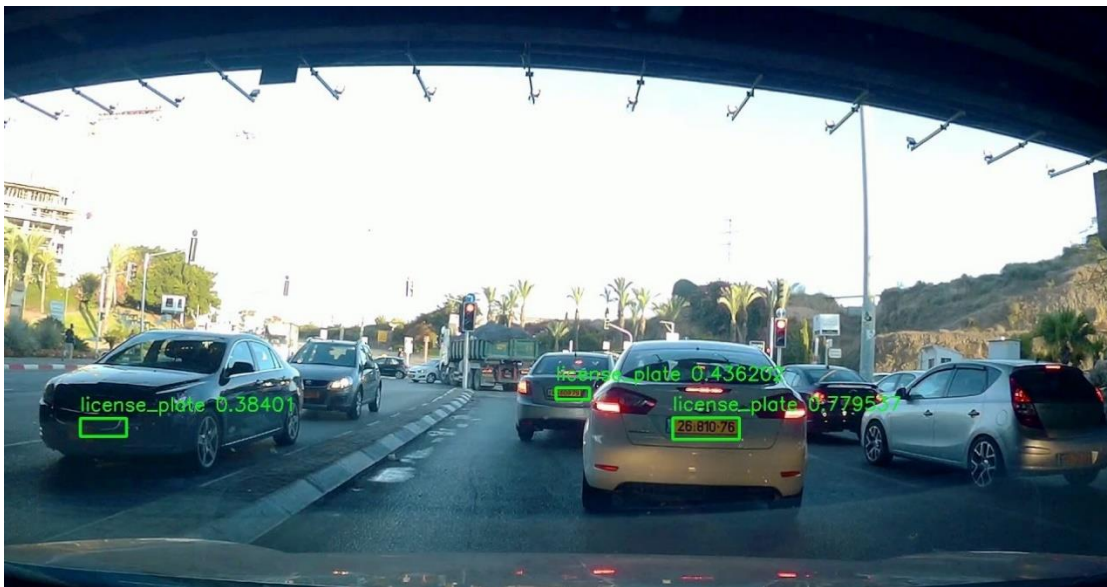
Ползување со карактеристиките на боја е занимлива идеја во принцип ако се земе в предвид споменатата регуларност на палетата бои дозволена на таблички, но сензитивноста на осветленост кај RGB и на шум кај HSL е значителна пречка кон нивно пошироко користење. Алгоритмите за анализа на алфабетски и нумерички региони така што ќе се скенира сликата хоризонтално имајќи ја на ум ширината на знаците во склоп на сликата преминуваат директно кон пребарување за она што е релевантно и од интерес, не се афектирани од ротирали таблички, но можат да бидат залажани од друг текст во околината на возилото.

Имајќи ја диференцираната табличка, се преминува кон сегментација на карактерите. И тука пожелно е да се извршат одредени претпроцесирачки акции како што е корекцијата за

закосеност, трансформација на перспектива, отворање, земање на прагова вредност, заматување, особено ако се работи со ограничени оптички препознавачи на карактери (OCR) бидејќи од квалитетот на сликата ќе зависи колку успешно ќе можат да диференцираат индивидуални карактери (па споени контури се непожелни и препорачливо е да се еродираат). Во често користени техники се вбројуваат анализата на проекциониот профил на табличката која има карактеристична забеста структура од црни пиксели проследени со бели во шема карактер-позадина-карактер, користење на претходно знаење за карактерите што е изводливо бидејќи се знае во колку групи од колку знаци е поделен одреден тип на регистрација и одредување согласно кои пиксели се пополнети за кој знак станува збор.

Финалниот чекор е препознавање на карактерите кое може да се изведе преку стандардни OCR препознавачи на карактери^[4] како што е бесплатниот Tesseract со отворен код развиен од Hewlett-Packard во 80-тите, а одржуван под крило на Google од 2006, чие здобиено искуство се согледува во комерцијалниот Google Cloud Vision API. Microsoft Azure Computer Vision и Abby Cloud се слични решенија преку облак со одредена цена по страница и извесна количина на бесплатни страници на месечно ниво за непретплатените. Навраќајќи се кон системи со отворен код, тука се Kraken и OCRopus за анализа на документи кои се тесно поврзани и ги презентираат резултатите во текстуален документ, а Calamari ползува со моќноста на библиотеката Tensorflow за креирање невронски мрежи.

Најрудиментарниот принцип на функционирање се базира врз обработка на сировите податоци преку шематско поврзување на карактерите со веќе познати темплејти, па затоа најдобро диференцирање се добива кога се предава текст со еден фонт, неротиран, непрекинат и со фиксна големина. Пораспространет пристап меѓу наведените системи е со користење на извлечени карактеристики каде што не се дава подеднаква релевантност на пикселите кои не се клучни во препознавањето.



Слика 2. Автоматска детекција на регистарски таблички во реално време со користење на YOLO (You Only Look Once) конволуциски модел

Посебно внимание и почит може да се даде на машинското учење како алатка при двете фази на детекција и препознавање на карактери кои покажуваат извонредни резултати во сите форми: генетски алгоритми, класификација, регресија, кластерирање и длабоко машинско учење со конволуциски мрежи^[5], значително издигнувајќи го прагот на прецизност со информирано одредување на степенот на влијание на одредени карактеристики во финалната интерпретација на табличката од огромен број на податоци кои се предаваат при процесот на тренирање на моделот.

III. Објаснување на извршената имплементација

Во духот на изучените концепти од предметот, изборот е целосно да се изврши имплементацијата користејќи функционалности од пакетот OpenCV без користење на длабоко машинско учење (ако не се смета вградената невронска мрежа во Tesseract). Изолацијата на регистарската табличка се врши преку детекција на рабови користејќи го Sobel филтерот затоа што е доста распространет и поголема ефикасност се добива компаративно со другите филтри. Претпроцесираните карактери се даваат за сегментација и препознавање на Tesseract OCR бидејќи е пред сè бесплатен и лесен за инсталација. Приказот на имплементацијата ќе се изврши врз пример сликата зададена при изборот на тема.



Слика 3. Влезна слика

Прашалната слика се предава преку ArgumentParser во терминал со цел да се избегне директно менување на кодот (предадениот стринг) на `cv2.imread()`. Следи конверзија од BGR, основниот формат на боја во OpenCV, во HSV и притоа со `cv2.split()` се изолира само каналот V кој дава grayscale приказ (конверзија од `cv2.cvtColor()` со flag `cv2.COLOR_BGR2GRAY` е можно, но почесто во материјалите за консултација одлуката се сведува на HSV бидејќи евентуалните сенки не се дотолку истакнати, односно grayscale сликата е 'понежна'). Ваквата конверзија овозможува намалување на бројот на обработени информации од чекор во чекор со доделување на една вредност за канал по пиксел.



Слика 4. V канал (grayscale слика)

Grayscale сликата се предава на билатерален филтер кој го отстранува шумот, но ги задржува рабовите подобро од median филтер и губи помалку од изостреноста. Дијаметарот на секое соседство пиксели користен при филтрирањето е поставен на 13, сигма филтерот е поставен на 17 за да се постигне поголемо мешање на боите (што значи поголеми површини со приближно иста вредност), истото важи и за сигма просторот.



Слика 5. По применет билатерален филтер

Зголемување на контрастот преку adaptive histogram equalization користејќи CLAHE е користен бидејќи доведува до посуптилен контраст пресметан врз нивоата на сивило во сликата, локални својства и просторни координати на пикселите, па слика која што имала огромно сино небо нема во несразмерна количина да влијае врз сите предели на сцената.

Големината на матрицата која се користи е (8, 8), а clip limit е 2.



Слика 6. Зголемен контраст

За јадро на претстојната морфолошка операција на отворање се зема структурен елемент во форма на елипса со големина (45, 45). Вака добиената отворена слика е одземена од контрастираната слика со што се добива претстава во која се истакнуваат белите делови (каква што е позадината на табличката).



Слика 7. Морфолошко отворање



Слика 8. Резултат од одземање на Слика 6. со Слика 7.

Логички чекор за уште поголемо акцентирање на белите делови е бинаризација со праговата функција `cv2.threshold()`, рангот на вредности е од 0 до 255, а се користат `cv2.THRESH_OTSU + cv2.THRESH_BINARY` бидејќи Otsu најточно би пресметал пожелен глобален праг автоматски наместо мануелно нагодување.



Слика 9. По бинаризација

Се пресметуваат рабовите по хоризонтална и вертикална оска користејќи го Sobel операторот со сите стандардни вредности (и јадро од големина (3,3)). Вака се изолираат најважните рабови кои може да преставуваат регистарска табличка, Canny дава премногу нерелевантни контури додека Prewitt има релативно сличен перформанс со Sobel.



Слика 10. Диференцирани рабови по Sobel

Претпоставувајќи дека контурата на табличката е правилно диференцирана, се бараат контурите со `cv2.findContours()`, се запазува хиерархијата бидејќи се бараат со flag `cv2.RETR_TREE`, а се пресметуваат само клучните точки со додавање на аргумент `cv2.CHAIN_APPROX_SIMPLE` и се сортираат по големина и тоа по опаѓачки редослед. Потоа, за секоја контура се пресметува правоаголната рамка во која истата целосно е опфатена со `cv2.boundingRect()`, а меѓу параметрите кои ги враќа е (x,y) координати на една од рабните точки, широчина и висина на правоаголникот. Ако пресметаниот однос од добиените вредности (ширина поделено со висина) се наоѓа измеѓу 3 и 5, се изолира правоаголникот од grayscale сликата и се чува во листа на потенцијални регистарски таблички. Волкава слобода на варијација за односот се дозволува бидејќи е непознато каков тип на табличка ќе биде доставена за анализа, но голем број на таблички сепак се наоѓаат во овој интервал. Воедно, `boundingRect()` не го исцртува минималниот правоаголник за дадената контура, па се дава простор за грешка во апроксимацијата.

Неконвенционално, но со надеж дека ќе се зголеми прецизноста, се користи SIFT алгоритам за детекција на карактеристики за да се изолира вистинската слика со табличка од листата на потенцијални. Идејата е да се споредува во циклус секоја од нив со претходно избрани слики со регистарски таблички од различни држави и со различни формати, да се одреди бројот на совпаѓања на прашалната слика со сите вистински таблички и се пресмета

просечното совпаѓање. Онаа слика која што имала најголем број на совпаѓања ќе се смета за регистарска табличка.

За таа цел, од фолдерот `Standard_plates` кој содржи слики од европски, британски, македонски, кинески, индиски и американски таблички се вчитуваат истите во `grayscale` формат. Се креира SIFT трансформатор и се пресметуваат клучни точки и дескриптори за нив и потенцијалните слики. Се креира Flann совпаѓач и се иницијализира променлива `licensePlate` на `None` и `max_average_matching` на `-1`. Потоа за секоја потенцијална слика се пресметуваат совпаѓања користејќи ги стандардните препорачани вредности од Lowe: од совпаѓањата со метод на `k nearest neighbours` за `k=2` се филтрираат оние чие растојание од првиот најдобар `match` до вториот е поголемо до `0.7`, се пресметува маската преку процес на RANSAC хомографија и се наоѓаат `inliers`, односно оние совпаѓања за кои маската враќа вредност `1`. Пресметаниот просек на совпаѓања откако истиот процес ќе се примени со секоја слика од `Standard_plates` се споредува со дотогаш најдениот максимален број на совпаѓања. Ако е поголем тоа индиректно ќе се земе да значи дека е поверојатно да е слика со табличката, па се чува во `licensePlate` и се нагудува `max_average_matching` на новата вредност. Во примерот, сликата со најголем број совпаѓања е прикажана подолу.



Слика 11. Изолуирана регистарска табличка

Добиениот регион со табличка потоа се бинаризира користејќи `cv2.THRESH_BINARY_INV` во `bitwise or` комбинација со Otsu за автоматски да се пресмета прагот, а да се добијат бели букви на црна позадина.



Слика 12. Бинаризирана

Ако бинаризираната слика се предаде на функцијата `clear_border` од пакетот `skimage.segmentation`, таа ќе ги отстрани сите контури кои ја допираат границата на сликата што е одличен начин за елиминирање на другите делови од табличката кои не се од интерес или во слики каде што добиената слика со табличка опфаќа и дел од браникот.



Слика 13. По примена на `clear_border`

За поуспешно парсирање, сликите кои се предаваат пожелно е да се поголеми од `300 dpi`, затоа се зголемува сликата `1.2` пати во двете насоки. Шумот околу буквите се еродира со стандардното јадро дадено од `cv2.erode()` и дилатира еднаш со `cv2.dilate()` за да се зголеми дебелината на буквите на истиот начин.



Слика 14. По ерозија



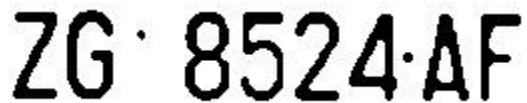
Слика 15. По дилатација

Дотогаш направените операции често доведуваат до нерамна површина на буквите со што се добиваат скалести рабови кои можат да попречат во препознавањето. `cv2.medianBlur()` со јадро (3,3) ќе ги пополни ‘скаличките’ со средна вредност од околните, што значи дека ќе имаат сива обоеност. За да се одржи бинарноста, повторно се користи `cv2.threshold()` со истите flags како и при минатото користење. Дополнително, сликата е појасна и границите на буквите се позаоблени.



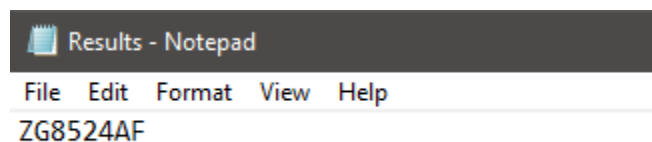
Слика 16. Резултат од `medianBlur + threshold`

Бидејќи се работи со 4.0.0.20190314 верзија на Tesseract и препорачливо е буквите да се темни врз светла позадина, за последен пат се користи `cv2.threshold()`, но овојпат само со `cv2.THRESH_BINARY_INV`.



Слика 17. Влезна слика за Tesseract OCR

Вака издвоените карактери се предаваат на `pytesseract.image_to_string()` при што се специфицира дека ќе се работи со алфабетичката репрезентација на англиската азбука и нумеричките цифри од 0-9 во `-psm 7` мод (се анализира една линија на текст). Добиениот стринг потоа се чисти од сите не-алфанумерички знаци кои можат да се добијат од преостанати остатоци на непотребни контури според ASCII вредност. На крај се отвора текстуалната датотека `Results.txt` во мод за `append`, се запишува финалниот стринг и се затвора.

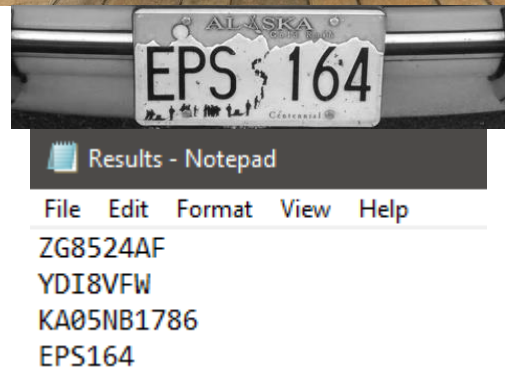
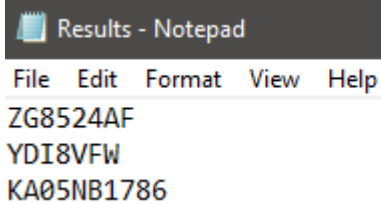
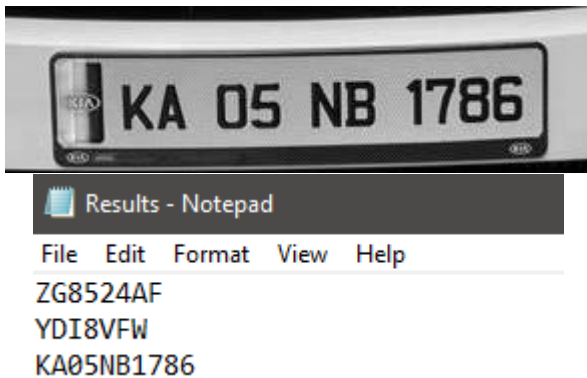
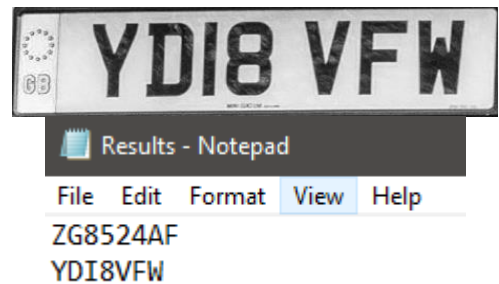


Слика 18. Конечен стринг

Може да се забележи точно препознавање на сите карактери од регистарската табличка.

IV. Коментирање на резултатите

Генерално гледано, ако како влезна слика се даде автомобил поставен во иста линија со камерата така што табличката е во целосен поглед, голема е веројатноста за точна изолација. Помага ако квалитетот и големината на сликата се солидни бидејќи мали таблички го зголемуваат процентот на грешно препознавање со OCR. Сепак, Tesseract одлично се справува со нејзина неидеална изолација, па дури и изместена перспектива не претставува особен проблем. Во прилог се покажуваат успешни излези, по еден за секој тип даден при SIFT совпаѓање од фолдерот стандардни таблички. Да се напомене дека се работи само со англиска азбука, па карактерот од кинеското писмо е интерпретиран како тројка во последниот пример.





Results - Notepad

File Edit Format View Help

ZG8524AF

YDI8VFW

KA05NB1786

EPS164

SU9354AB



Results - Notepad

File Edit Format View Help

ZG8524AF

YDI8VFW

KA05NB1786

EPS164

SU9354AB

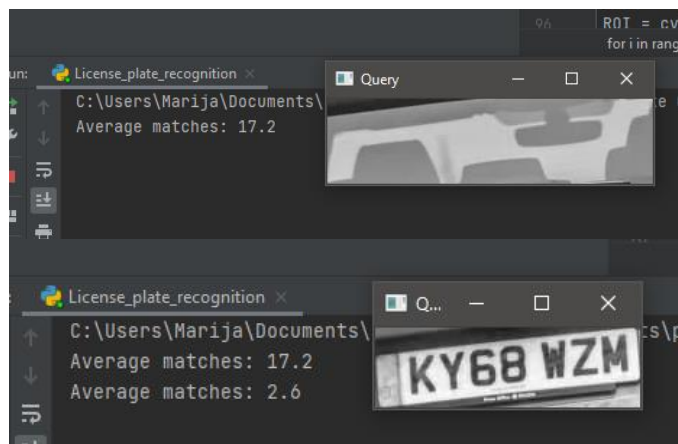
4138898

Не секогаш се забележува очекуваното однесување. Бидејќи се дозволува огромна варијација на влезови врз кои се прават прилично деструктивни операции, можно е спојување на контури или комплетно еродирање на клучната контура, елиминирајќи го од старт точното диференцирање.



Слика 19. Неуспех како резултат на губење на контурата на табличката

Друг збунувачки ефект се забележува при совпаѓањето со SIFT, при што слики кои немаат табличка во неа успеваат да добијат поголем број на просечни совпаѓања во споредба со сликите каде што е прикажана вистински табличката. Не може да се даде специфична причина зад ваквата препрека, но може да се забележи преференца кон сликите со огледалата или ветробранот.



Слика 20. Поголем број на просечни совпаѓања со SIFT се добиваат за ветробранското стакло наместо самата табличка

OCR-от на успешно изолирана табличка е некогаш оневозможена бидејќи сликата има мала големина, па сликата од табличката има премногу мали димензии за да ги издржи претпроцесирачките активности.



Слика 21. Релативно успешно изолираната табличка понатаму се губи

Секако, можно е сè до парсирањето на карактерите да се одвива непречено, но да не се препознаат карактерите правилно, што е редок, но забележан случај. Исто така, од експериментирање со разни влезови, установено е дека дијагоналната наклонетост на возилото, односно ситуации во кои е сликано настрана или е наведено во некој акциски трик значително ја намалуваат можноста за правилно одвивање во било која од претходно дискутираните точки на слабост.



SK 1016 AM
SK 1016 AM

```
Results - Notepad
File Edit Format View Help
ZG8524AF
YDI8VFW
KA05NB1786
EPS164
SU9354AB
4138898|
016AM
```

Слика 22. Неуспех при препознавање на првите три карактери со Tesseract

Она што може да се направи за подобрување на прецизноста на изнесената имплементација е ограничено. Вредностите за кои беше изнесено дека се користат во сите функции се добиени преку незавиден број на часови поминати минимално нагодувајќи и тестирајќи ја покажаната ефикасност со нив. Традиционалните имплементации согласно кажаното секогаш ќе страдаат од непредвидени влезови за кои истите тие вредности се максимално разорувачки или недоволни за добра пресметка. Единствена препорака е да се користи машинско учење во двете фази на изолација на табличката и препознавање на карактери бидејќи само така може да се добие посакуваната робусност.

V. Заклучок

Од сето што беше изложено јасно може да се увиди дека детекцијата на регистарски таблички и препознавање на карактери бележи позитивен тренд на подобрување со сериозни аргументи кон нивна масовна комерцијална употреба за зголемување на сигурноста на сите учесници во сообраќајот и потпомагање на економската ситуираност на државните регулаторни органи, а напредоците во компјутерската визија овозможуваат да се размислува низ повеќе авени и комбинирање на разни методологии, со што дури и почетничките обиди налик приложениот произведуваат скромни, но сепак значителни успеси.

VI. Референци

- [1] 19. Convention on Road Traffic. Vienna, 8 November 1968, Chapter XI. Transport and Communications B. Road Traffic, VOL-1
- [2] Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Psoroulas, I.D., Loumos, V. and Kayafas, E., 2008. License plate recognition from still images and video sequences: A survey. IEEE Transactions on intelligent transportation systems, 9(3), pp.377-391.
- [3] Mufti, N. and Shah, S.A.A., 2021. Automatic number plate Recognition: A detailed survey of relevant algorithms. Sensors, 21(9), p.3028.
- [4] Ted Han, Amanda Hickman, Source, 2019. Our search for the best OCR Tool, and what we found [Accessed 21 June, 2021]
- [5] Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.S., Gonçalves, G.R., Schwartz, W.R. and Menotti, D., 2018, July. A robust real-time automatic license plate recognition based on the YOLO detector. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-10). IEEE.