Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Архитектура ЭВМ. Часть 2»

Телеграмм бот

Выполнил студент гр. 3530904/10004: Нефедев В. К.

Руководитель:

проф. д.т.н. Молодяков С.А.

**Санкт-Петербург**

**2023**

Оглавление

[Оглавление 2](#_Toc136452690)

[Задача 2](#_Toc136452691)

[Примеры 4](#_Toc136452692)

[Код 10](#_Toc136452693)

# Задача

Создать бота, который имеет команды:

* /start – команда для вывода приветствия с пользователем

Обработка изображений:

* /edge – оставить только границы на изображении
* /blur – размыть изображение
* /sketch – наложить эффект рисунка карандашом
* /blue\_shift – сдвинуть изображение по синей компоненте цвета
* /sepia – наложить эффект сепии
* /yellow – применить жёлтый фильтр
* /thrange – эффект порогового значения
* /discard – удалить фото, находящееся у бота

Обработка видео:

* /сut <число> <число> – обрезать видео с началом, определяемым первым аргументом, и концом, определяемым вторым аргументом
* /blackwhite – сделать видео чёрно-белым
* /vdiscard – удалить видео, находящееся у бота

При отправке боту изображения или видео, он сохраняет их для последующей обработки. Накладываться эффекты будут последовательно. Можно также указать команду для обработки сразу при отправке видео или изображения.

В работе бота были использованы библиотеки wand и moviepy. Wand в своей работе использует ImageMagick, а moviepy – ffmpeg.

Использованные в программе функции

|  |  |
| --- | --- |
| moviepy.video.fx.all.blackwhite(video\_file) | Применяет чёрно-белый фильтр к видео |
| video\_file.subclip(startend[0], startend[1]) | Обрезка видео |
| img.blur(radius=0, sigma=3) | Размытие изображения |
| img.edge(radius=1) | Оставить только границы изображения |
| img.transform\_colorspace("gray") | Перевести изображение в чёрно-белый формат |
| img.blue\_shift(factor=1.25) | Сдвинуть изображение по синей компоненте цвета |
| img.sketch(0.5, 0.0, 98.0) | Наложить эффект рисунка карандашом |
| img.colorize(color="yellow", alpha="rgb(10%, 0%, 20%)") | Наложить фильтр определённого цвета (в данном случае жёлтый) |
| img.sepia\_tone(threshold=0.8) | Наложить эффект сепии |
| img.range\_threshold(  low\_black=black\_point - delta,  low\_white=white\_point - delta, high\_white=white\_point + delta, high\_black=black\_point + delta) | Наложить эффект порогового значения |

# Примеры

Изображение выглядит как текст, электроника, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, вода, озеро, снимок экрана

Автоматически созданное описание

Изображение выглядит как вода, озеро, мультимедиа, снимок экрана

Автоматически созданное описание

Изображение выглядит как снимок экрана, вода, мультимедиа, программное обеспечение

Автоматически созданное описание

Изображение выглядит как дирижабль, транспорт, Дирижабль, самолет

Автоматически созданное описание

Изображение выглядит как текст, вода, на открытом воздухе

Автоматически созданное описание

Изображение выглядит как вода, текст, снимок экрана, мультимедиа

Автоматически созданное описание

# Код

import telebot  
import time  
import ffmpeg  
import os  
import moviepy  
from moviepy.editor import \*  
from wand.image import Image  
  
bot = telebot.TeleBot("")  
image\_filename = ""  
video\_filename = ""  
got\_image = False  
got\_video = False  
handler\_to\_blur = False  
handler\_to\_edge = False  
handler\_to\_bs = False  
handler\_to\_yellow = False  
handler\_to\_sepia = False  
handler\_to\_sketch = False  
handler\_to\_threshold\_range = False  
time\_given = False  
time\_wait = False  
startend = list()  
  
  
@bot.message\_handler(commands=['start'])  
def start(message):  
 global image\_filename, got\_image, handler\_to\_blur, handler\_to\_edge, handler\_to\_bs, \  
 handler\_to\_sketch, handler\_to\_sepia, handler\_to\_yellow, handler\_to\_threshold\_range, got\_video, time\_wait, time\_given  
 bot.send\_message(message.from\_user.id, "Бот для курсовой по Архитектуре компьютера. Обрабатывает видео и изображения\n"  
 "\nblue\_shift - Сместить изображение по синей компоненте"  
 "\nedge - Оставить только границы на изображении"  
 "\nblur - Размыть изображение"  
 "\nsketch - Наложить эффект рисунка карандашом"  
 "\nsepia - Наложить эффект сепии"  
 "\nyellow - Применить жёлтый фильтр"  
 "\nthrange - Эффект порогового значения"  
 "\ncut число число - Обрезать видео"  
 "\nblackwhite - Сделать видео чёрно-белым"  
 "\nvdiscard - Удалить видео, находящееся у бота"  
 "\ndiscard - Удалить изображение, находящееся у бота")  
 for f in os.listdir("files/"+str(message.chat.id)):  
 os.remove(os.path.join("files/"+str(message.chat.id),f))  
 image\_filename = ""  
 got\_image = False  
 got\_video = False  
 handler\_to\_blur = False  
 handler\_to\_edge = False  
 handler\_to\_bs = False  
 handler\_to\_yellow = False  
 handler\_to\_sepia = False  
 handler\_to\_sketch = False  
 handler\_to\_threshold\_range = False  
 time\_given = False  
 time\_wait = False  
  
"""  
@bot.message\_handler(commands=['add\_watermark'])  
def add\_watermark(message):  
 global got\_video, video\_filename  
 if got\_video:  
 ffmpeg.input(video\_filename, i=watermark.png)  
 ff.options("-i input.mp4 -i watermark.png -filter\_complex overlay=1500:10 output.mp4")  
 else:  
 bot.send\_message(message.from\_user.id, "У бота не было файла")  
"""  
  
@bot.message\_handler(commands=['blackwhite'])  
def bw\_video(message):  
 global video\_filename  
 if got\_video:  
 video\_file = VideoFileClip(video\_filename)  
 video\_file = moviepy.video.fx.all.blackwhite(video\_file)  
 video\_file.write\_videofile(video\_filename.split(".")[0] + "\_new.mp4")  
 os.remove(video\_filename)  
 video\_filename = video\_filename.split(".")[0] + "\_new.mp4"  
 video = open(video\_filename, 'rb')  
 bot.send\_video(message.from\_user.id, video, caption="Видео обработано!")  
 else:  
 bot.send\_message(message.from\_user.id,  
 "У бота нет видео для обработки!")  
  
@bot.message\_handler(commands=['cut'])  
def cut\_video(message):  
 global time\_given, startend, got\_video, time\_wait, video\_filename  
 if got\_video:  
 split\_message = message.text.split(" ")  
 if len(split\_message) == 3 and split\_message[1].isnumeric() and split\_message[2].isnumeric():  
 startend = [int(split\_message[1]), int(split\_message[2])]  
 time\_given = True  
 if time\_given:  
 video\_file = VideoFileClip(video\_filename)  
 if (startend[1] > video\_file.duration or startend[0] > video\_file.duration):  
 bot.send\_message(message.from\_user.id,  
 "Начало или конец видео больше, чем его длина")  
 time\_given = False  
 else:  
 video\_file.subclip(startend[0], startend[1])  
 video\_file.write\_videofile(video\_filename.split(".")[0] + "\_new.mp4")  
 os.remove(video\_filename)  
 video\_filename = video\_filename.split(".")[0] + "\_new.mp4"  
 video = open(video\_filename, 'rb')  
 bot.send\_video(message.from\_user.id, video, caption="Видео обрезано!")  
 time\_given = False  
 else:  
 time\_wait = True  
 bot.send\_message(message.from\_user.id, "Отправьте время начала видео и время конца видео в секундах:\nПример: 5 10")  
 else:  
 bot.send\_message(message.from\_user.id,  
 "У бота нет видео для обработки!")  
  
  
@bot.message\_handler(content\_types=['video'])  
def handler\_video(message):  
 global got\_video, video\_filename  
 if not got\_video:  
 file\_info = bot.get\_file(message.video.file\_id)  
 downloaded\_file = bot.download\_file(file\_info.file\_path)  
 src = f'files/{message.chat.id}/' + file\_info.file\_path.replace('videos/', '')  
 with open(src, 'wb') as new\_file:  
 new\_file.write(downloaded\_file)  
 video\_filename = new\_file.name  
 bot.send\_message(message.from\_user.id, "Видео cохранено!\n"  
 "\ncut число число - Обрезать видео"  
 "\nblackwhite - Сделать видео чёрно-белым"  
 "\nvdiscard - Удалить видео, находящееся у бота")  
 got\_video = True  
  
@bot.message\_handler(commands=['discard'])  
def discard\_image(message):  
 global got\_image, image\_filename  
 if got\_image:  
 os.remove(image\_filename)  
 bot.send\_message(message.from\_user.id, "Файл удалён")  
 image\_filename = ""  
 got\_image = False  
 else:  
 bot.send\_message(message.from\_user.id, "У бота не было файла")  
  
@bot.message\_handler(commands=['vdiscard'])  
def discard\_video(message):  
 global got\_video, video\_filename  
 if got\_video:  
 os.remove(video\_filename)  
 bot.send\_message(message.from\_user.id, "Файл удалён")  
 video\_filename = ""  
 got\_video = False  
 else:  
 bot.send\_message(message.from\_user.id, "У бота не было файла")  
  
@bot.message\_handler(commands=['blur'])  
def blur\_answer(message):  
 global handler\_to\_blur  
 handler\_to\_blur = True  
 if got\_image:  
 handler\_to\_blur = False  
 blur(message)  
 elif message.content\_type == 'photo':  
 handler\_image(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Отправьте файл для размытия")  
  
@bot.message\_handler(commands=['edge'])  
def edge\_answer(message):  
 global handler\_to\_edge  
 handler\_to\_edge = True  
 if got\_image:  
 handler\_to\_edge = False  
 draw\_edges(message)  
 elif message.content\_type == 'photo':  
 handler\_image(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Отправьте файл для обработки")  
  
@bot.message\_handler(commands=['blue\_shift'])  
def blue\_shift\_answer(message):  
 global handler\_to\_bs  
 handler\_to\_bs = True  
 if got\_image:  
 handler\_to\_bs = False  
 blue\_shift(message)  
 elif message.content\_type == 'photo':  
 handler\_image(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Отправьте файл для обработки")  
  
@bot.message\_handler(commands=['sketch'])  
def sketch\_answer(message):  
 global handler\_to\_sketch  
 handler\_to\_sketch = True  
 if got\_image:  
 handler\_to\_sketch = False  
 sketch(message)  
 elif message.content\_type == 'photo':  
 handler\_image(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Отправьте файл для обработки")  
  
@bot.message\_handler(commands=['colorize'])  
def colorize\_answer(message):  
 global handler\_to\_yellow  
 handler\_to\_yellow = True  
 if got\_image:  
 handler\_to\_yellow = False  
 yellow(message, message.split(" ")[1])  
 elif message.content\_type == 'photo':  
 handler\_image(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Отправьте файл для обработки")  
  
@bot.message\_handler(commands=['sepia'])  
def sepia\_answer(message):  
 global handler\_to\_sepia  
 handler\_to\_sepia = True  
 if got\_image:  
 handler\_to\_sepia = False  
 sepia(message)  
 elif message.content\_type == 'photo':  
 handler\_image(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Отправьте файл для обработки")  
  
@bot.message\_handler(commands=['thrange'])  
def thrange\_answer(message):  
 global handler\_to\_threshold\_range  
 handler\_to\_threshold\_range = True  
 if got\_image:  
 handler\_to\_threshold\_range = False  
 thrange(message)  
 elif message.content\_type == 'photo':  
 handler\_image(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Отправьте файл для обработки")  
  
def blur(message):  
 with Image(filename=image\_filename) as img:  
 img.blur(radius=0, sigma=3)  
 img.save(filename=image\_filename)  
 photo = open(image\_filename, 'rb')  
 bot.send\_photo(message.from\_user.id, photo, "Фото размыто!")  
  
def draw\_edges(message):  
 with Image(filename=image\_filename) as img:  
 img.transform\_colorspace("gray")  
 img.edge(radius=1)  
 img.save(filename=image\_filename)  
 photo = open(image\_filename, 'rb')  
 bot.send\_photo(message.from\_user.id, photo, "Фото обработано!")  
  
def blue\_shift(message):  
 with Image(filename=image\_filename) as img:  
 img.blue\_shift(factor=1.25)  
 img.save(filename=image\_filename)  
 photo = open(image\_filename, 'rb')  
 bot.send\_photo(message.from\_user.id, photo, "Фото смещено!")  
  
def sketch(message):  
 with Image(filename=image\_filename) as img:  
 img.transform\_colorspace("gray")  
 img.sketch(0.5, 0.0, 98.0)  
 img.save(filename=image\_filename)  
 photo = open(image\_filename, 'rb')  
 bot.send\_photo(message.from\_user.id, photo, "Фото обработано!")  
  
def yellow(message):  
 with Image(filename=image\_filename) as img:  
 img.colorize(color="yellow", alpha="rgb(10%, 0%, 20%)")  
 img.save(filename=image\_filename)  
 photo = open(image\_filename, 'rb')  
 bot.send\_photo(message.from\_user.id, photo, "Фото обработано!")  
  
def sepia(message):  
 with Image(filename=image\_filename) as img:  
 img.sepia\_tone(threshold=0.8)  
 img.save(filename=image\_filename)  
 photo = open(image\_filename, 'rb')  
 bot.send\_photo(message.from\_user.id, photo, "Фото смещено!")  
  
def thrange(message):  
 with Image(filename=image\_filename) as img:  
 img.transform\_colorspace('gray')  
 white\_point = 0.9 \* img.quantum\_range  
 black\_point = 0.5 \* img.quantum\_range  
 delta = 0.05 \* img.quantum\_range  
 img.range\_threshold(low\_black=black\_point - delta,  
 low\_white=white\_point - delta,  
 high\_white=white\_point + delta,  
 high\_black=black\_point + delta)  
 img.save(filename=image\_filename)  
 photo = open(image\_filename, 'rb')  
 bot.send\_photo(message.from\_user.id, photo, "Фото обработано!")  
  
  
@bot.message\_handler(content\_types=['photo', 'document'])  
def handler\_image(message):  
 global image\_filename, got\_image, handler\_to\_blur, handler\_to\_edge, handler\_to\_bs, handler\_to\_yellow,\  
 handler\_to\_threshold\_range, handler\_to\_sepia, handler\_to\_sketch  
 if not got\_image:  
 from pathlib import Path  
 Path(f'files/{message.chat.id}/').mkdir(parents=True, exist\_ok=True)  
 if message.content\_type == 'photo':  
 file\_info = bot.get\_file(message.photo[-1].file\_id)  
 downloaded\_file = bot.download\_file(file\_info.file\_path)  
 src = f'files/{message.chat.id}/' + file\_info.file\_path.replace('photos/', '')  
 with open(src, 'wb') as new\_file:  
 new\_file.write(downloaded\_file)  
  
  
 elif message.content\_type == 'document':  
 file\_info = bot.get\_file(message.document.file\_id)  
 downloaded\_file = bot.download\_file(file\_info.file\_path)  
 src = f'files/{message.chat.id}/' + message.document.file\_name  
 with open(src, 'wb') as new\_file:  
 new\_file.write(downloaded\_file)  
  
 image\_filename = new\_file.name  
 got\_image = True  
 if handler\_to\_blur or message.caption == "/blur":  
 handler\_to\_blur = False  
 blur(message)  
 elif handler\_to\_edge or message.caption == "/edge":  
 handler\_to\_edge = False  
 draw\_edges(message)  
 elif handler\_to\_bs or message.caption == "/blue\_shift":  
 handler\_to\_bs = False  
 blue\_shift(message)  
 elif handler\_to\_sketch or message.caption == "/sketch":  
 handler\_to\_sketch = False  
 sketch(message)  
 elif handler\_to\_yellow or message.caption == "/yellow":  
 handler\_to\_yellow = False  
 yellow(message)  
 elif handler\_to\_threshold\_range or message.caption == "/thrange":  
 handler\_to\_threshold\_range = False  
 thrange(message)  
 elif handler\_to\_sepia or message.caption == "/sepia":  
 handler\_to\_sepia = False  
 sepia(message)  
 else:  
 bot.send\_message(message.from\_user.id, "Бот сохранил изображение.\nblue\_shift - Сместить изображение по синей компоненте"  
 "\nedge - Оставить только границы на изображении"  
 "\nblur - Размыть изображение"  
 "\nsketch - Наложить эффект рисунка карандашом"  
 "\nsepia - Наложить эффект сепии"  
 "\nyellow - Применить жёлтый фильтр"  
 "\nthrange - Эффект порогового значения"  
 "\ndiscard - Удалить изображение, находящееся у бота")  
  
 else:  
 bot.send\_message(message.from\_user.id, "У бота уже есть файл. (Удалить файл: /discard)")  
  
  
@bot.message\_handler(content\_types=['text'])  
def handler\_text(message):  
 global startend, time\_given, time\_wait  
 split\_message = message.text.split(" ")  
 if time\_wait and len(split\_message) == 3 and split\_message[1].isnumeric() and split\_message[2].isnumeric():  
 startend = [int(split\_message[1]), int(split\_message[2])]  
 time\_given = True  
 time\_wait = False  
 else:  
 bot.send\_message(message.from\_user.id, "Неопознанная команда!")  
  
while True:  
 try:  
 bot.polling(non\_stop=True, interval=0)  
 except Exception as e:  
 print(e)  
 time.sleep(5)  
 continue