

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

КУРСОВАЯ РАБОТА
по дисциплине: «Методы и средства обработки видеоинформации»
на тему:
«Обработка видеоинформации с использованием telegram bot»

Выполнил
студент группы 3540904/10101

Томилин И. С.

Преподаватель

Молодяков С. А.

«___»_____202__ г.

Санкт-Петербург 2022

Содержание

Цель работы и постановка задачи.....	3
Создание бота.....	4
Пример работы.....	5
Использованные API и библиотеки.....	8
Вывод.....	9
Исходный код.....	10

Цель работы и постановка задачи

Цель курсовой работы, создать интерактивное приложение использующее интерфейс telegram бота для обработки изображений на языке python.

Приложение должно удовлетворять следующим условиям:

- принимать на вход изображение с описанием;
- в зависимости от описания **number** или **background**, выполнять манипуляции на сервере;
- при выборе **number**, бот возвращает изображение найденного номера и его распознанное текстовое представление или сообщает о том, что распознать номер не удалось;
- при выборе **background**, бот возвращает видеофайл с наложенным изображением на фон.

Создание бота

Воспользуемся ботом @BotFather, для создания нового бота необходимо следовать инструкциям указанном на рисунке 1.

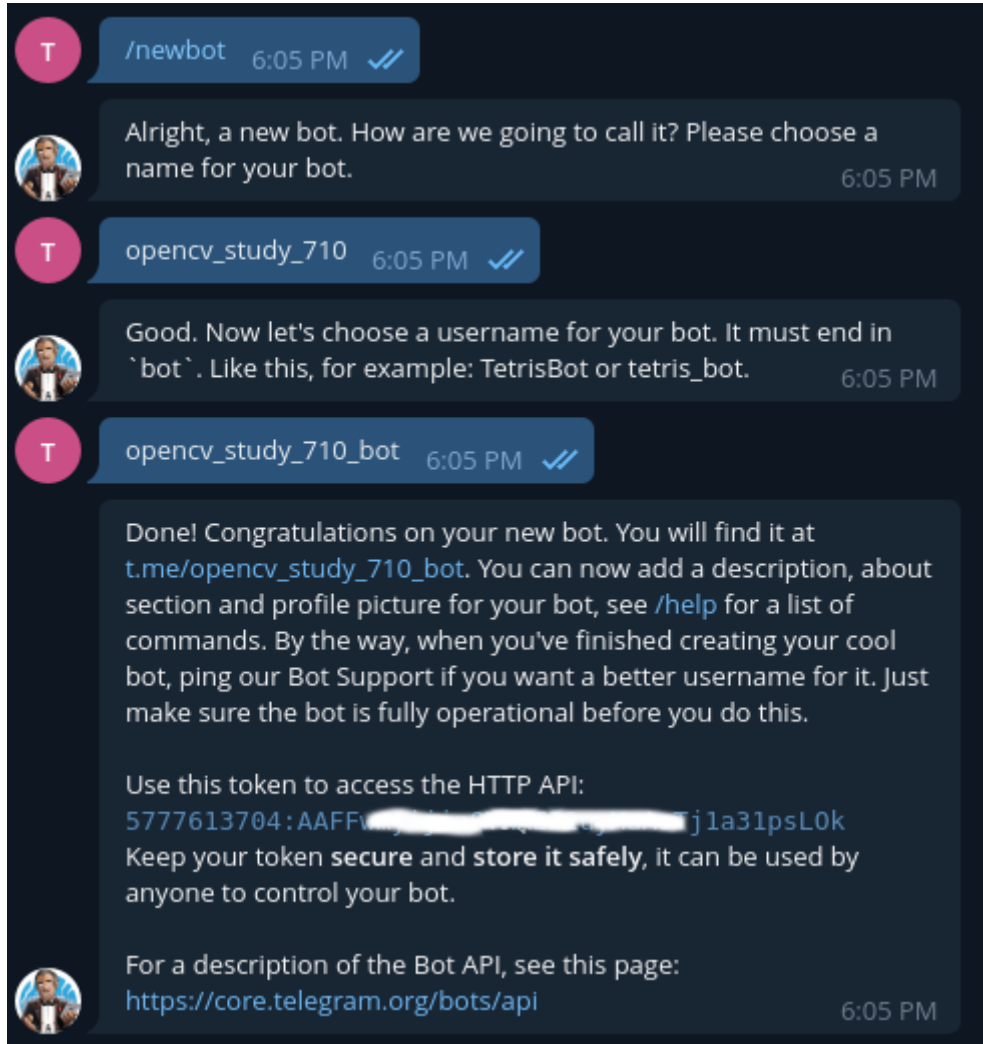


Рисунок 1 – создание нового бота.

После этого бот создан и можно открыть чат.

Для того, чтобы бот начал обрабатывать входящие изображения, нужно запустить на своей машине приложение, предварительно добавив две переменные окружения:

```
export OPENCV_BOT_TOKEN=<token_http_api>
```

```
export OPENCV_BOT_CHAT_ID=<your_telegram_chat_id>
```

Пример работы

Изображения необходимо отправлять с выбранным checkbox “compress images” как на рисунке 2, поскольку при отправке без выбранного checkbox тип файла будет распознан ботом как document и не будет обработан.

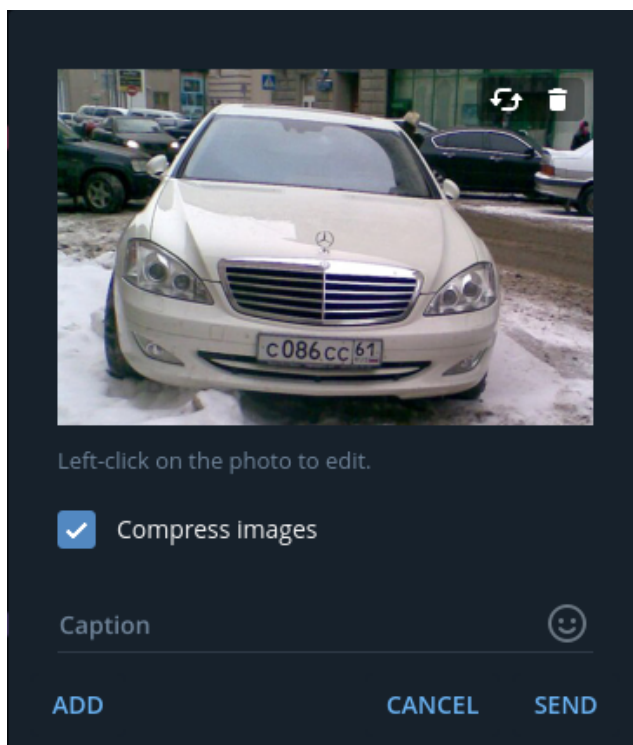


Рисунок 2 – правильный вариант загрузки изображения.

Если не задан caption, то бот сообщит о том, что его нужно ввести, рисунок 3.

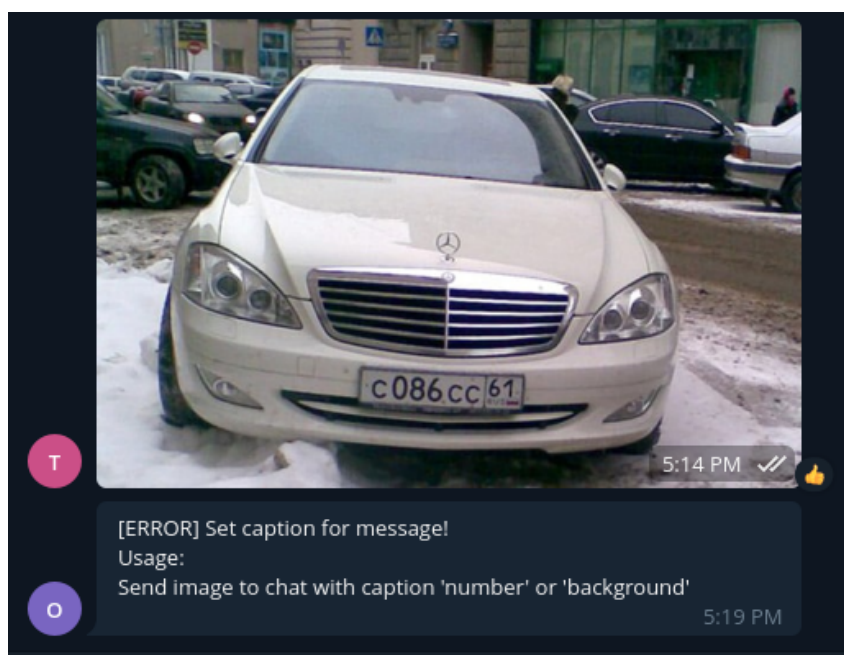


Рисунок 3 – сообщение бота, если не указан тип.

Добавим к caption **“number”** к отправляемому изображению и проверим результат обработки, рисунок 4.

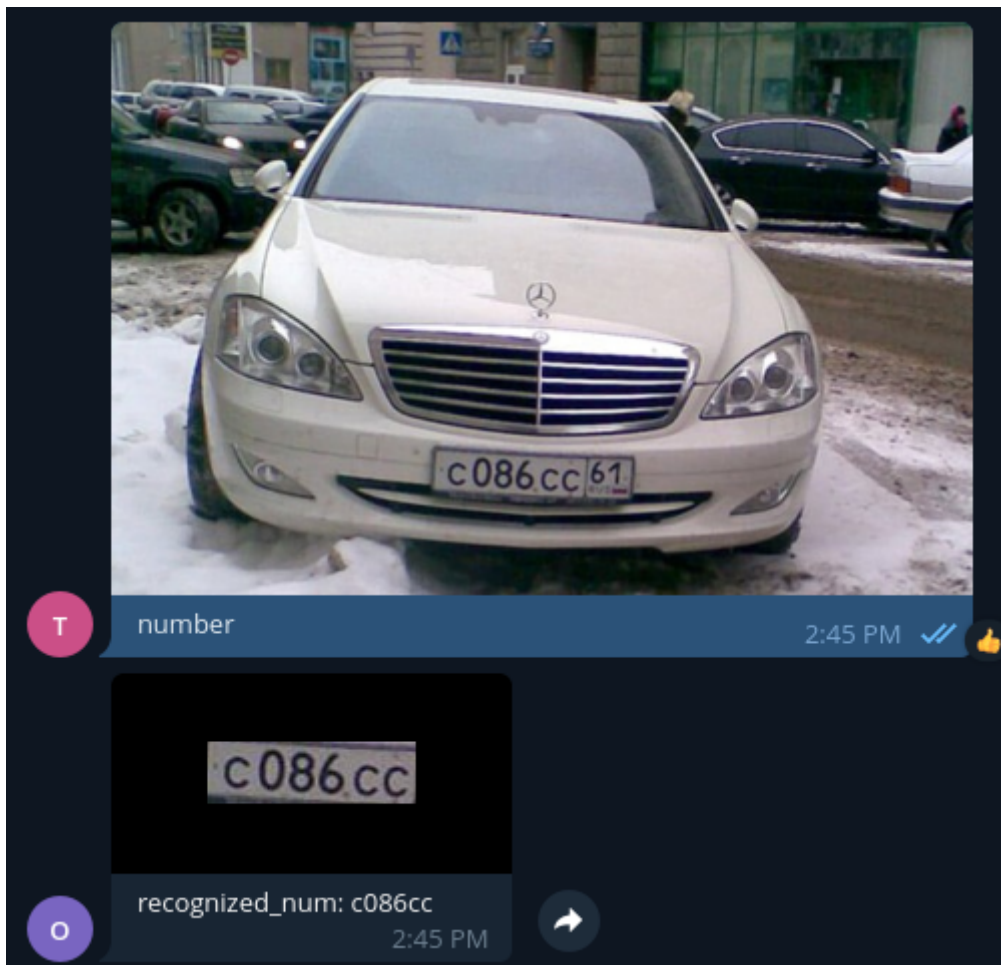


Рисунок 4 – успешный результат распознавания номера.

Пример неудачного распознавания номера, рисунок 5.

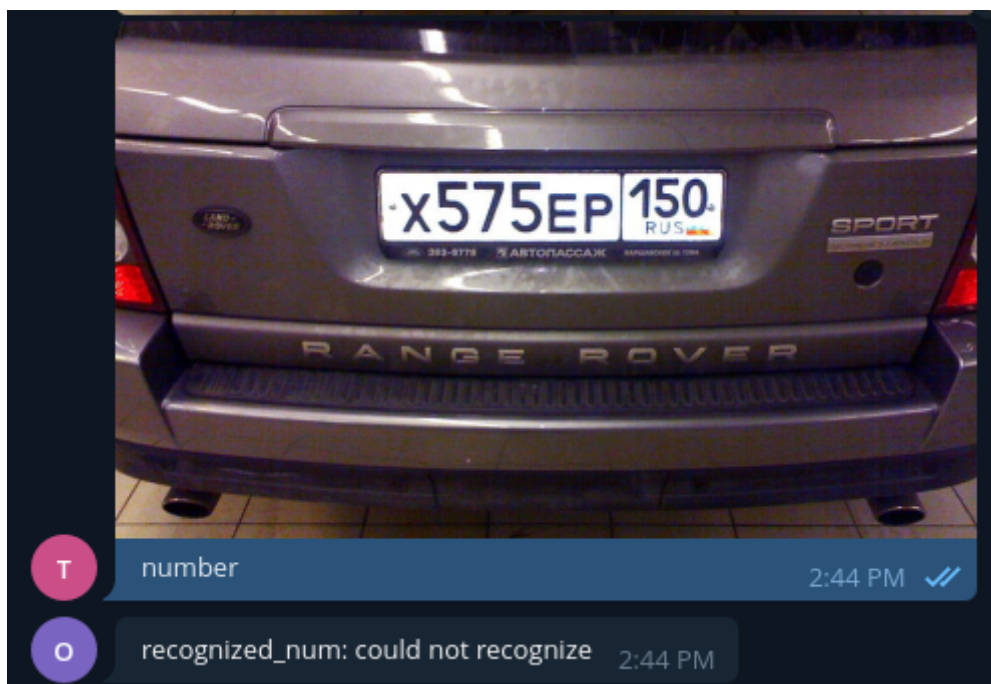


Рисунок 5 – неудачное распознавание номера.

Теперь добавим второй вариант **“background”**, показанный на рисунке 6.

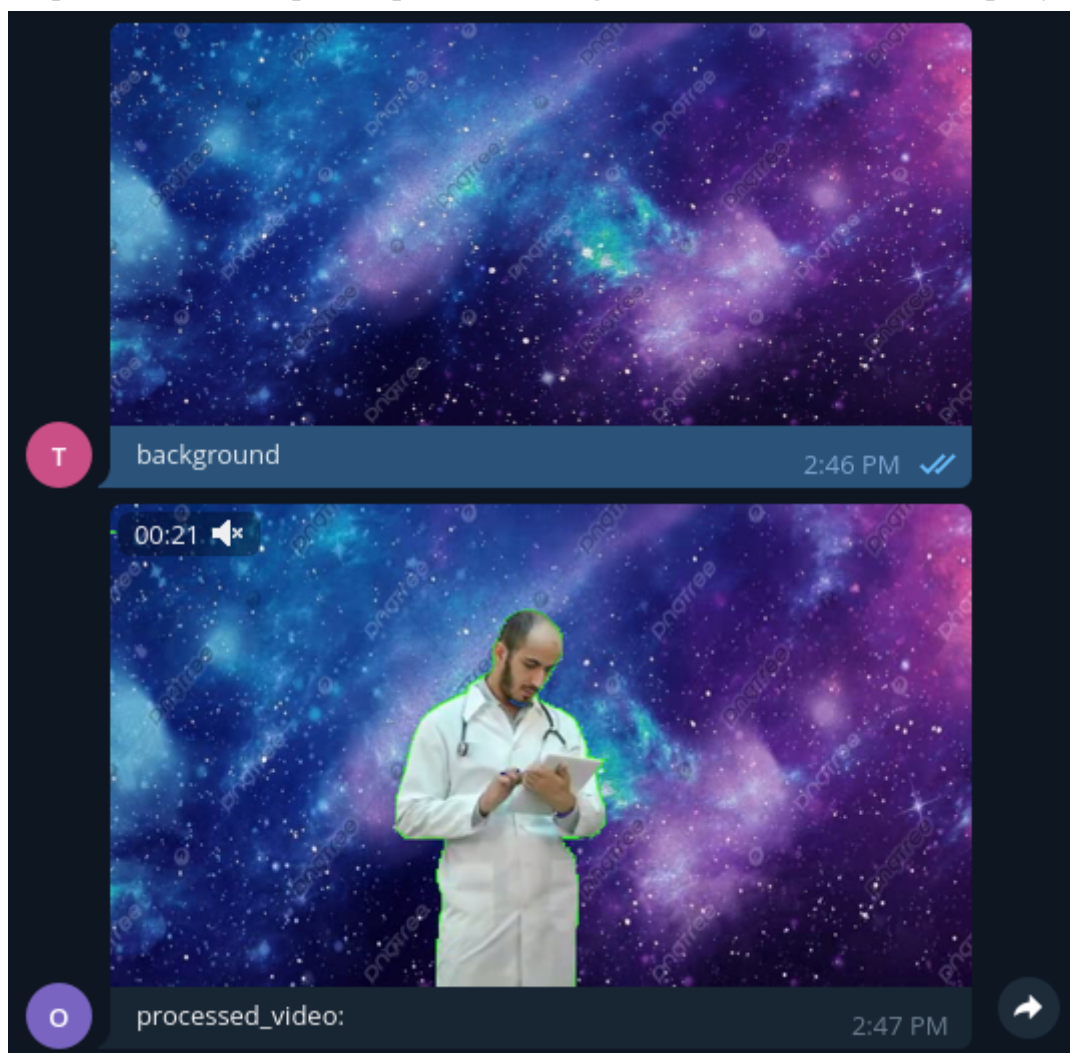


Рисунок 6 – кадр замененного фона у видео.

Как видно на рисунке, отправленное изображение стало фоном видео.

Использованные API и библиотеки

В ходе курсовой работы была использована python библиотека для работы с telegram ботами **telepot** – <https://github.com/nickoala/telepot>.

Вывод

В ходе выполнения курсовой работы было создано приложение, которое полностью соответствует всем поставленным задачам.

Проект доступен на github по ссылке:

https://github.com/itomilin/opencv_play/tree/master/course

Исходный код

Telegram бот:

```
#!/usr/bin/env python3

import sys
import os
import time

import telepot
from telepot.loop import MessageLoop
from pprint import pprint

from number_recognition import number_recognition
from background_remover import change_background

BOT_TOKEN = os.environ['OPENCV_BOT_TOKEN']
CHAT_ID = int( os.environ['OPENCV_BOT_CHAT_ID'] )

def usage():
    return "Usage:\nSend image to chat with caption 'number' or 'background'"

def handle( msg ):
    content_type, chat_type, chat_id = telepot.glance( msg )
    if content_type == 'photo':
        case = None
        try:
            case = msg['caption']
            pprint( msg['caption'] )
        except KeyError:
            err_msg = f"[ERROR] Set caption for message!\n{usage()}"
            print( err_msg )
            bot.sendMessage( CHAT_ID, err_msg )
            return

    img_path = f"./chat_files/{msg['photo'][-1]['file_unique_id']}.png"
    _, file_name = os.path.split( img_path )

    bot.download_file( msg['photo'][-1]['file_id'], img_path )

    if case == "number":
```

```

text_number, status = number_recognition( img_path )
if status:
    bot.sendPhoto( CHAT_ID,
                    photo=open( f"./processed_files/{file_name}", 'rb'),
                    caption=f'recognized_num: {text_number}')
else:
    bot.sendMessage( CHAT_ID, f'recognized_num: {text_number}' )
elif case == "background":
    change_background( img_path )
    bot.sendVideo( CHAT_ID,
                    video=open(f"./processed_files/{file_name.split('.')[0]}.mp4", 'rb'),
                    caption=f'processed_video:')
else:
    print( "Wrong case, try again!" )
    return

```

```

print( "!!processing done!!" )

```

```

elif content_type == 'video':
    pass
elif content_type == 'text':
    pass
else:
    print( "Type is not handled" )

```

```

bot = telepot.Bot( BOT_TOKEN )
MessageLoop( bot, handle ).run_as_thread()

```

```

print( usage() )

```

```

# Keep the program running.
while True:
    time.sleep( 1200 )

```

Распознавание номеров:

```

import os

```

```

import cv2
import pytesseract

```

```

from imutils import contours

```

```

def number_recognition( img ):
    _, file_name = os.path.split( img )
    car_number_img = None

```

```

msg = "could not recognize"
status = 0
image = cv2.imread( img )

height, width = image.shape[:2]

image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # преобразовать
изображение в

#cv2.imshow("Gray Image", image_gray) #вывести на экран для проверки
#cv2.waitKey()

# пороговая обработка для выделения контуров
threshold = cv2.threshold(image_gray, 0, 255, cv2.THRESH_OTSU)[1]
#cv2.imshow("Threshold", threshold) #вывести на экран для проверки
#cv2.waitKey()

# Найдем все контуры
contourss = cv2.findContours(threshold, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
contourss, _ = contours.sort_contours(contourss[0]) # отсортируем их

chars = set('0123456789,') # создадим множество цифр для отсекаания
"мусорных" текстов

for c in contourss: # переберем все контуры
    area = cv2.contourArea(c) # найдем площадь контура
    x, y, w, h = cv2.boundingRect(c)

    if area > 500: # если площадь соизмерима с номером
        img = image[y:y+h, x:x+w] # получим этот контур из исходного
изображения
        # cv2.waitKey()
        result = pytesseract.image_to_string(img, lang='rus+eng')

        if len(result) > 7:
            car_number_img = img.copy()
            chars = "-`\'\"() ,."
            for char in chars:
                result = result.replace( char, "" )
            msg = result
            cv2.imwrite( f"./processed_files/{file_name}", car_number_img )
            status = 1

#cv2.imshow( "car_n", car_number_img )

```

```
#cv2.waitKey()
```

```
return msg, status
```

Замена фона на видео

```
import cv2
import numpy as np
import os

def change_background( img_path ):
    cap = cv2.VideoCapture( '../imgs/green_screen_360.mp4' )
    _, file_name = os.path.split( img_path )
    drop_ext = file_name.split('.')[0]

    bg = cv2.imread( img_path )
    bg = cv2.resize(bg, (640, 360))

    l_green = np.array([0, 0, 45])
    u_green = np.array([177, 216, 255])

    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
    out = cv2.VideoWriter( f'./processed_files/{drop_ext}.mp4',
                          0x7634706d,
                          20.0,
                          (frame_width, frame_height) )

    while True:
        ret, frame = cap.read()
        if not ret:
            break

        if cv2.waitKey(25) & 0xFF == ord('q'):
            break

        #frame = cv2.resize(frame, (640, 360))

        hsv = cv2.cvtColor( frame, cv2.COLOR_BGR2HSV )

        mask = cv2.inRange(hsv, l_green, u_green)
        frame[mask == 0] = bg[mask == 0]

        out.write(frame)
        #cv2.imshow("video2", frame)
```

```
cap.release()  
out.release()
```