

Извештај од изработен проект на тема

**Детекција на број на лична карта и возачка дозвола**

Предмет: Дигитално Процесирање на Слика

**Јуни 2021**

Ментор:

Д-р. Ивица Димитровски

Изработиле:

Анастасија Петровска 181059

Вероника Огњановска 181045

Содржина

[**Абстракт** 3](#_Toc73967351)

[**1.** **Вовед** 3](#_Toc73967352)

[**2.** **Методологија** 4](#_Toc73967353)

[**Python** 4](#_Toc73967354)

[**Користени библиотеки** 4](#_Toc73967355)

[**OpenCv** 4](#_Toc73967356)

[**NumPy** 4](#_Toc73967357)

[**Mathplotlib и mathplotlib.pyplot** 4](#_Toc73967358)

[**Glob** 4](#_Toc73967359)

[**Os** 5](#_Toc73967360)

[**Imutils** 5](#_Toc73967361)

[**3.** **Решение на проблемот** 5](#_Toc73967362)

[**4.** **Резултати** 5](#_Toc73967363)

[**5.** **Дискусија** 5](#_Toc73967364)

[**6.** **Литература** 5](#_Toc73967365)

# **Абстракт**

Автоматизацијата на работата на човекот е познат проблем во областа на детекција на броеви на лични документи, како лична карта и возачка дозвола. Ова се решава преку дигитално процесирање на скенирана слика од документот и препознавање на бројот на документот со помош на шаблон. Изборот на шаблон е клучен дел во ова. Целта на овој труд е да го опфати решението на прикажаниот проблем и да даде критички осврт врз истото.

# **Вовед**

Информациската технологија се развива побрзо од кога и да било, и во теорија и во пракса. Во последно време, истражувачите се повеќе се фокусираат кон тоа да се изврши оптимизација на работата на човекот преку компјутерски програми.

Личните податоци на граѓаните на една држава можат да се добијат преку пребарување на бројот на нивната лична карта од страна на овластените служби. Претходно, внесувањето на овој број се извршуваше рачно, што не е ефикасен процес бидејки е потребно многу време за да се изврши. Заради ова, се јавува потреба од систем кој што процесот ќе го извршува на побрз и автоматизиран начин.

Познат пример за ова е автоматизирање на работата на полициските службеници на граничните премини со тоа што наместо рачно внесување на бројот на документот за идентификација на граѓаните, се користи скенирање на самиот документ и автоматско препознавање во минимален временски период. Дополнително, со новиот начин на живот по Корона вирусот, повеќето од работите кои еден просечен граѓанин ги извршуваше преку шалтерскиот систем, сега можат да се извршуваат во домашни услови. Примери за потреба од компјутерско препознавање на личен документ се одредени банкарски услуги кои можат или би можеле да се извршат преку компјутер, аплицирање за работно место, патување во странство и многу други.

The development of Information Technology has developed quite rapidly, both in theory and

application. A lot of research technology has used to facilitate and accelerate human works. The

researches have been implemented to computer and used to accomplishing human works optimally.

One example of the development of information technology in business is how to purchase goods.

Currently, we don’t have to visit the store for purchasing some goods. Purchase of goods can also be

done by online. In various businesses, companies need customer data that should be inputted into

database for online or offline purchase. Data of customers who buy item by online are usually

requested when registering an account, while customers who buy item by offline are usually asked to

get their identity. Data of costumer’s identity can be obtained from their ID Card. The ID Card that

used for this case is citizen ID Card. Previously, customers data inputted manually. That is not

efficient process because we need a lot of time to input data one by one. Therefore, we need a system

that processes automatically.

Based on that problem, Image Processing technique can be used as an alternative solution of

manually input process. This process starts by extracting information in ID Card image. Then, it will

be pre-processed to obtain the necessary part of image. Furthermore, Optical Character Recognition

(OCR) will be performed in order to recognize text in images. OCR can recognize handwriting and

text characters automatically through optical mechanism. OCR is designed to process images

consisting of text with little non-text data interference. While the OCR performance depends on the

quality of the inputted document [1].

Based on some research above, this study compares the result of character recognition of name and

NIK (identity number) in ID Card using two different tesseract models. The first model uses the train

data manually that created from five ID Card as data set and training on tesseract 3.05 with the support

of software QT-box version 1.08. While the second model uses train data that already contained in

tesseract 4.0, which is a data train that contains text data in Indonesian language with different font

The development of Information Technology has developed quite rapidly, both in theory and

application. A lot of research technology has used to facilitate and accelerate human works. The

researches have been implemented to computer and used to accomplishing human works optimally.

One example of the development of information technology in business is how to purchase goods.

Currently, we don’t have to visit the store for purchasing some goods. Purchase of goods can also be

done by online. In various businesses, companies need customer data that should be inputted into

database for online or offline purchase. Data of customers who buy item by online are usually

requested when registering an account, while customers who buy item by offline are usually asked to

get their identity. Data of costumer’s identity can be obtained from their ID Card. The ID Card that

used for this case is citizen ID Card. Previously, customers data inputted manually. That is not

efficient process because we need a lot of time to input data one by one. Therefore, we need a system

that processes automatically.

Based on that problem, Image Processing technique can be used as an alternative solution of

manually input process. This process starts by extracting information in ID Card image. Then, it will

be pre-processed to obtain the necessary part of image. Furthermore, Optical Character Recognition

(OCR) will be performed in order to recognize text in images. OCR can recognize handwriting and

text characters automatically through optical mechanism. OCR is designed to process images

consisting of text with little non-text data interference. While the OCR performance depends on the

quality of the inputted document [1].

Based on some research above, this study compares the result of character recognition of name and

NIK (identity number) in ID Card using two different tesseract models. The first model uses the train

data manually that created from five ID Card as data set and training on tesseract 3.05 with the support

of software QT-box version 1.08. While the second model uses train data that already contained in

tesseract 4.0, which is a data train that contains text data in Indonesian language with different font

The development of Information Technology has developed quite rapidly, both in theory and

application. A lot of research technology has used to facilitate and accelerate human works. The

researches have been implemented to computer and used to accomplishing human works optimally.

One example of the development of information technology in business is how to purchase goods.

Currently, we don’t have to visit the store for purchasing some goods. Purchase of goods can also be

done by online. In various businesses, companies need customer data that should be inputted into

database for online or offline purchase. Data of customers who buy item by online are usually

requested when registering an account, while customers who buy item by offline are usually asked to

get their identity. Data of costumer’s identity can be obtained from their ID Card. The ID Card that

used for this case is citizen ID Card. Previously, customers data inputted manually. That is not

efficient process because we need a lot of time to input data one by one. Therefore, we need a system

that processes automatically.

Based on that problem, Image Processing technique can be used as an alternative solution of

manually input process. This process starts by extracting information in ID Card image. Then, it will

be pre-processed to obtain the necessary part of image. Furthermore, Optical Character Recognition

(OCR) will be performed in order to recognize text in images. OCR can recognize handwriting and

text characters automatically through optical mechanism. OCR is designed to process images

consisting of text with little non-text data interference. While the OCR performance depends on the

quality of the inputted document [1].

Based on some research above, this study compares the result of character recognition of name and

NIK (identity number) in ID Card using two different tesseract models. The first model uses the train

data manually that created from five ID Card as data set and training on tesseract 3.05 with the support

of software QT-box version 1.08. While the second model uses train data that already contained in

tesseract 4.0, which is a data train that contains text data in Indonesian language with different font

Целта на овој извештај од изработен проект е да се прикаже процесот на екстракција и детекција на број на лична карта и возачка дозвола. Имплементацијата на кодот е во програмскиот јазик Python. Исто така, прикажан е и процесот на креирање на шаблон (анг. “pattern”) кој е потребен при препознавање на бројот.

Во овој труд подетално се прикажани алатките кои се употребени и како тие функционираат, детално објаснување на чекорите преку кои се пишува кодот како решение на проблемот. Прикажани се и резултатите од програмата за детекција и дискусија преку која се интерпретираат резултатите, ограничувањата и сугестии за во иднина. На крај, преставена е користената литература.

# **Методологија**

Овој проект беше изработен во програмскиот јазик Python во работната околина JetBrains PyCharm.

### **Python**

Python е програмски јазик од високо ниво кој има генерална цел на користење. Поддржува повеќе програмски парадигми како структурно, објектно – ориентирано и функциско програмирање. Овој програмски јазик овозможува брзо работење и уште побрза интеграција на системите. Python има широка примена, како на пример во развој на веб апликации, дизајнирање на графички кориснички интерфејс (анг. GUI), машинско учење и систем администрација.

## **Користени библиотеки**

### **OpenCv**

OpenCv e библиотека во програмскиот јазик Python која се користи за различни видови на анализа и процесирање на слики и видеа, како детекција и препознавање на лица, читање на регистарски таблички на возила, едитирање на слики, напредна роботска визија, оптичко препознавање на карактери и уште многу други. Оваа библиотека најголема примена наоѓа во машинското учење и компјутерската визија во реално време.

### **NumPy**

NumPy е Python библиотека која овозможува работа со низи и се стреми да овозможи објект низа кој е до педесет пати побрз од традиционалната Python листа. Низите од оваа библиотека најмногу се користат во науката за податоци (анг. Data science) каде што брзината и ресурсите се од големо значење. Дополнително, се користи и во други области како на пример во линеарна алгебра, фуриева трансформација и матрици.

### **Mathplotlib и mathplotlib.pyplot**

Mathplotlib е библиотека која овозможува исцртување графици од податоците во таканаречени фигури кои можат да бидат прозорци, Jypiter алатки и слично. Ваквиот вид графици во оваа библиотека се нарекуваат plots и преку неа можат да се вградат во различни алатки за графички кориснички интерфејс.

Mathplotlib.pyplot е колекција од функции кои прават библиотеката Mathplotlib да работи како алатката MATHLAB. Секоја pyplot функција прави некоја промена на фигурата, како на пример креира фигура, креира област за plot во фигурата, декорира со различни лабели и наслови и слично.

### **Glob**

Оваа библиотека се користи за преземање на документи или патишта до документи преку специфичен шаблон. Правилата за шаблонот се водат според стандардните правила во Unix оперативниот систем.

### **Os**

Модулот os се користи за различни интерфејси на оперативни системи, интеракција со нив и овозможува користење на функционалности кои зависат од оперативниот систем. Во овој проект најкористена функција од оваа библиотека е функцијата os.path.join() која интелегентно спојува една или повеќе компоненти на еден пат до документ.

### **Imutils**

Библиотеката imutils содржи серија корисни функции кои извршуваат основни функции за процесирање на слики како транслација, ротација, менување на големината на сликата, скелетонизација, приказ на слики направени со библиотеката mathplotlib, сортирање на контури, детектирање на рабови и многу други.

# **Решение на проблемот**

## **Креирање на шаблон**

Бидејки овој проект е наменет за препознавање на број на лична карта и возачка дозвола, документи кои се со различен изглед во различни држави, пронаоѓање на веќепостоечки шаблон беше невозможна опција. Како решение на ова, во продолжение е претставен детален опис за креирање на ваков шаблон со кој подоцна ќе се препознае бројот на документот.

За почеток, искористени се веќепостоечки намалени слики од цифри и букви кои ги има на документите. На секоја слика има само една цифра или само една буква.

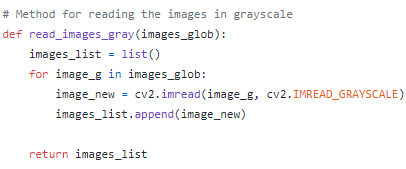
Овие слики се преземаат преку нивниот пат до папката во која се наоѓаат преку следниот код.



Потоа, сликите се читаат во црно бел формат со повикување на функцијата read\_images\_gray.

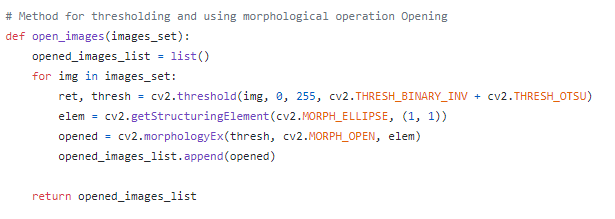


Оваа функција ги зема сите слики од дадената база и секоја од нив ја чита во црно бел формат и ја додава на листата images\_list. Оваа листа се враќа и се зачувува под името pattern\_images\_gray како на сликата подоле. Кодот за функцијата е прикажан на сликата подолу.

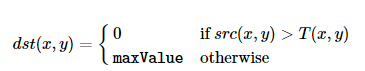


Следно, на листата прочитани слики се повикува функцијата open\_images. Ова е прикажано на двете слики подолу.





Оваа функција ги зема сликите како аргумент и врз секоја од нив применува Otsu трешхолдирање(анг. tresholding) и бинарно инвертирање. Ова се изведува според следната формула.



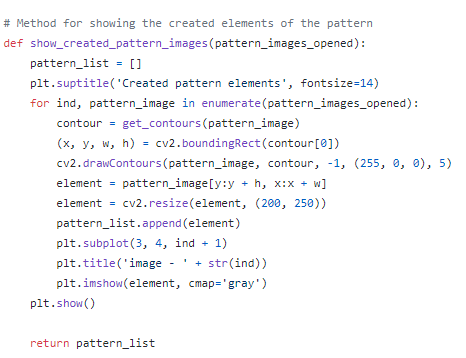
T(x,y) е внимателно пресметан трешхолд за секој пиксел. Во оваа формула, вредноста на пикселот се менува во 0 доколку првично била над T(x,y), а во спротивниот случај се зема максималната вредност.

Потоа, врз сликите се применува и морфолошка операција отворање со кое се овозможува отстранување на непотребните шумови кои се во позадината на сликата.

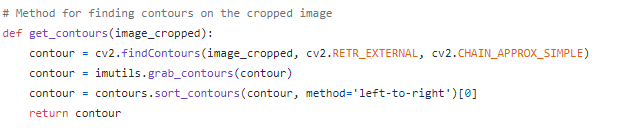
Овие функции се достапни во библиотеката opencv.

Следно, за проверка на резултатите се повикува функцијата show\_created\_pattern\_images.



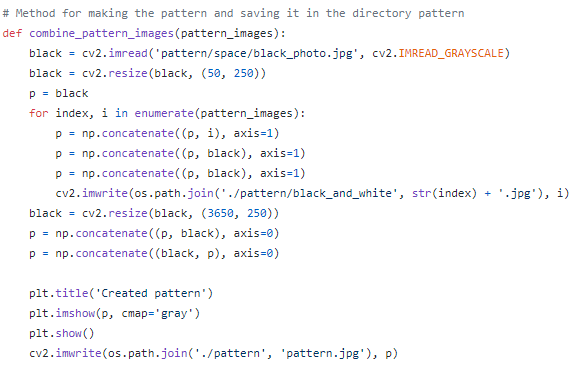


Дополнително во оваа функција се повикува и функцијата get\_contours која овозможува исцртување на контури на секоја од сликите. Со ова се заоблуваат сите нерамни рабови и цифрите и буквите добиваат подобар изглед кој при препознавањето дава подобри резултати. Функцијата за контури е прикажана подолу.



По извршувањето на функцијата show\_created\_pattern\_images се добива листа од слики врз кои се направени сите потребни трансформации за препознавањето преку шаблон. Единствено што треба да се доправи е креирањето на самиот шаблон преку спојувањето на сликите од листата. Ова е прикажано на двете слики подолу.





Во функцијата combine\_pattern\_images најпрвин се менува големината на секоја од трансформираните слики во 50,250. Дополнително се повикува и една црна слика со помала должина која ќе се искористи како простор меѓу цифрите и буквите во шаблонот. Првата цифра се спојува со црната слика, а потоа на овој спој наизменично се додаваат слика од друга цифра или буква и црната слика. Ова се прави преку повикување на функцијата concatenate() од библиотеката NumPy.

# **Резултати**

# **Дискусија**

# **Литература**

<https://www.python.org/>

<https://en.wikipedia.org/wiki/Python_(programming_language)>

<https://pythonprogramming.net/loading-images-python-opencv-tutorial/>

<https://en.wikipedia.org/wiki/OpenCV>

<https://www.w3schools.com/python/numpy/numpy_intro.asp>

<https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py>

<https://en.wikipedia.org/wiki/Matplotlib>

<https://matplotlib.org/stable/tutorials/introductory/pyplot.html>

<https://www.geeksforgeeks.org/how-to-use-glob-function-to-find-files-recursively-in-python/#:~:text=In%20Python%2C%20the%20glob%20module,to%20match%20pathnames%20in%20directories>.

<https://docs.python.org/3/library/os.html>

<https://www.geeksforgeeks.org/python-os-path-join-method/>

<https://pypi.org/project/imutils/>