Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий

02.03.02

Алгоритмы цифровой обработки мультимедиа

Лабораторная работа № 7

Системы оптического распознавания текста

Работа будет осуществляется средствами языка Python 3.1 и IDE PyCharm2022.1.2 с учебной лицензией. Для работы необходимо установить библиотеки PyTesseract и Easy OCR.

В рамках данной лабораторной работы будут рассматриваться решения для Tesseract на примере распознавания капчи.

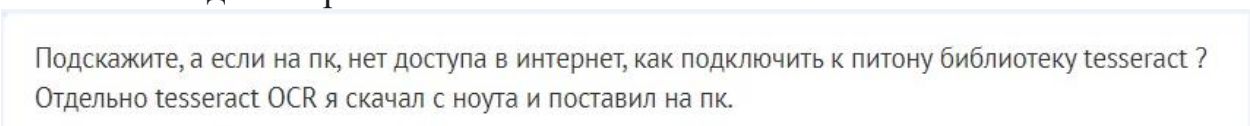
Устанавливаем TesseractOCR, импортируем библиотеку pytesseract. Для работы с этой библиотекой необходимо указать путь к исполняемому файлу непосредственно tesseract`a.

```
import pytesseract
import cv2


pytesseract.pytesseract.tesseract_cmd = (r"C:\Program Files\Tesseract-OCR\tesseract.exe")

img = cv2.imread('test1.jpg')
text = pytesseract.image_to_string(img, lang='rus+eng')
print(text)
```

Результат работы

```
"C:\Users\Andrey Kram\Desktop\kubik\ACOM\1_tes\tes\Scripts\python.exe" "C:\Users\Andrey Kram\Desktop\kubik\ACOM\Tessaract_1\Tessaract_test.py"
Подскажите, а если на пк, нет доступа в интернет, как подключить к питону библиотеку tesseract ?
'Отдельно tesseract OCR я скачал с ноута и поставил на пк.
```

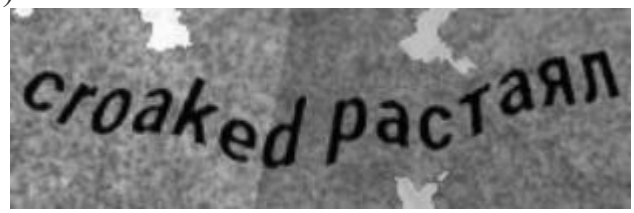
И исходная картинка



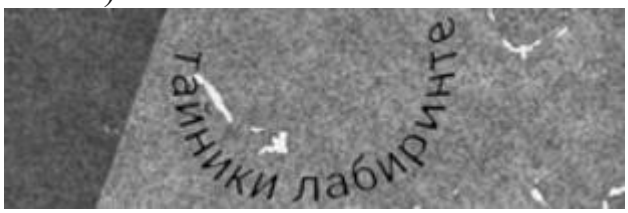
Как мы можем заметить TesseractOCR, несмотря на моральное устаревание, неплохо справляется с обработкой изображения без шумов и прочих дефектов.

Рассмотрим возможности TesseractOCR для распознавания капч. В основном капча, это деформация текста на любой вкус и цвет. Капчи черно-бело-серые, с добавлением фоновых сегментов схожих цветов. Однако, если проанализировать то, что видно на изображении, то можно

прийти к выводу, что в подавляющем большинстве текст на капчах выглядит либо так («змейка»):



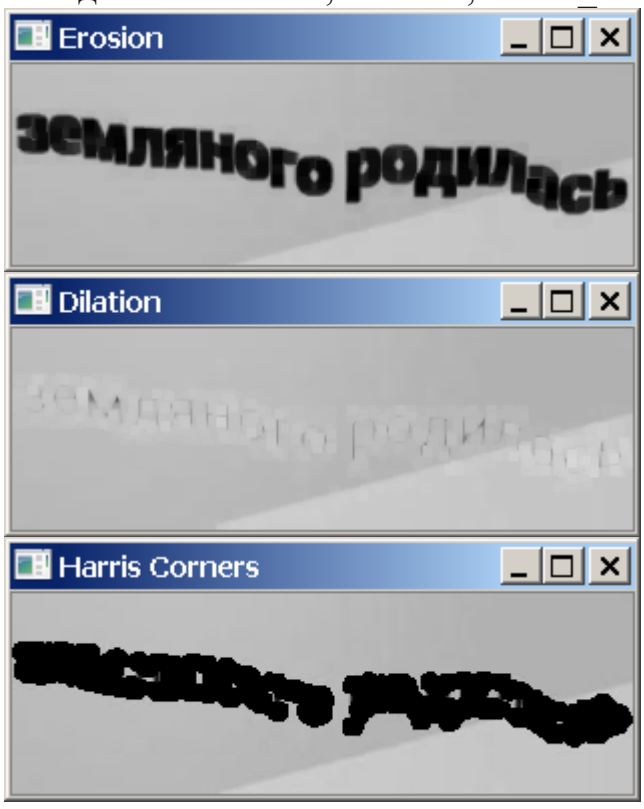
Либо так («улыбка»)



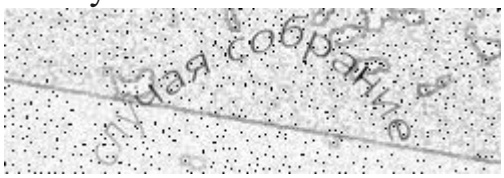
Либо так («горка»)

Также известно, что на изображениях может присутствовать как русский, так и английский текст, представленный большей частью двумя словами. Данные слова не связаны в какую-либо вменяемую фразу, случайны.

Первичный анализ с помощью пакета opencv показал, что капча устойчива к таким методам как Erosion, Dilation, Harris_corners:

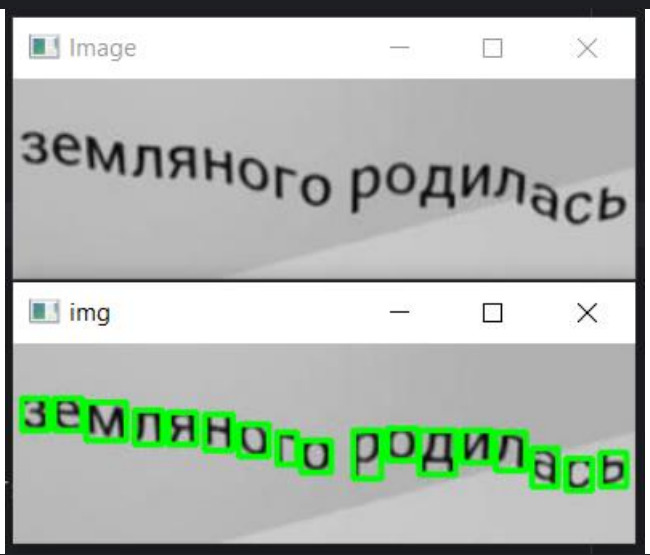


Также ничего не дает попытка «вырезать» пиксели с нужным цветом, так как в капчу добавлены шумы:



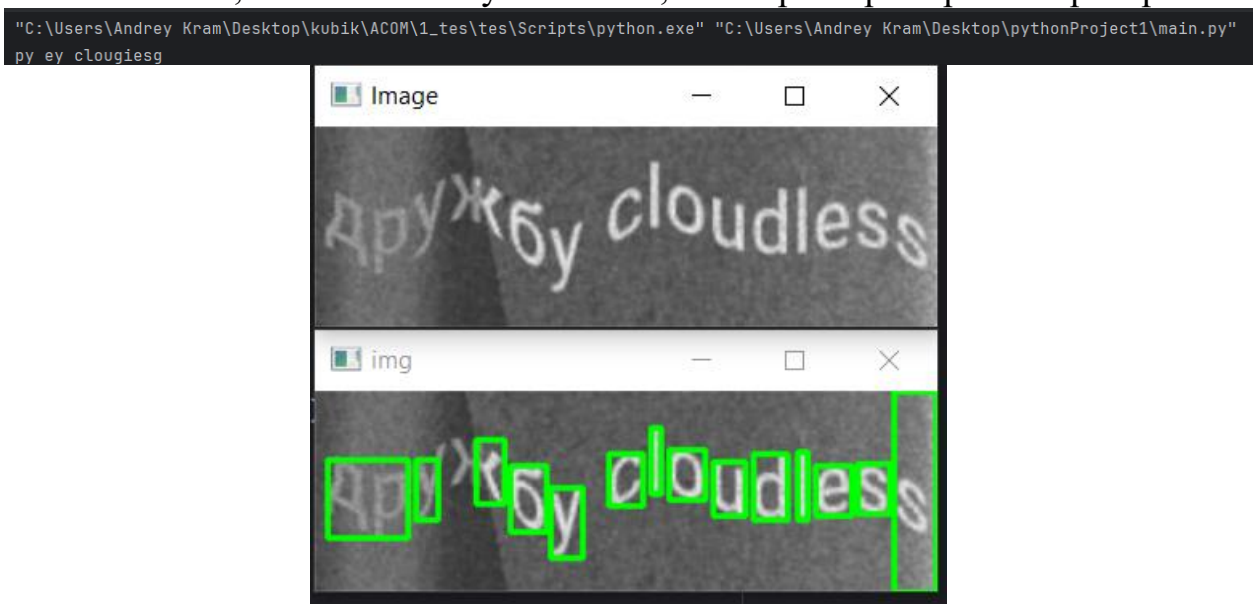
Выделение каждой буквы так же не дает желаемого результата, так как буквы на капче могут быть с некоторым искажением. Листинг и результат ниже.

```python
import cv2
import pytesseract
from pytesseract import Output
import enchant

dictionary = enchant.dict("ru_RU")
pytesseract.pytesseract.tesseract_cmd = (r"C:\Program Files\Tesseract-OCR\tesseract.exe")

img = cv2.imread(r'C:\Users\Andrey Kram\Desktop\yandex-capchi\1-.jpg')
cv2.imshow( winname: 'Image', img)

def convert_grayscale(img):
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    return img
def blur(img, param):
    img = cv2.medianBlur(img,param)
    return img
def threshold(img):
    img = cv2.threshold(img,  thresh: 0,  maxval: 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)[1]
    return img
h, w, c = img.shape
boxes = pytesseract.image_to_boxes(img)
for b in boxes.splitlines():
    b = b.split(' ')
    img = cv2.rectangle(img, (int(b[1]), h - int(b[2])), (int(b[3]), h - int(b[4])), (0,255,0), 2)

text = pytesseract.image_to_string(img, lang='rus+eng')
```




```
"C:\Users\Andrey Kram\Desktop\kubik\ACOM\1_tes\tes\Scripts\python.exe" "C:\Users\Andrey Kram\Desktop\pythonProject1\main.py"
ЗеМпЯНого Родилась
```

Кажется, что все не так уж и плохо, но второй пример это опровергает.



Задание 1.

Разметить любым способом изображения в архиве (названия изображений или отдельный файл – txt, json).

Написать метод test_recognition(rec_type, val_type), который в зависимости от rec_type будет по разному вызывать методы распознавания текста в OCR тессеракт и easyOCR. В простейшем случае – метод straight_recognition, который просто вызывает метод image_to_string у тессеракта. Продумать способы оценки качества распознавания val_type (как минимум два способа: в простейшем случае – полное совпадение, но попробуйте и другие способы придумать). Метод должен записать в отдельный файл список пар строк (распознано-должно быть) и вернуть статистические результаты валидации. В дальнейшем в заданиях Вам нужно будет писать обработку rec_type новый способ распознавания – новая функция.

Задание 2.

Запустить тестирование на исходном датасете и классическом предобученном тессеракте. Записать в сводную таблицу результаты оценки качества распознавания.

Провести аугментацию датасета: Повернуть каждое изображение на угол от -20 до 20 с шагом в 1° градус и снова обработать, то есть вместо каждого из изображений должно появится 41 изображение, соответсвующее повороту на целое число градусов. Провести тестирование на изменившимся датасете, записать в сводную таблицу результаты оценки качества распознавания. Новый датасет будем называть датасет2.

Задание 3.

Написать новый метод для rect_type, который будет работать следующим образом. Для каждого изображения формировать 41 изображение,

как в предыдущем задании, и применять тессеракт на каждое изображение следующим образом: получить 41 ответ, сформировать итоговый ответ с использованием этих 41 результата.

Протестировать новый метод на исходном датасете, записать результаты в сводную таблицу.

Задание 4.

Написать новый метод для rect_type, который будет работать следующим образом: вызываем распознавание для изображения стандартным методом to_string, а дальше выполняем постобработку ответа – удаляем спецсимволы, приводим буквы к одному регистру.

Протестировать новый метод на исходном датасете, записать результаты в сводную таблицу. Протестировать новый метод на датасете2, записать результаты в сводную таблицу.

Задание 5.

Дообучить тессеракт на датасете 2. Выполнить запуск тестов на переобученной модели на двух датасетах для 3 написанных rectype.

Задание 6.

Написать новый метод для rect_type, который будет работать отдельно с каждым символом. Определить буквы на капче с помощью встроенного в tesseract метода image_to_boxes, поделить на несвязанные компоненты с помощью connected components (можно ознакомиться с информацией о методе по ссылке https://pyimagesearch.com/2021/02/22/opencv-connected-component-labeling-and-analysis/), встроенного в OpenCV, для каждого компонента найти геометрический центр, связать каждый центр отрезками с двумя ближайшими центрами, после чего удалить две самые длинные связи (получить цепь упорядоченных символов). Распознать с помощью TesseractOCR каждый символ ОТДЕЛЬНО. Причём организовать распознавание символа следующим образом – изображение с символом вращаем на градус до тех пор, пока тессеракт не вернет какой-то символ в ответ. Распознанные символы в форме строки вернуть методом.

Протестировать полученный метод на двух датасетах, записать результаты в таблицу.

Задание 7.

Найти готовый датасет с русским символами, где все картинки – один повернутый символ. Если нужно, провести аугментацию поворотом самостоятельно. Дообучить тессеракт на этом датасете.

Выполнить задание 6 на переобученном тессеракте на двух датасетах, записать результаты в таблицу.

Задание 8.

Запустить тестирование на исходном датасете и датасете 2 и классическом предобученном easyOCR. Записать в сводную таблицу результаты оценки качества распознавания.

Задание 9.

Запустить переобучение easyOCR на датасете2.

Запустить тестирование на исходном датасете и датасете 2 и переобученном easyOCR. Записать в сводную таблицу результаты оценки качества распознавания.

Контрольные вопросы:

1. Опишите API тессеракта и стуктуру возвращаемых значение в разных методах.
2. Опишите структуру интеллектуальной модели распознавания текста, применяемой в тессеракте.
3. Опишите, как технически производится обучение tesseract
4. Опишите API easyOCR и стуктуру возвращаемых значение в разных методах.
5. Опишите структуру интеллектуальной модели распознавания текста, применяемой в easyOCR.
6. Опишите, как технически производится обучение easyOCR

Формат оценивания выполнения заданий на лабораторной работе:

- оценка «+» ставится на лабораторной работе, если студент выполняет задания 1-2 на занятии;
- оценка «удовлетворительно» ставится на лабораторной работе, если студент выполняет четыре любых задания;
- оценка «хорошо» ставится на лабораторной работе, если студент выполняет шесть любых заданий;
- оценка «отлично» ставится на лабораторной работе, если студент выполняет шесть любых заданий и отвечает правильно на все теоретические вопросы;
- оценка «отлично» ставится на лабораторной работе, если студент выполняет восемь любых заданий.

Если студент сдаёт работу позже, то применяется формат оценивания:

- оценка «+» ставится на защите, если студент выполняет 7 заданий без обучения моделей, предоставляет гит, таблицу, отвечает на вопросы по таблице, ответы на вопросы по теории и по коду необязательны;

- оценка «+» ставится на защите, если студент выполняет 5 заданий без обучения моделей, предоставляет гит, таблицу, отвечает на любые вопросы преподавателя, код и теория, кроме обучения моделей, ВОЗМОЖНЫ ОШИБКИ ПРИ ОТВЕТАХ;

- оценка «+» ставится на защите, если студент выполняет 5 заданий без обучения моделей, предоставляет гит, таблицу, отвечает на вопросы преподавателя по коду БЕЗ ОШИБОК, ответы на вопросы по теории необязательны;

- оценка «+» ставится на защите, если студент выполняет 3 любых задания, обязательно включая обучение одной из существующих моделей, предоставляет гит, таблицу, отвечает на любые вопросы преподавателя, код и теория, ВОЗМОЖНЫ ОШИБКИ ПРИ ОТВЕТАХ;

- оценка «3» ставится на защите, если студент выполняет 7 заданий без обучения моделей, предоставляет гит, таблицу, отвечает на вопросы преподавателя по коду БЕЗ ОШИБОК, ответы на вопросы по теории необязательны;

- оценка «3» ставится на защите, если студент выполняет 5 заданий без обучения моделей, предоставляет гит, таблицу, отвечает на любые вопросы преподавателя, код и теория, кроме обучения моделей, БЕЗ ОШИБОК;

- оценка «3» ставится на защите, если студент выполняет 5 любых заданий, обязательно включая обучение одной из существующих моделей, предоставляет гит, таблицу, отвечает на любые вопросы преподавателя, код и теория, ВОЗМОЖНЫ ОШИБКИ ПРИ ОТВЕТАХ;

- оценка «4» ставится на защите, если студент выполняет 7 заданий без обучения моделей, предоставляет гит, таблицу, отвечает на любые вопросы преподавателя, кроме обучения моделей, БЕЗ ОШИБОК;

- оценка «4» ставится на защите, если студент выполняет 7 любых заданий, обязательно включая обучение одной из существующих моделей, предоставляет гит, таблицу, отвечает на любые вопросы преподавателя ВОЗМОЖНЫ ОШИБКИ ПРИ ОТВЕТАХ;

- оценка «4» ставится на защите, если студент выполняет 5 любых заданий, обязательно включая обучение одной из существующих моделей, предоставляет гит, таблицу, отвечает на любые вопросы преподавателя БЕЗ ОШИБОК;

- оценка «5» ставится на защите, если студент выполняет 7 любых заданий, обязательно включая обучение одной из существующих моделей, предоставляет гит, таблицу отвечает на вопросы по коду, может выполнить любой фрагмент работы, который попросит преподаватель и отвечает на теоретические вопросы включая обучение одной из моделей.