

Московский государственный университет имени М. В. Ломоносова.

Лабораторная работа №2.

Чванкина Дарья
317 группа
Июнь 2022

Отчет по курсу "Обработка и распознавание изображений"

1 Введение

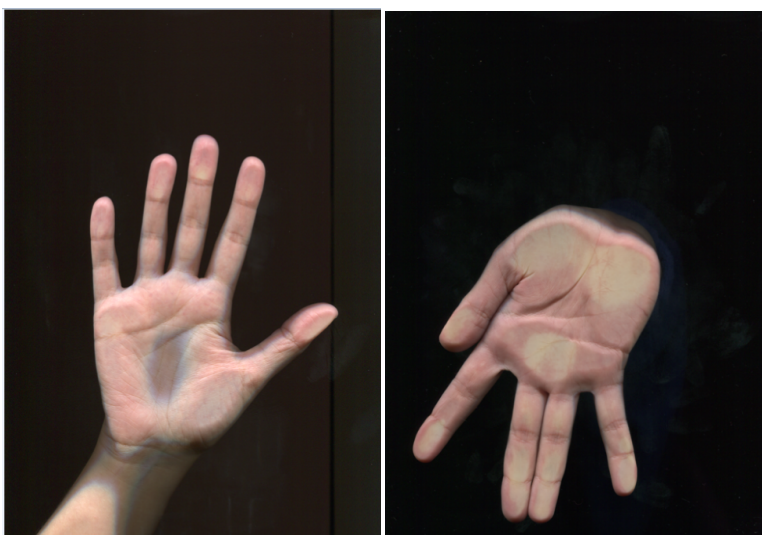
В работе с изображениями полезно выделять новые признаки, с помощью которых решаются задачи. Например, новым признаком может быть скелет изображения. В данной работе предлагается использовать изображения ладоней с пальцами. С помощью генерации новых признаков нужно определить сомкнуты или разомкнуты пальцы на изображении.

2 Постановка задачи

Задача состоит в том, чтобы разработать и реализовать программу.

На вход поступают RGB изображения в формате TIF, на которых изображены ладони с пальцами. Ладонь может быть расположена любым образом, то есть под любым углом. Пальцы также могут быть расположены любым образом, могут быть сомкнуты, могут быть разомкнуты.

Рис. 1: Примеры входных данных



На выходе программа выдает:

- Код позы ладони, описывающий сжатые и разомкнутые пальцы. Пусть большой палец будет под номером один. Тогда если большой и указательные пальцы сомкнуты, то в коде должно быть "1 + 2". Если разомкнуты - "1 - 2".
- Координаты кончиков пальцев.

В результате работы программы формируется файл "Result.txt в котором для каждого изображения в первой строке записан код позы ладони, на второй - координаты кончиков пальцев. Пример:

```
1 * 2 * 3 * 4 * 5
```

```
!,0000.tif,T Xt1 Yt1,T Xt2 Yt2,T Xt3 Yt3,T Xt4 Yt4,T Xt5 Yt5,?
```

где (Xti, Yti) – координаты пикселей i-й найденной точки, определяющих кончики пальцев T (tips).

Также программа визуализирует исходное изображение, изображение с отмеченными кончиками пальцев и кодом позы.

Ответ на задачу: 1-2-3-4+5

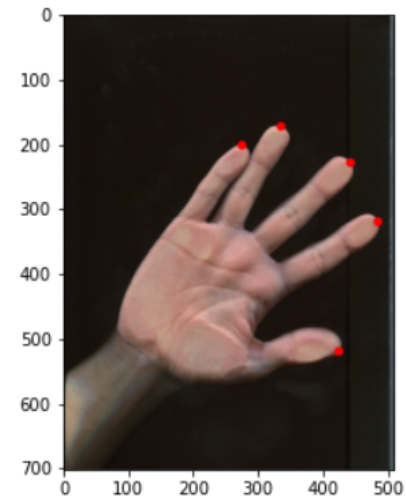
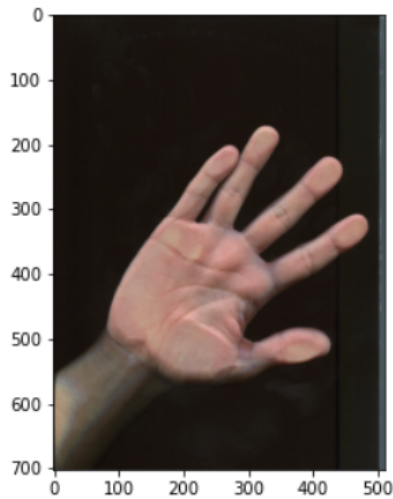


Рис. 2: Пример визуализации программы

3 Метод решения

Чтобы решить данную задачу прежде всего изображение надо бинаризовать. В данной работе бинаризация была проведена методом Оцу. Метод заключается в том, чтобы выставить порог бинаризации такой, чтобы можно было разделить изображений на два класса, одним из которых будет объект, который нас интересует. Метод сводится к минимизации внутриклассовой дисперсии, которая определяется как взвешенная сумма дисперсий двух классов:

$$\sigma_m^2 = w_1\sigma_1^2 + w_2\sigma_2^2$$

Здесь w_1 и w_2 — вероятности первого и второго классов соответственно.

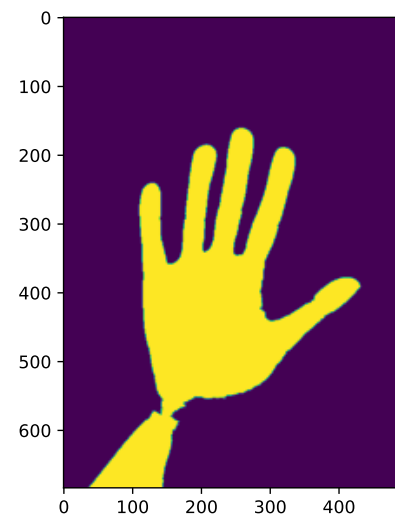
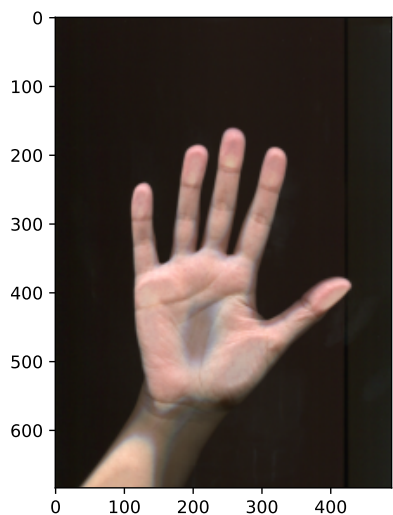


Рис. 3: Бинаризация изображения

Далее по бинаризованному изображению находится контур с помощью метода `cv.findContours`. Данный метод находит множество контуров. Находим контур, длина которого больше 400. Этот контур соответствует контуру ладони.

Далее находится центр ладони по бинаризованному изображению следующим образом:

```

m00 = binary.sum()

m01 = binary.sum(axis = 0) * np.arange(binary.shape[1])
m01 = m01.sum()

m10 = binary.sum(axis = 1) * np.arange(binary.shape[0])
m10 = m10.sum()

y = round(m10/m00)
x = round(m01/m00)

```

, где " * " - покоординатное умножение, (x, y) - координаты центра.

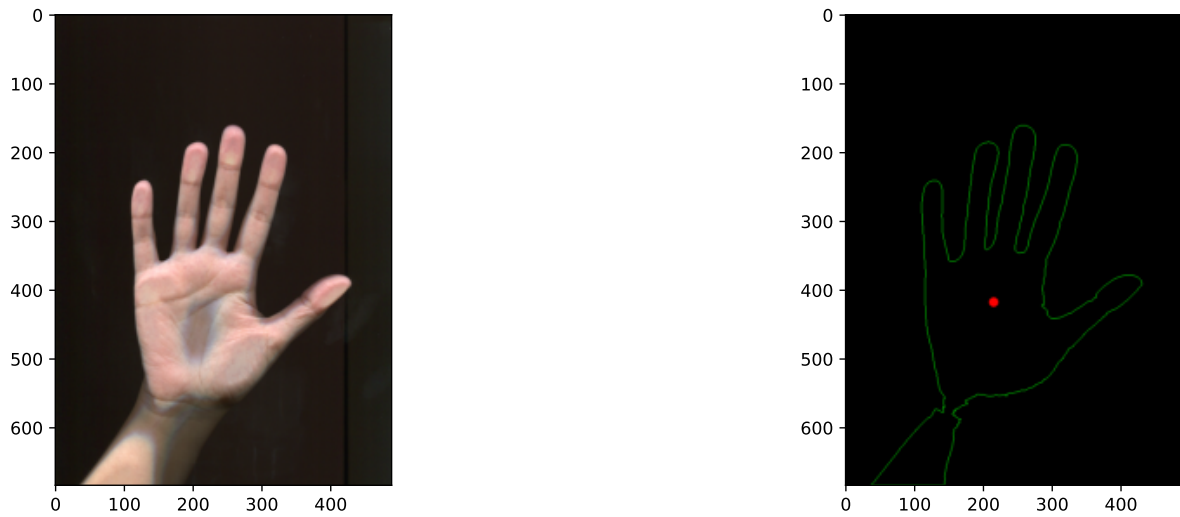


Рис. 4: Контур с центром.

Ищется расстояние от каждой точки контура до центра. Кончики пальцев - это локальные максимумы графика, где по оси x расположены номер точки на контуре, а по оси y откладывается расстояние до центра. Локальные максимумы находятся с помощью функции `findpeaks`. Далее будем называть локальные максимумы пиками.

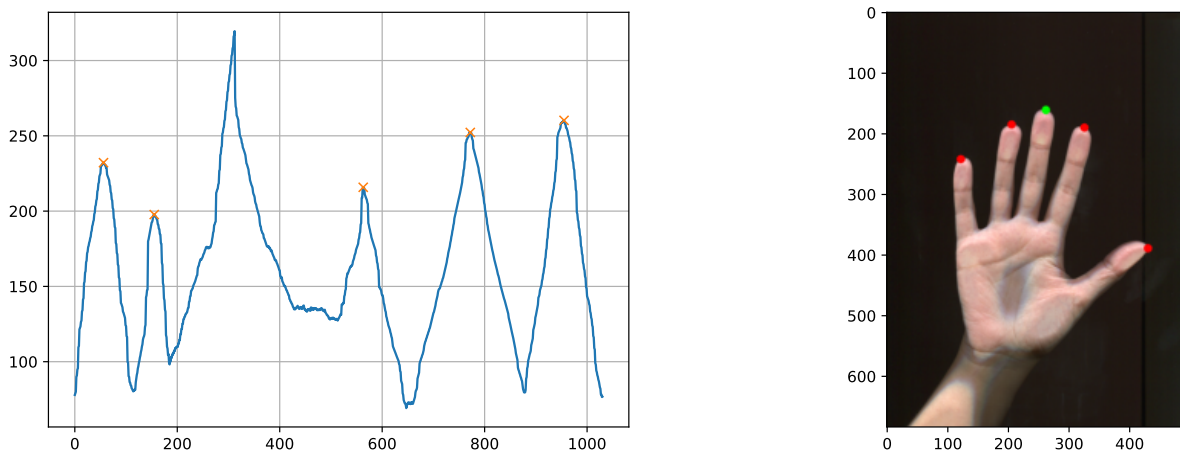


Рис. 5: График пиков и изображение с отмеченными кончиками пальцев.

В пики попадают и лишние точки. Например, иногда в контур попадает запястье или центр ладони вычислен с некоторой погрешностью Поэтому пики надо отобрать.

Предлагается сделать это следующим образом. Для начала уберем вершины, которые находятся рядом друг с другом на графике и на изображении.

Далее посчитаем среднее высот пиков, а затем для каждого пика посчитаем модуль отклонения от среднего. Уберем из пиков те точки, у которых модуль отклонения больше 62(порог вычислялся экспериментальным путем), если пиков больше чем 5.

Далее найдем средний палец. Средний палец - это палец, который расположен дальше всех от центра. Его пик будет максимальным.

Далее оставшиеся пики провернем циклически так, чтобы средний палец оказался 3-им по счету и возьмем первые пять пиков. Это и будут пики, соответствующие кончикам пальцев. Далее для каждой последовательной пары точек координат кончиков пальцев найдем расстояние между ними и по порогу определим сомкнуты или разомкнуты пальцы.

4 Программная реализация

Программа была реализована на языке python с помощью библиотеки OpenCV.

1. $H \rightarrow H_{red}$. В исходном изображении H выделяется красная компонента RGB изображения.
2. $H_{red} \rightarrow Bin$. Бинаризация по Оцу изображения, полученного на предыдущем шаге.
3. $Bin \rightarrow Contour$. Находится контур ладони.
4. $Bin \rightarrow Center$. Вычисляется центр ладони.
5. $Center, Contour \rightarrow Dist$. Формирует Dist - массив расстояний от каждой точки в контуре до центра ладони.
6. $Dist \rightarrow peaks, peaks_heights$. Находятся точки, в которых расстояние до центра максимально (максимумы локальные). Также выделяется информация о высоте пиков(см. Метод решения.).
7. $peaks, peaks_heights \rightarrow peaks, peaks_heights$. Убираем пики, которые находятся слишком близко друг к другу.
8. $peaks, peaks_heights \rightarrow var$. Находим отклонение от средней высоты пиков.
9. $peaks, peaks_heights, var \rightarrow peaks, peaks_heights$. Убираем пики, у которых отклонение от средней высоты слишком большое.
10. $peaks, peaks_heights \rightarrow middle_finger$. Сортируем пики(peaks) по их высоте, и находим средний палец, как палец с наибольшей высотой.
11. $peaks, middle_finger \rightarrow peaks_sort$. Прокручиваем(циклически) массив peaks так, чтобы на третьем месте находился средний палец.
12. $peak_sort, Contour \rightarrow tips$. В массиве peaks находятся индексы точек в массиве Contour, соответствующие кончикам пальцев. В tips под индексом 0 - большой палец, дальше по порядку - указательный, средний и т.д.
13. $tips \rightarrow res$. Находим расстояние между 1 и 2 пальцами, между 2 и 3 и т.д. Если расстояние между первым и вторым пальцами больше 174 - то в переменной res записывается "-" иначе "+". Так делаем для каждой пары пальцев со своим порогом. Для 2 и 3 пальцев порог равен 62, для 2 и 4 - 55, для 4 и 5 - 90. Пороги подбирались экспериментальным путем.
14. Записываем результат в файл и визуализируем.

5 Эксперименты

Главный недостаток данной реализации в том, что если средний палец определяется неправильно, дальше алгоритм точно ошибется.

Далее после определения среднего пальца нам никто не гарантирует, что пики слева и справа от него будут пиками, соответствующие кончикам пальцев. Таким образом на 13 изображениях из 67 кончики пальцев визуально определяются неправильно.

1+2-3-4-5

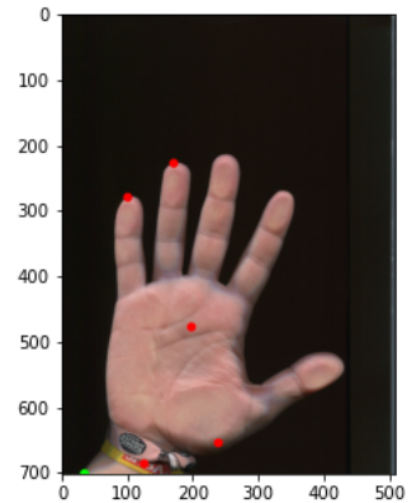
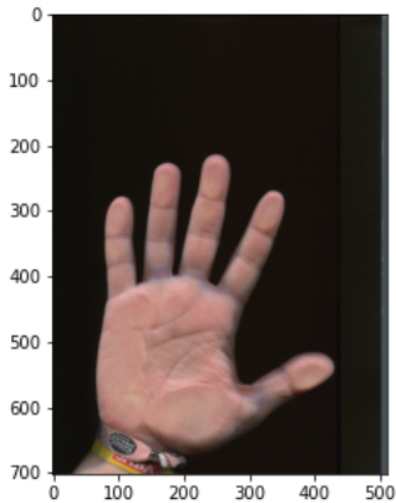


Рис. 6: Пример неправильного определения кончиков пальцев.

Иногда пороги, которые используются на этапе 13 определения сомкнуты или разомкнуты пальцы не подходят для данной картинки. Код позы определяется неверно и данный порог было бы хорошо подбирать к каждой ладони индивидуально. Итак, на 5-ти изображениях с правильно определенными кончиками пальцев код позы был неверным.

1-2+3+4-5

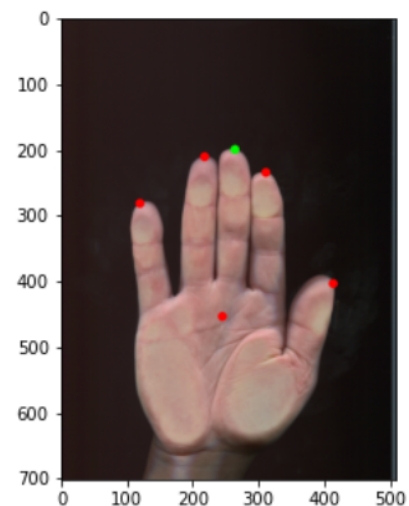
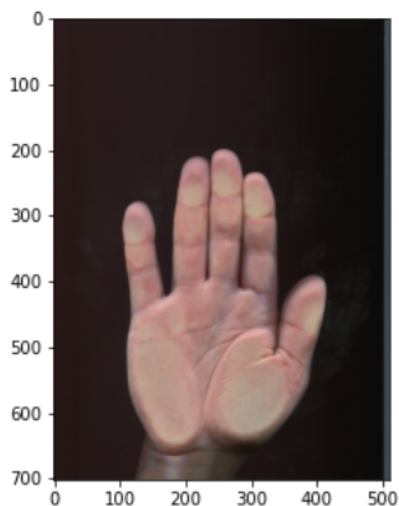


Рис. 7: Пример неправильного определения кончиков пальцев.

Итого, из 67 изображениях алгоритм ошибается на 18-ти изображениях, что соответствует точности 73%.

6 Вывод

Данный алгоритм хорошо решает задачу. Также он имеет ряд недостатков. Следующий шаг: улучшить распознавание пиков, соответствующие кончикам пальцев. Также алгоритм имеет фиксированные пороги, которые могут не подойти, так как люди уникальны и данные пороги надо подбирать исходя из параметров ладони. Предложенные алгоритм выделяет следующие признаки, которые могут помочь для других задач - центр ладони и кончики пальцев. Следующий шаг: находить точки, которые ладони между пальцами.