Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и

информатики»

(СибГУТИ)

09.03.01 Информатика и вычислительная техника

Профиль: Программное обеспечение средств вычислительной техники и автоматизированных систем

(очная форма обучения)

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

на *кафедре прикладной математики* и кибернетики

(наименование профильной организации/структурного подразделения СибГУТИ)

Распознавание автомобильных номеров в видеофайлах

Выполнил:

студент ИВТ

гр. ИП-014 / Обухов А.И.

«27 мая» 2023г. (подпись)

Проверил:

Руководитель от СибГУТИ / Приставка П.А.

«27 мая» 2023г. (подпись)

Новосибирск 2023 г.

**План-график проведения**  производственной  **практики**

Вид практики

Обухова Артема Игоревича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Фамилия Имя Отчество студента

института Информатики и вычислительной техники , 3 курса, гр. ИП-014

Направление: 09.03.01 – Информатика и вычислительная техника

Код – Наименование направления (специальности)

Профиль: Программное обеспечение средств вычислительной техники и автоматизированных систем

Место прохождения практики *Кафедра прикладной математики* и кибернетики

Объем практики: **360/10** часов/ЗЕ

Вид практики  ***производственная***

Тип практики Производственная практика - Практика по получению профессиональных умений и опыта профессиональной деятельности

Срок практики с "30" января 2023 г.

по "27" мая 2023 г.

Содержание практики\*:

| Наименование видов деятельности | Дата (начало – окончание) |
| --- | --- |
| 1. Общее ознакомление со структурным подразделением предприятия, вводный инструктаж по технике безопасности | 30.01.2023-01.02.2023 |
| 2. Выдача задания на практику, деление студентов на группы (если необходимо), определение конкретной индивидуальной темы, формирование плана работ | 02.02.2023-04.02.2023 |
| 3. Работа с библиотечными фондами структурного подразделения или предприятия, сбор и анализ материалов по теме практики | 06.02.2023-11.02.2023 |
| 4. Выполнение работ в соответствии с составленным планом:   1. Сбор материала для тестирования 2. Реализация программного продукта | 13.02.2023-20.05.2023 |
| 5. Тестирование полученной программы, анализ производственной работы  Составление отчета по практике, защита отчета | 22.05.2023-27.05.2023 |

\*В соответствии с программой практики

Руководитель от СибГУТИ / Приставка П.А./

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023г. (подпись)

### ОГЛАВЛЕНИЕ

[ЗАДАНИЕ НА ПРАКТИКУ](#_heading=h.q2dx0marar9h) 3

[ВВЕДЕНИЕ](#_heading=h.2ngz5aiajs42) 4

[ПОСТАНОВКА ЗАДАЧИ](#_heading=h.49npeb7lzpmr) 5

[ВЫБОР ПРОГРАММНЫХ СРЕДСТВ](#_heading=h.fqpqik7y130l) 6

[ПРОГРАММНАЯ РЕАЛИЗАЦИЯ](#_heading=h.rgvv9ppoxlrh) 7

[ОПИСАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ](#_heading=h.a07ftac8geej) 9

[ЗАКЛЮЧЕНИЕ](#_heading=h.pefmlphfb7q2) 11

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ](#_heading=h.iyxgn27p3c2) 12

[ПРИЛОЖЕНИЕ](#_heading=h.ujjac6ir19zz) 13

### 

### ЗАДАНИЕ НА ПРАКТИКУ

Изучить необходимый инструментарий и реализовать рабочую программу для распознавания автомобильных номеров в видеофайлах.

Для реализации будут использованы:

* Python — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении (ML)
* OpenCV — это open-source библиотека компьютерного зрения, которая предназначена для анализа, классификации и обработки изображений. Широко используется в таких языках как C, C++, Python и Java.
* EasyOCR — это модуль Python для извлечения текста из изображения.

### ВВЕДЕНИЕ

OCR (англ. optical character recognition) — технология автоматического анализа текста и превращения его в данные, которые может обработать компьютер. Человек распознает символы с помощью глаз и мозга. Компьютер использует камеру сканера, которая создает графическое изображение текстовой страницы. Для компьютера нет разницы между сканом текстового документа и фотографией: и то, и другое — набор пикселей.

Человек легко поймет, что на картинке изображен текст. Для компьютера есть два способа: распознавать символы целостно, то есть распознавать паттерн, или выделять отдельные черты, из которых состоит символ — выявлять признаки.

В 1968 году компания American Type Founders, которая с конца XIX века занималась созданием печатных шрифтов, придумала шрифт OCR-A с буквами одинаковой ширины. В основном шрифт использовали в банковских чеках, а для его чтения компьютером было создано специальное программное обеспечение.

Поскольку шрифт был стандартизирован, его распознавание стало относительно простой задачей. Программы «знали» OCR-A и могли переводить информацию с чеков в машиночитаемую форму. Однако в случае ошибки человек мог вмешаться в процесс и также прочитать банковский чек. Следующим шагом стало обучение программ OCR распознавать символы еще в нескольких самых распространенных шрифтах.

### ПОСТАНОВКА ЗАДАЧИ

Разработать программу, которая будет считывать видео-файл, распознавать в нем номерные знаки машин и выдавать картинки с распознанным данными.

### ВЫБОР ПРОГРАММНЫХ СРЕДСТВ

PyCharm - это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA.

EasyOCR - Готовое распознавание текста с более чем 80 [поддерживаемыми языками](https://www.jaided.ai/easyocr) и всеми популярными письменными средствами, включая: латиницу, китайский, арабский, деванагари, кириллицу и т.д.

OpenCV - библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом.

### ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Для начала видеофайл разбивается на отдельные изображения для удобства работы с библиотеками с помощью split\_video\_into\_frames(path: str).

После успешного разделения видеофайла в отдельные кадры начинается обход всех полученных кадров и процесс распознавания номеров.

Мы преобразуем наше изображение в градации серого, удаляем шум, применяя гауссово размытие, и, наконец, определяем края с помощью детектора краев Кэнни.

*gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)*

*blur = cv2.GaussianBlur(gray, (5, 5), 0)*

*edged = cv2.Canny(blur, 10, 200)*

Далее, мы н находим контуры на изображениях и выделяем контуры номеров, и затем мы проходим по контурам, аппроксимируем каждый из них, и если мы нашли контур с четырьмя точками, то помечаем его как обнаруженный номер

*contours, \_ = cv2.findContours(edged, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)*

*contours = sorted(contours, key=cv2.contourArea, reverse=True)*

*approximations = []*

*for c in contours:*

*peri = cv2.arcLength(c, True)*

*approx = cv2.approxPolyDP(c, 0.02 \* peri, True)*

*if len(approx) == 4:*

*approximations.append(approx)*

*break*

*if len(approximations) == 0:*

*continue*

Теперь мы готовы к распознанию номерного знака с помощью библиотеки EasyOCR

*pattern = re.compile(r"^[^+\_=/\*?@#$%&()'\"|â„;:{}.,`~<>}{][^\\]{1,20}$")*

*for approx in approximations:*

*(x, y, w, h) = cv2.boundingRect(approx)*

*icense\_plate = gray[y:y + h, x:x + w]*

*result = reader.readtext(license\_plate)*

*for detection in result:*

*if detection[2] > 0.5:*

*license\_plate\_text = str(detection[1]).replace(' ', '').upper()*

*if pattern.match(license\_plate\_text):*

*print(license\_plate\_text)*

*plates.add(license\_plate\_text)*

*text = str(license\_plate\_text)*

*cv2.putText(image, text, (x, y - 20), cv2.FONT\_HERSHEY\_COMPLEX, 0.75, (0, 255, 0), 2)*

*if not os.path.exists('detections'):*

*os.makedirs('detections')*

*cv2.imwrite(os.getcwd() + '/detections/' + str(license\_plate\_text) + '.jpg', image)*

### ОПИСАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

****

Пример 1. Видео с YouTube

****

Пример 2.

****

Пример 3.

****

Пример 4.



Пример 5. Пример с собственноручно снятого видео.

### ЗАКЛЮЧЕНИЕ

В ходе работы над проектом я расширил свои познания в области языка программирования Python, изучил основные функции библиотеки OpenCV, ознакомился с основами модуля EasyOCR.

В результате была полученная программа на языке программирования Python для распознавания автомобильных номеров на видеофайлах с использованием библиотек OpenCV и EasyOCR.

Данная программа может быть использована в различных областях, а также может быть серьезно модифицирована, ведь распознавание текста с видео-файла достаточно полезная функция. Такой функционал может стать как дополнением к основной программе, так и центральной идеей для других проектов.

### СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. <https://opencv.org/> документация (дата обращения 01.02.2023)
2. <https://pypi.org/project/easyocr/> документация (дата обращения 01.02.2023)
3. [https://habr.com/ru](https://habr.com/ru/articles/678260/) OpenCV — быстрый старт: начало работы с изображениями (дата обращения 05.02.2023)

### ПРИЛОЖЕНИЕ

**ЛИСТИНГ**

import cv2

import os

import easyocr

import re

def split\_video\_into\_frames(path: str):

try:

if not os.path.exists('images'):

os.makedirs('images')

except OSError:

print('Error creating directory')

cam = cv2.VideoCapture(path)

currentFrame = 0

while True:

cam.set(cv2.CAP\_PROP\_POS\_MSEC, currentFrame \* 1000)

ret, frame = cam.read()

if not ret:

return

name = './images/frame' + str(currentFrame) + '.jpg'

print('Creating %s' % name)

cv2.imwrite(name, frame)

currentFrame += 1

if \_\_name\_\_ == '\_\_main\_\_':

split\_video\_into\_frames('video.mp4')

images\_path = os.getcwd() + '/images/'

images = os.listdir(images\_path)

plates = set()

reader = easyocr.Reader(['ru'])

for image\_file in images:

image\_file = images\_path + image\_file

image = cv2.imread(image\_file)

os.remove(image\_file)

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (5, 5), 0)

edged = cv2.Canny(blur, 10, 200)

contours, \_ = cv2.findContours(edged, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

contours = sorted(contours, key=cv2.contourArea, reverse=True)

approximations = []

for c in contours:

peri = cv2.arcLength(c, True)

approx = cv2.approxPolyDP(c, 0.02 \* peri, True)

if len(approx) == 4:

approximations.append(approx)

break

if len(approximations) == 0:

continue

pattern = re.compile(r"^[^+\_=/\*?@#$%&()'\"|â„;:{}.,`~<>}{][^\\]{1,20}$")

for approx in approximations:

(x, y, w, h) = cv2.boundingRect(approx)

license\_plate = gray[y:y + h, x:x + w]

result = reader.readtext(license\_plate)

for detection in result:

if detection[2] > 0.5:

license\_plate\_text = str(detection[1]).replace(' ', '').upper()

if pattern.match(license\_plate\_text):

print(license\_plate\_text)

plates.add(license\_plate\_text)

# text = f"{detection[0][1]} {detection[0][2] \* 100:.2f}%"

text = str(license\_plate\_text)

cv2.putText(image, text, (x, y - 20), cv2.FONT\_HERSHEY\_COMPLEX, 0.75, (0, 255, 0), 2)

if not os.path.exists('detections'):

os.makedirs('detections')

cv2.imwrite(os.getcwd() + '/detections/' + str(license\_plate\_text) + '.jpg', image)

### 

**Отзыв о работе студента**

|  |
| --- |
| (ФИО студента) |
|  |

Уровень освоения компетенций

|  |  | |
| --- | --- | --- |
|  | (ФИО студента) | |
|  |  | |
| Компетенции | | Уровень сформированности  компетенций |
| *ОК-6 – способностью работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия* | |  |
| *ОК-7 – способностью к самоорганизации и самообразованию* | |  |
| *ПК-2 - способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования* | |  |

отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от СибГУТИ:

Должность руководителя подпись ФИО руководителя

"\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.