

# Обработка и распознавание изображений

Генерация признаков формы объектов на изображении.

Ашинов Бислан Рамазанович  
317 группа  
Кафедра ММП ВМК МГУ

Москва 2022

# 1 Введение

В этой работе изучаются и применяются методы обработки изображений для задачи классификации ладони на изображении.

## 2 Постановка задачи

Имеется набор изображений, на которых скан ладони на черном фоне . Используя пространственные и точечные преобразования изображений, необходимо:

- Определить позу ладони: для каждой пары соседних пальцев определить прижаты ли пальцы
- Записать ответ в виде последовательности "1\*2\*3\*4\*5".
- Найти кончики пальцев и точки разгибов пальцев.
- Провести линии пальцев на изображении

## 3 Данные

Набор изображений в формате ".tif". Каждое изображение содержит фотографию ладони левой руки на черном фоне. На Рис.1 приведен один из образцов.

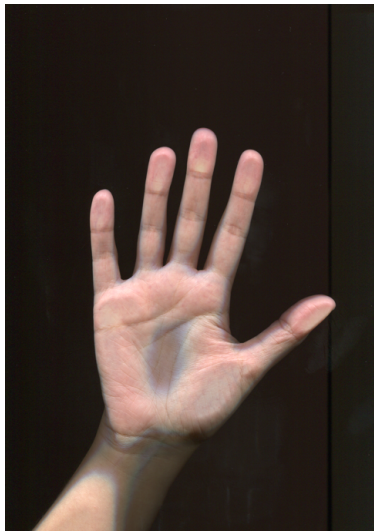


Рис. 1: Образец 1

## 4 Метод решения

К изображению применяется дилатация и размытие (чтобы сгладить неровности ладоней по краям, которые могут помешать поиску точек разгиба). После чего изображение бинаризируется с применением метода Оцу.

Далее ищем замкнутые контуры на изображении и выбираем наибольший по площади (очевидно, этот контур является контуром ладони). На Рис.2 показан пример бинаризованного изображения с отобранным контуром.



Рис. 2: Результат бинаризации

Следующий шаг - сделать аппроксимацию контура, то есть некоторые очень извилистые линии сгладить отрезком. Так мы получим более грубый контур, на котором почти все изгибы соответствуют важным точкам ладони - кончикам пальцев, точкам разгиба и т.д.



Рис. 3: Пример контура с выпуклой оболочкой

Получив грубый контур ладони, построим его выпуклую оболочку - точки контура, между которыми есть невыпуклость, соединяем отрезком. На Рис.3 показан пример.

Далее находим дефекты контура (точки невыпуклости). Из них будем отбирать точки, которые соответствуют точкам разгиба пальцев. Для каждого дефекта получим точку начала (точка контура, совпадающая с точкой выпуклой оболочки, после которой начинается невыпуклость), точку окончания невыпуклости, точку контура ладони, наиболее удаленную от выпуклой оболочки, расстояние от нее до выпуклой оболочки.

С помощью этих точек, можем найти угол в наиболее удаленной точке, образованный началом и концом дефекта. Найти можем по теореме косинусов - знаем координаты всех трех точек. Отберем дефекты, которые образуют угол меньше  $5\pi/4$ . Из них выкинем те, расстояние от которых до выпуклой оболочки очень мало. Таким образом мы избавимся от всяких неровностей ладони.

Из выбранных дефектов выбираем 4 наиболее удаленных от выпуклой оболочки и принимаем их за точки разгибов пальцев.

Если в дефекте расстояние до выпуклой оболочки очень большое (порог подобран примерно, исходя из вычисленных примеров), то считаем, что пальцы не прижаты, то есть зазор между ними очень большой, иначе - что прижаты.

Далее надо найти точку отсчета пальцев - точку разгиба большого пальца и указательного. Точность этого метода оставляет желать лучшего, поэтому можно заметить, что на изображениях довольно точно определяется прижатость пальцев, а в некоторых строках вида "1\*2\*3\*4\*5" произошел циклический сдвиг.

Мы знаем порядковые номера точек на контуре, в которых начинаются и заканчиваются контуры. Посчитаем разницу в номерах точек для соседних дефектов, то есть

$$start_i - end_{i-1}$$

Дефект, для которого это число максимальное будет разгибом между большим пальцем и указательным (так как от между мизинцем и большим пальцем зачастую очень много точек контура ладони).

За кончики пальцев примем середину отрезка, концами которого являются начало и конец соседних дефектов.

## 5 Программная реализация

Метод реализован на языке Python3 с использованием библиотеки OpenCV.

В файле `utils.py` реализованы функции:

- *get\_thimb\_extension* - получает на вход контур, дефекты и возвращает номер дефекта - разгиба между большим и указательным
- *to\_bin* - получает на вход RGB изображение и преобразует в бинарное
- *get\_approx\_ccontour* - получает на вход бинарное изображение и возвращает аппроксимированный контур
- *find\_extensions* - получает на вход контур и возвращает точки невыпуклостей, соответствующие разгибам пальцев
- *markup\_image* - получает на вход контур, дефекты, выдает ответ на задачу, делает пометки на изображении при указывании флага

## 6 Эксперименты

На Рис.4 и Рис.5 показаны примеры работы программы. Красный символ на изображении показывает в точке разгиба прижаты ли пальцы. Зеленая ломаная - линия пальцев.

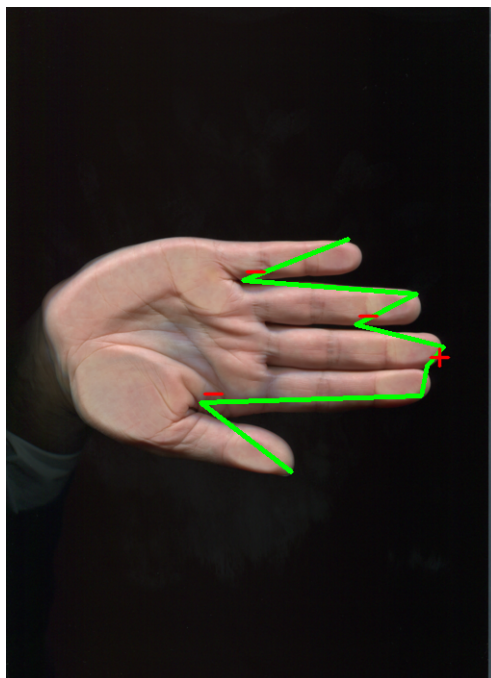


Рис. 4: Пример 1 работы программы

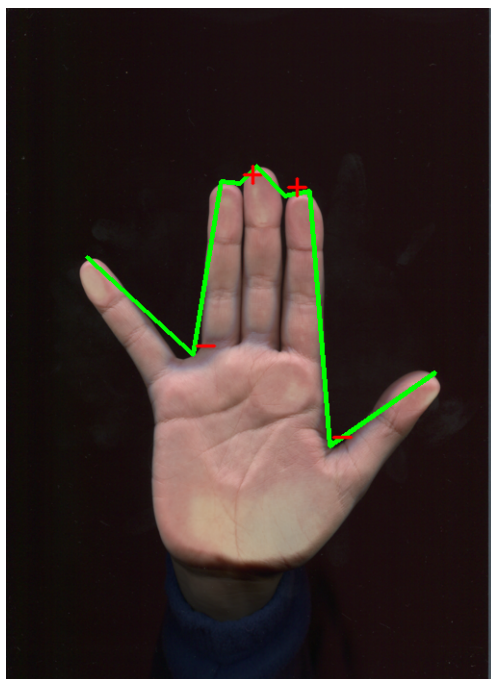


Рис. 5: Пример 2 работы программы

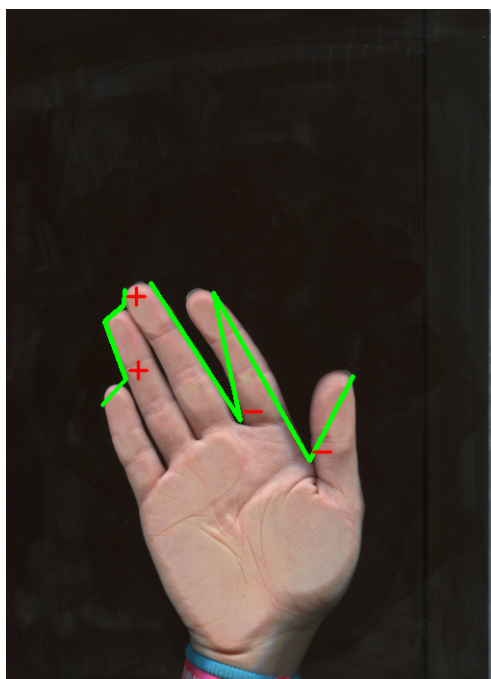


Рис. 6: Пример 3 работы программы

На Рис.7 показан пример неверной работы программы. Здесь запястье не было отделено от ладони, из-за чего контур захватил также часть руки. Из-за чего неверно были найдены разгибы и т.д.

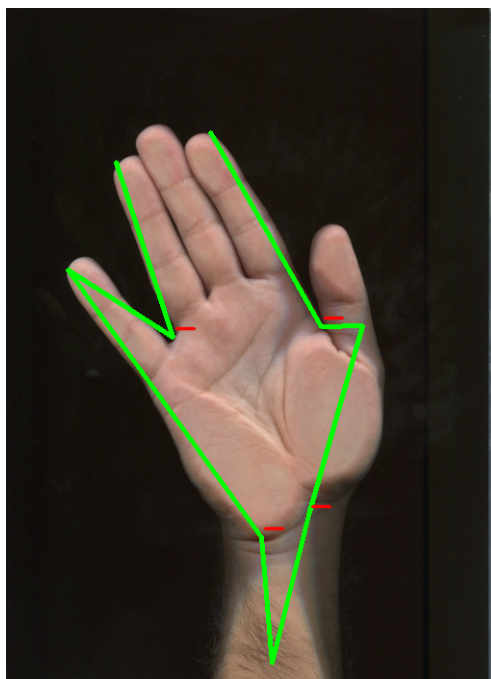


Рис. 7: Пример 4 работы программы

## 7 Выводы

Как можно видеть, с помощью преобразований изображений можно выделять такие важные точки/признаки объектов на изображении, как кончики пальцев, точки разгибов и т.д. Выделение ладони и нужных сгибов было самым сложным в поставленной задаче. На этом этапе алгоритм совершает наибольшее количество ошибок.