

ПРОГРАММА РАСПОЗНАВАНИЯ ЛИЦ НА PYTHON

Чуллийев Шохрух Ибодуллаевич

ассистент, ТУИТ им. Мухаммада аль-Хорезми
chuliyevshohruh@gmail.com

Сафарова Мафтунахон Журабековна

ассистент, ТУИТ им. Мухаммада аль-Хорезми
maftunasafarova94@mail.ru

Абдуллаева Замира Шамшаддиновна

доцент, ТУИТ им. Мухаммада аль-Хорезми
sabina-07-14@mail.ru

<https://doi.org/10.5281/zenodo.7483015>

Аннотация. Язык программирования Python за последние годы настолько изменился, что охватил многие области. Например, Python считается одним из самых надежных языков программирования в области науки о данных, глубокого обучения, машинного обучения, компьютерного зрения и искусственного интеллекта. При работе с данными в искусственном интеллекте, нейронных сетях и в вышеуказанных областях визуализация решений проблем помогает визуализировать точность решения. Библиотека Matplotlib помогает нам создать визуальный образ. В этой диссертации мы сосредоточимся на том, как обрабатывать изображение, насколько оно массивное и как изменить контрастность при обработке изображений.

Ключевые слова: python, matplotlib, opencv python, jupyter, numpy, pandas, шаг, массив, псевдо-цветовые схемы.

FACE RECOGNITION PROGRAM IN PYTHON

Abstract. The Python programming language has changed so much in recent years that it has covered many areas. For example, Python is considered one of the most trusted programming languages in data science, deep learning, machine learning, computer vision, and artificial intelligence. When working with data in artificial intelligence, neural networks and the above fields, visualization of problem solutions helps to visualize the accuracy of the solution. The Matplotlib library helps us create a visual image. In this dissertation, we will focus on how to process an image, how massive it is, and how to change the contrast in image processing.

Keywords: python, matplotlib, opencv python, jupyter, numpy, pandas, stride, array, pseudo color schemes.

Введение

Python может обнаружить и распознать ваше лицо на фото или видео

Приложения для распознавания лиц включают в себя разблокировку лица (идентификацию лица), безопасность и защиту и т. д. Врачи и медицинские работники используют распознавание лиц для доступа к медицинским записям и истории болезни пациентов, а также для лучшей диагностики заболеваний.

Основная часть

В этом проекте Python мы хотим создать модель машинного обучения, которая распознает людей на изображениях. В нашем проекте мы используем API распознавания лиц и OpenCV.

Библиотеки

- Python – 3.x
- cv2 – 4.5.2
- numpy – 1.20.3
- face_recognition – 1.3.0

Мы используем следующую команду для установки вышеуказанных пакетов.

1. pip установить numpy opencv-python
2. pip установить dlib

3. pip установить face_recognition

Сбор данных проекта

Мы можем реализовать этот проект распознавания лиц, используя наш собственный набор данных. Для этого проекта я использовал изображение своих друзей в качестве набора данных.

Шаги по разработке модели распознавания лиц

Прежде чем продолжить, давайте разберемся, что такое обнаружение и обнаружение лиц.

Распознавание лиц — это процесс идентификации или проверки лица человека по фотографиям и видеозаписям.

Обнаружение лиц определяется как процесс идентификации и извлечения лиц (местоположение и размер) из изображения для использования алгоритма обнаружения лиц.

Метод распознавания лиц используется для идентификации уникальных отображаемых функций на изображении. В большинстве случаев изображение лица уже удалено, вырезано, масштабировано и стало серым. Распознавание лиц включает в себя 3 этапа: распознавание лиц, выпуск функции, распознавание лиц.

OpenCV — это библиотека с открытым исходным кодом, написанная на C++. Он включает в себя реализацию различных алгоритмов и глубоких нейронных сетей, используемых для задач компьютерного зрения.

1. Готовим набор данных

Создадим 2 каталога, попробуем обучение. Нам нужно убедиться, что изображения, которые мы выбираем, достаточно хорошо представляют черты лица для классификатора.

Давайте протестируем модель, сделаем снимок под названием «Друзья» и поместим его в нашу директорию «common_picture».

2. Обучение т.е. мы научим модель

■ Сначала импортируем необходимые модули.

■ Библиотека face_recognition содержит различные утилиты, помогающие в процессе распознавания лиц.

■ Теперь мы создаем 2 списка, в которых хранятся имена изображений (лиц) и их кодировки лиц.

```
path = "./training_pict/"
known_names = []
known_name_encodings = []
images = os.listdir(path)
```

■ Кодировка лица — это вектор значений, представляющих важные измерения между отличительными чертами лица, такими как расстояние между глазами, ширина лба и т. д.

■ Мы просматриваем каждое изображение в нашем каталоге training_image, извлекаем имя человека на изображении, вычисляем его вектор кодирования лица и сохраняем данные в соответствующих списках.

```
for _ in images:
    image = fr.load_image_file(path + _)
    image_path = path + _
    encoding = fr.face_encodings(image)[0]
    known_name_encodings.append(encoding)
    known_names.append(os.path.splitext(os.path.basename(image_path))[0].capitalize())
```

3. Протестируйте модель на тестовом наборе данных.

■ Как упоминалось выше, наш набор данных в нашем каталоге public_image содержит только 1 изображение со всеми людьми в нем.

■ Cv2 считывает тестовое изображение с помощью метода imread().

```
test_image = "./umumiy_rasm/do'stlar.jpg"
```

```
image = cv2.imread(test_image)
```

■ Библиотека face_recognition предоставляет полезный метод face_locations(), который определяет координаты (слева, снизу, справа, сверху) каждого лица, обнаруженного на изображении. Используя эти значения местоположения, мы можем легко найти кодировку лица.

```
face_locations = fr.face_locations(image)
```

```
face_encodings = fr.face_encodings(image, face_locations)
```

■ Учитываем расположение каждого лица и его кодировку на изображении. Затем мы сравниваем эту кодировку с кодировками лиц в наборе данных «training_image».

■ Затем вычисляем расстояние до лица, то есть определяем сходство между кодировкой тестового изображения и изображениями обучающего изображения. Теперь мы выбираем минимальное значение расстояния от него, которое указывает, что лицо на изображении common_image является одним из лиц в наборе данных training_image.

■ Теперь методами модуля cv2 рисуем прямоугольник с координатами положения лица

```
for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
```

```
    matches = fr.compare_faces(known_name_encodings, face_encoding)
```

```
    name = ""
```

```
    face_distances = fr.face_distance(known_name_encodings, face_encoding)
```

```
    best_match = np.argmin(face_distances)
```

```
    if matches[best_match]:
```

```
        name = known_names[best_match]
```

```
        cv2.rectangle(image, (left, top), (right, bottom), (0, 0, 255), 2)
```

```
        cv2.rectangle(image, (left, bottom - 15), (right, bottom), (0, 0, 255), cv2.FILLED)
```

```
        font = cv2.FONT_HERSHEY_DUPLEX
```

```
        cv2.putText(image, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)
```

Показывает изображение с помощью метода imshow() модуля Cv2.

```
cv2.imshow("Natija", image)
```

Сохраняет изображение в наш рабочий каталог с помощью метода imwrite().

```
cv2.imwrite("./Natija_rasm.jpg", image)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

REFERENCES

1. A. Müller, S. Guido. An introduction to machine learning with Python. IC "Gevissta", 2017.
2. A. Rosebrock. Deep Learning for Computer Vision with Python. Practitioner Bundle. 1st Edition. PyImageSearch (www.PyImageSearch.com), 2017.
3. A. Sweigart. Automate the Boring Stuff with Python. Practical Programming for Total Beginners. San Francisco, 2015.
4. A.B. Barsky. Neural networks: recognition, control, decision making. - Moscow: "Finance and Statistics", 2004.

Web-сайты

1. <https://www.python.org/>
2. <https://pypi.org/>
3. <https://pillow.readthedocs.io/en/3.1.x/reference/Image.html>
4. <https://matplotlib.org/>