

Лабораторная работа №2. Отчет. курс «Обработка и распознавание изображений».

Пацация Александр, 317 группа

9 мая 2022 г.

1 Постановка задачи

Целью задания является разработка программы для работы с фотографиями ладоней на темном фоне. Требуется:

1. Определить позу ладони (уровень сложности **Intermediate**). Пальцы нумеруются от 1 до 5, начиная с большого против часовой стрелки. Поза описывается кодом мида "1*2*3*4*5", где вместо "*" ставится "-" - если пальцы разомкнуты, "+" - если сомкнуты.
2. Определить линию пальцев – ломанную линию, соединяющую точки на кончиках пальцев с точками в основаниях пальцев (уровень сложности **Expert**).

Оба пункта предполагают визуализацию результата и запись полученных данных в файл. Сначала в файл записывается код позы, на следующей строке - координаты кончиков и оснований.

2 Описание данных

Входные данные - трёхканальные фотографии ладоней рук на чёрном фоне в формате TIF. На рис. 1 приведены примеры входных изображений.

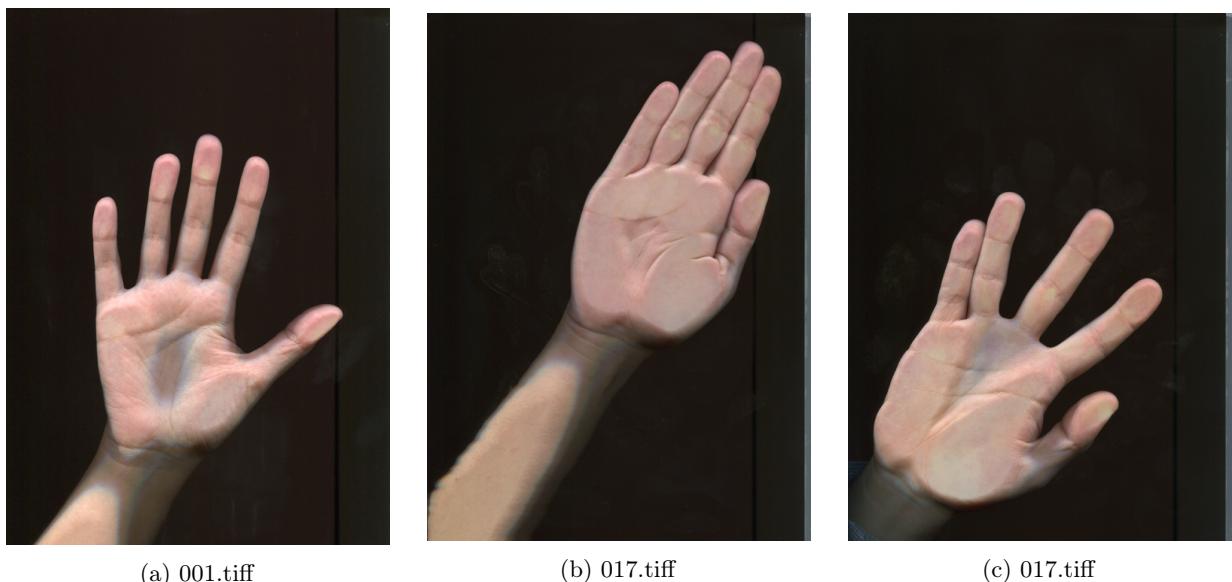


Рис. 1: Примеры входных изображений

На каждой фотографии ладонь занимает определенное положение, которое и требуется распознать.

3 Метод решения

Решение состоит из нескольких этапов:

1. Сегментация руки.
2. Семантическая разметка руки, получения информации о пальцах.
3. Определение позиции ладони требуемом формате
4. Определение линии пальцев.

3.1 Сегментация руки

В первую очередь необходимо получить бинарное изображение руки с четким разделением пальцев. В основном проблемы возникают с сомкнутыми пальцами на изображении. Для начала, получим одноканальную картинку, переведя исходное изображение в канал красного цвета. Далее применим фильтр Собеля к одноканальному изображению. Проведем бинаризацию обоих изображениям методом Оцу. Далее, из бинаризованного изображения вычтем бинаризованный результат применения фильтра Собеля. В полученном изображении заполним небольшие отверстия. Описанный способ дает неплохое результирующее изображение и хорошо разделяет сомкнутые пальцы (рис. 2). Также, по возможности, удаляются объекты не относящиеся к ладони.

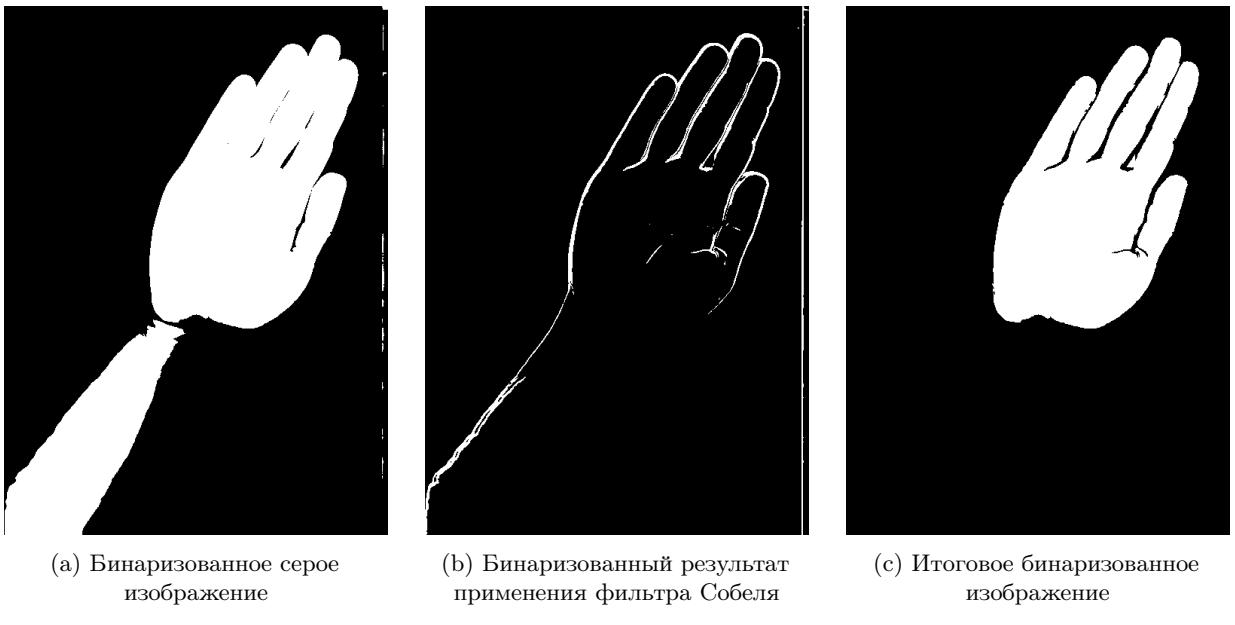
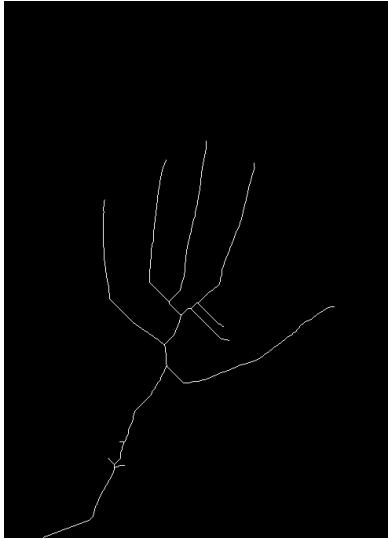


Рис. 2: Шаги бинаризации

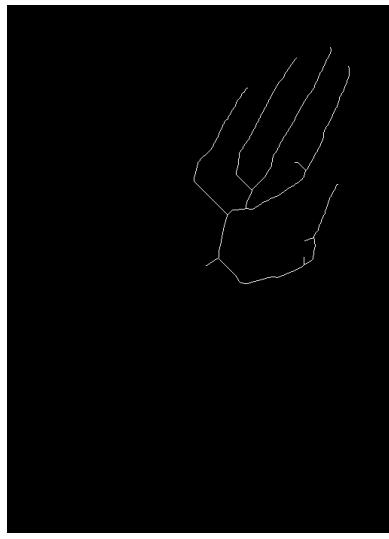
3.2 Разметка пальцев

переходим к основной части решения - семантическая разметка, для которой используется скелет бинарного изображения ладони. В данном методе решения для построения скелета используется дискретный алгоритм Зон-Суня.

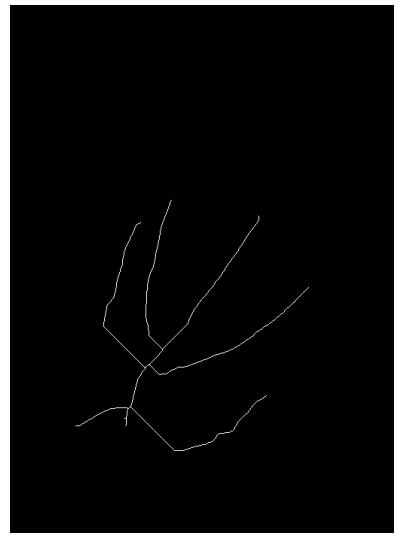
Перед построением скелета, применим к бинаризованному изображению фильтр Гаусса с дисперсией $\sigma = 1$. Это поможет добиться более регулярного скелета алгоритмом Зон-Суня. Пример скелета приведен на рис. 3. Помимо скелета, также, для каждого пикселя хранится значение радиуса максимальной окружности.



(a) скелет 001.tiff



(b) скелет 017.tiff



(c) скелет 044.tiff

Рис. 3: Примеры скелетов

После получения скелета, необходимо отобрать ветви пальцев. Для начала собирается информация о всех ветвях: координаты, степень, расстояние до центра концевых вершин каждой ветви. Центральной точкой ладони объявляется точка соответствующая центру наибольшей максимальной окружности всего скелета. На следующем этапе, происходит дополнительная "стрижка" скелета, удаляются ветви длины меньше определенного порога и достаточно далекие от центра. Таким образом мы избавляемся от мелких ветвей, загрязняющих скелет. Далее снова происходит пересчет параметров для концевых вершин каждой ветви (после стрижки информация могла измениться). На следующем этапе происходит удаление коротких изолированных ветвей (степень обеих концевых вершин равна 0) и очередной пересчет параметров. Оставшиеся ветви типа узел-терминальная вершина признаются кандидатами в пальцы. Для кандидатов в пальцы вычисляется положение "низа" пальца. Низом признается вершина скелета, удовлетворяющая следующим условиям:

$$\begin{cases} r_c < r_p \\ r_c < R_{max} \\ r_c > r_t \\ d_c > 1.2R_{max} \\ |(r_C, r_t)| < 150 \end{cases},$$

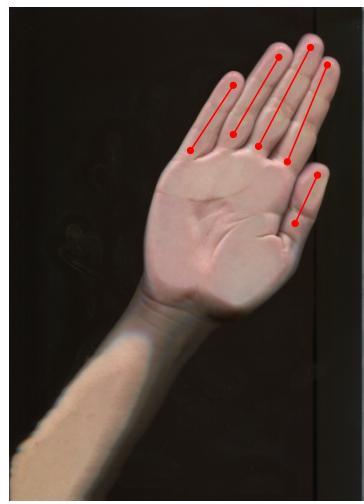
где r_c - радиус максимальной окружности, соответствующий основанию, r_p - радиус максимальной окружности, точке, предшествующей основанию, r_t - радиус максимальной окружности, соответствующий верхушке (терминальной вершине) ветви, d_c - расстояние низа до центра, R_{max} - радиус наибольшей максимальной окружности скелета, $|(a, b)|$ - означает мощность пути из a в b . Если под эти условия подходят несколько вершин, то выбирается наиболее удаленная от верхушки ветви. Если значение радиуса максимальной окружности для некоторых низов ниже некоторого экспериментально подобранного порога, то соответствующая ему ветвь удаляется из кандидатов в пальцы. Если после всех описанных действий кандидатов все еще больше 5, ветви сортируются по значению следующего эвристического коэффициента:

$$c = \sqrt{|branch|} \frac{d_t}{d_b},$$

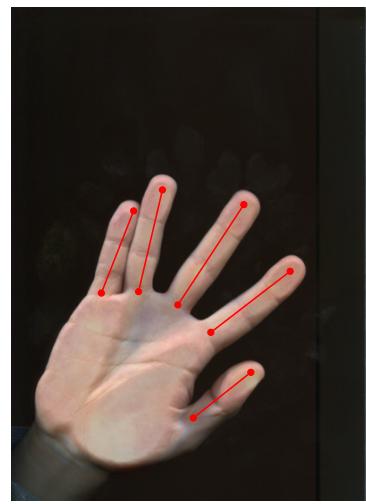
где d_t, d_b - расстояние от центра до верхушки и низа соответственно, $|branch|$ - количество вершин (мощность) ветви. Данный коэффициент хорошо отбирает ветви соответствующие пальцам, действительно: нам требуется найти ветви низы которых достаточно близки к центру, а верхушки достаточно удалены, при этом ветвь должна быть достаточно длинной. По этому коэффициенту отбирается 5 ветвей с наибольшим значением. В итоге имеем основания и верхушки пальцев (рис. 4).



(a) разметка пальцев 001.tiff



(b) разметка пальцев 017.tiff



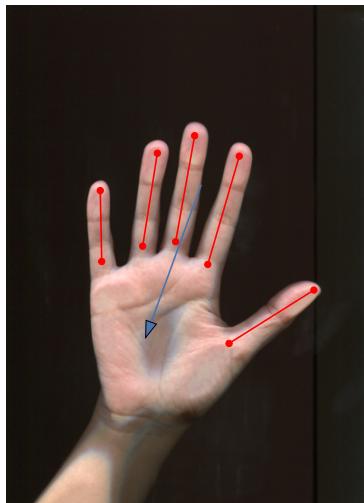
(c) разметка пальцев 044.tiff

Рис. 4: Примеры разметки пальцев

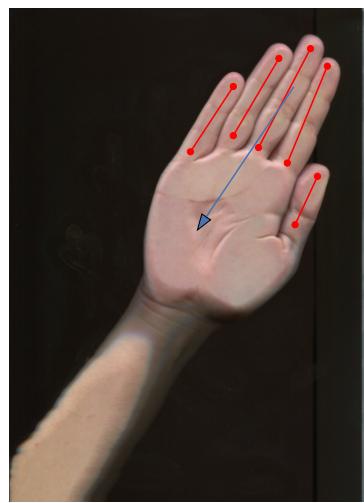
Далее необходимо отсортировать пальцы против часовой стрелки. Для этого необходимо корректно определить опорный вектор относительно которого будет проводиться сортировка. В данном методе был выбран следующий вектор (рис. 5):

$$\text{ref vec} = P_c - P_m,$$

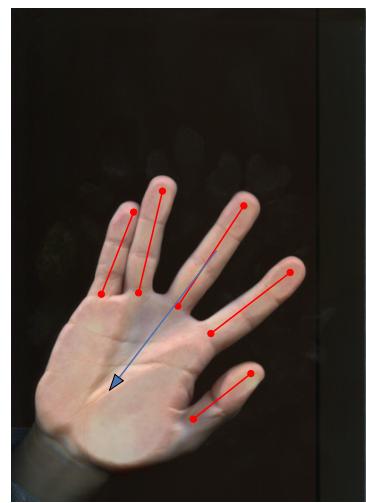
где P_c - координаты центра наибольшей максимальной окружности, P_m - координаты точки, представляющей из себя среднее между всеми верхушками пальцев. На этом стадия разметки пальцев завершается.



(a) опорный вектор 001.tiff



(b) опорный вектор 017.tiff



(c) опорный вектор 044.tiff

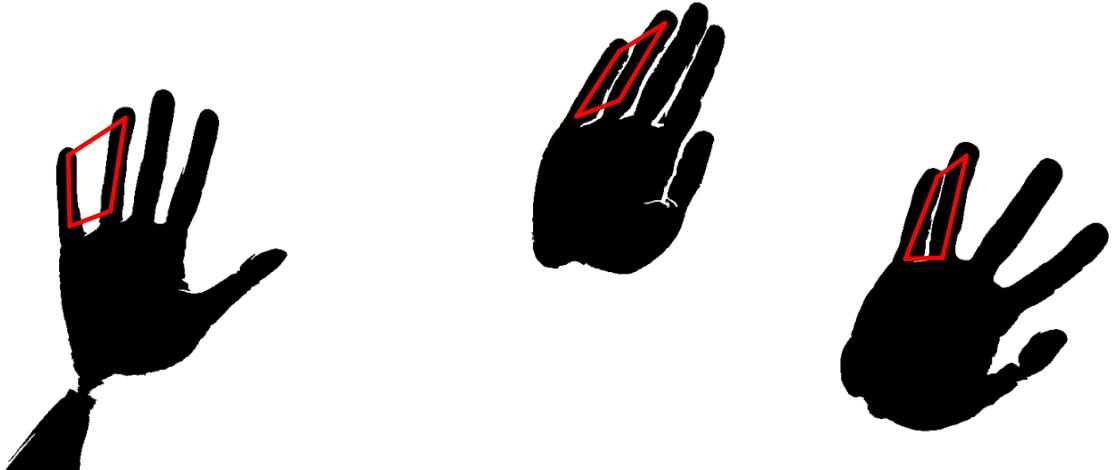
Рис. 5: Примеры опорных векторов

3.3 Определение позы ладони

Теперь перейдем к определению позы ладони. Для определения "сомкнутости" или "разомкнутости" пальцев используется следующий подход:

1. Вычисляются координаты всех пикселей, которые попадают в четырехугольник, образованный верхушками и основаниями соседних пальцев (рис. 6).
2. Высчитывается коэффициент равный отношению количества "черных пикселей" (то есть тех, что относятся к ладони) бинарного изображения ладони ко всему количеству пикселей в четырехугольнике.

Если значение коэффициента выше некоторого порога (подобран экспериментально для каждой пары пальцев), то пара признается сомкнутой, иначе разомкнутой. Из минусов данного подхода можно отметить вычислительную сложность, однако он дает хороший результат по качеству.



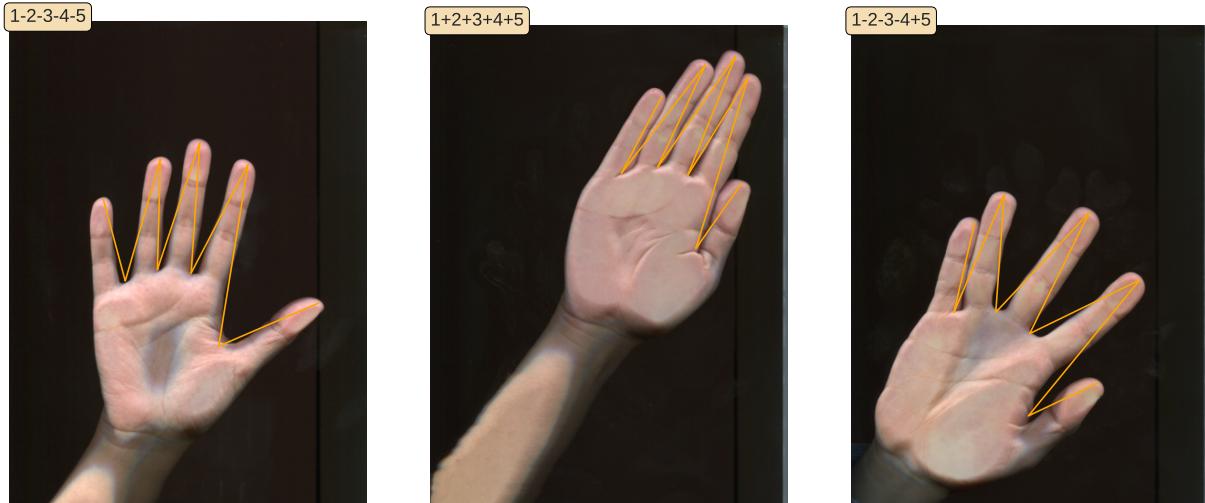
(a) Четырехугольник для 001.tif
 (b) Четырехугольник для 017.tif
 (c) Четырехугольник для 044.tif

Рис. 6: Примеры четырехугольников для последней пары пальцев ладоней

3.4 Определение линии пальцев

Имея данные о верхушках и низах пальцев нетрудно найти координаты кончиков и оснований. Верхушки определяются как ближайшие к верхушкам точки пересечения границы с прямой, образованной верхушкой и низом пальца. Основания определяются следующим образом:

1. Вычисляется среднее арифметическое между двумя основаниями соседних пальцев v_1 .
 2. Вычисляется множество точек границы, лежащих между прямыми, образованными верхушками и основаниями. Из этого множества удаляются точки отстающие от v_1 на расстояние выше некоторого порога.
 3. Среди точек этого множества выбирается, точка ближайшая к центру ладони - v_2 .
 4. Вычисляется взвешенная сумма $v = \alpha * v_2 + (1 - \alpha) * v_1$, если определяется основание пары пальцев, среди которых нет большого, и $v = \alpha * v_2 + (1 - \alpha) * b_{thumb}$ - иначе, где b_{thumb} - низ большого пальца.
- Результаты приведены на рис. 7.



(a) Результат работы метода для 001.tif

(b) Результат работы метода для 017.tif

(c) результат работы метода для 044.tif

Рис. 7: Результат работы метода

4 Программная реализация

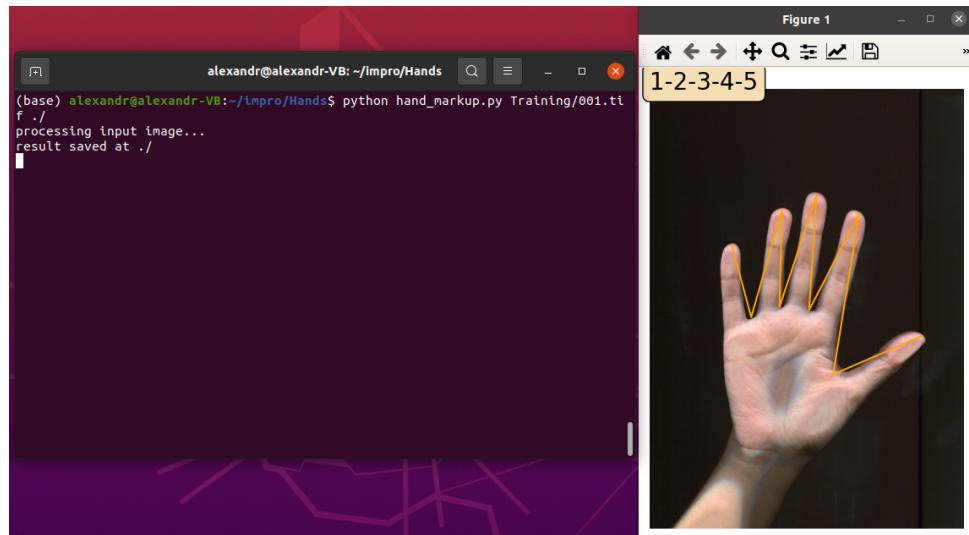
Решение реализовано на языке Python с использованием следующих библиотек:

1. библиотека **scikit-image** для работы с изображениями.
2. библиотека **skan** для работы со скелетом изображения.
3. библиотека **matplotlib** для разметки и вывода изображений.

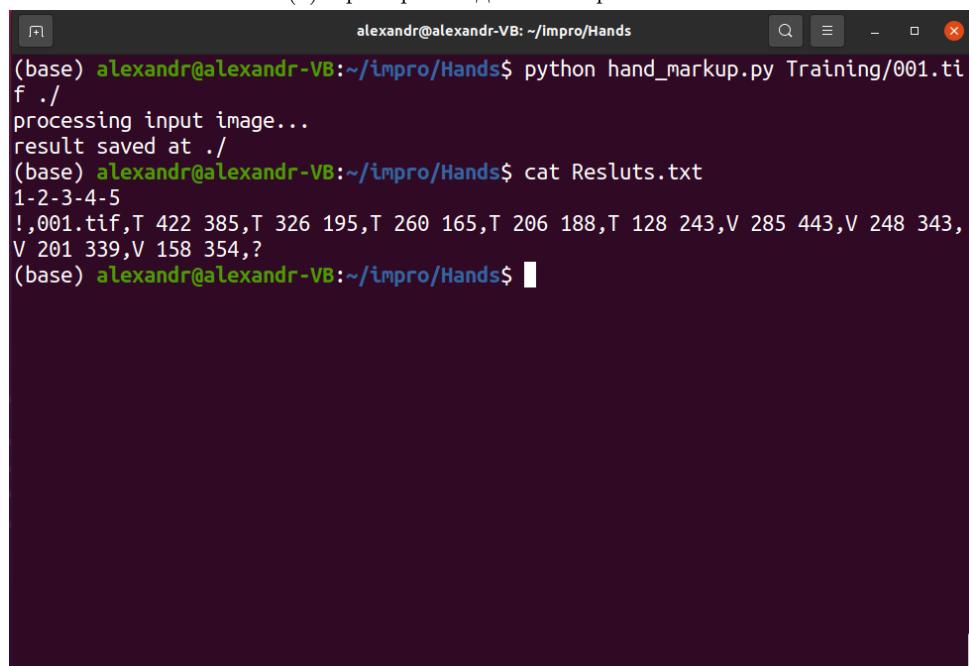
Запуск программы с помощью командной строки:

```
python hand_markup.py input_image output_directory
```

input_image – путь к входному изображению. **output_directory** – путь к выходной директории. Если **output_directory** не указан, по умолчанию выходное изображение сохраняется в текущей директории. Программа обрабатывает изображение, помещает выходной файл в указанную директорию и выводит его на экран. Формат выходного файла – TIF.



(a) Пример выходного изображения



(b) Пример выходного файла

Рис. 8: Пример работы программы

5 Эксперименты

В предыдущих разделах подробно рассматривалась работа программы для 001.tif, 0017.tif, 044.tif. Рассмотрим работу алгоритма для других изображений (рис. 9).

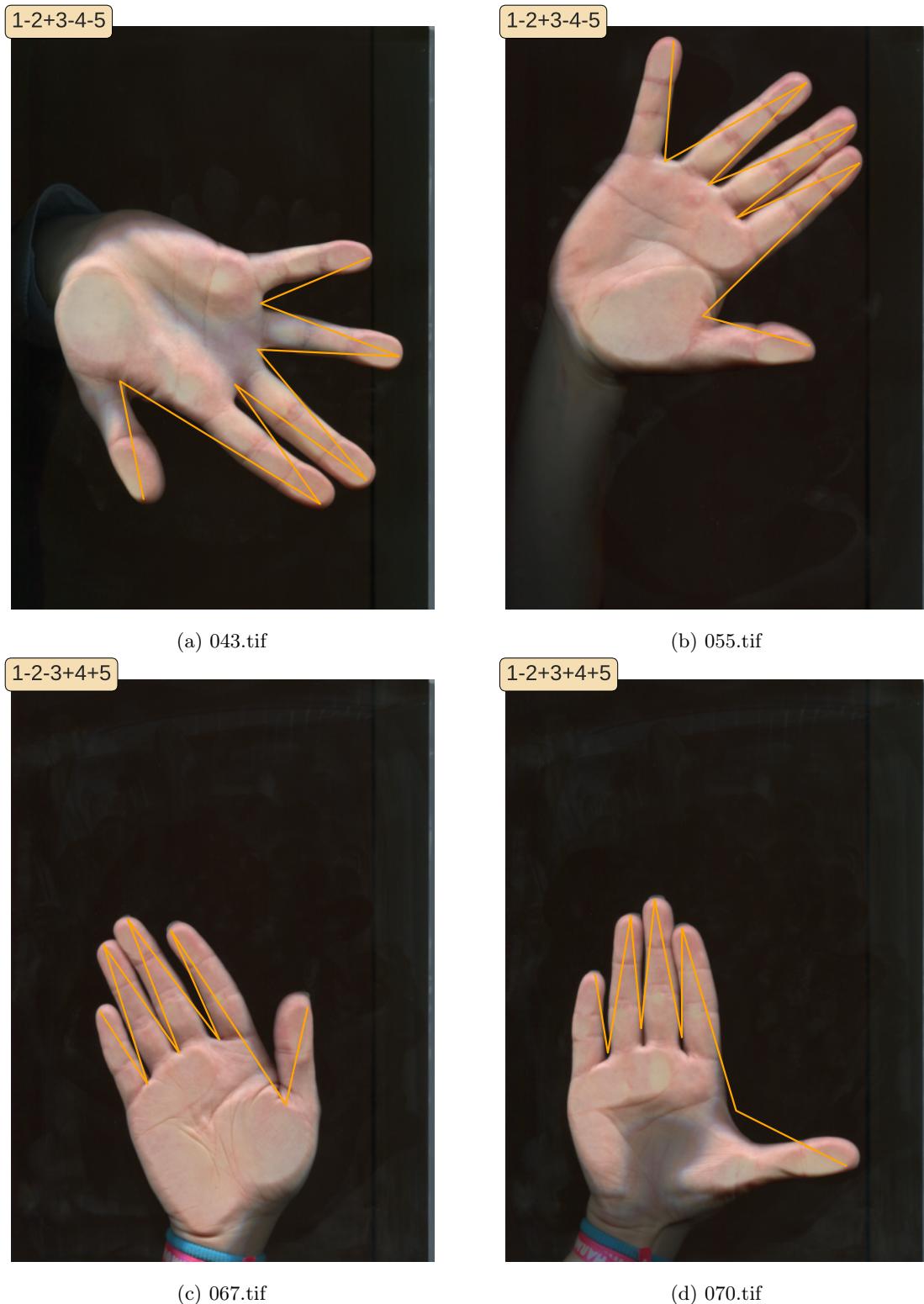
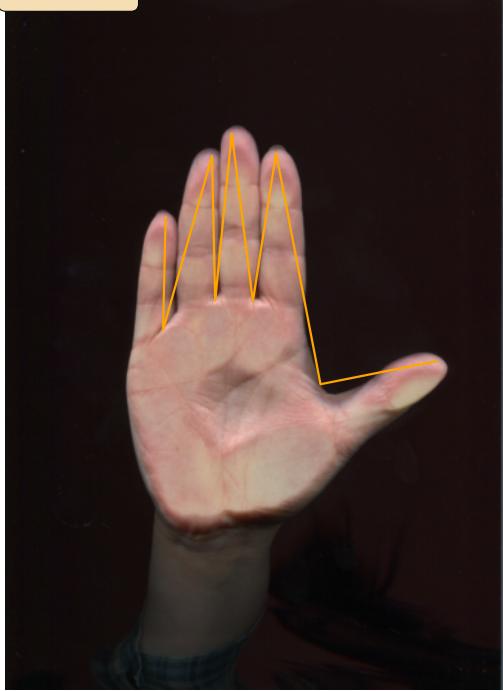


Рис. 9: Пример работы программы

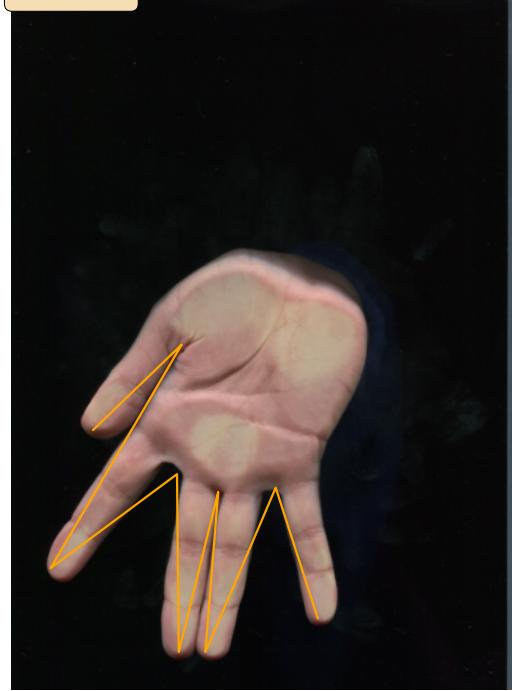
Как можно видеть (070.tif) иногда возникают проблемы с определением оснований пальцев. Это связано с тем, что "низы" пальцев на некоторых изображениях оказываются слишком близкими к верхушкам и основания смешаются вверх. Однако поза ладони находится очень точно. Рассмотрим еще примеры (рис. 10).

1-2+3+4+5



(a) 271.tif

1+2-3+4-5



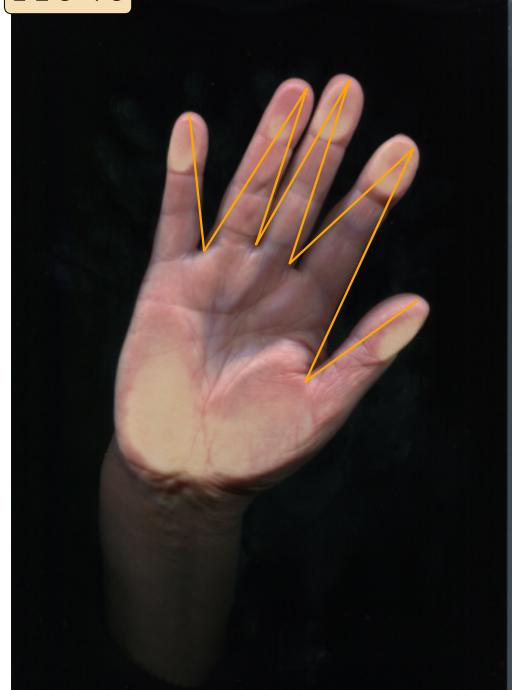
(b) 280.tif

1-2-3+4-5



(c) 295.tif

1-2-3+4-5



(d) 300.tif

Рис. 10: Пример работы программы

На примере 300.tif, 295.tif можно сказать, что определить сомкнуты или разомкнуты пальцы иногда сложно даже человеку. В целом, на тренировочной выборке алгоритм ошибается в определении позы ладони только на неочевидных даже для человека фотографиях.

6 Выводы

По итогам лабораторной работы была разработана программа для распознавания ладоней. Программа определяет позу ладони и линию пальцев, основанная на анализе скелета бинарного изображения.

Определение позы на тренировочной выборке показывает очень хорошее качество. Иногда возникают проблемы с определением оснований пальцев, иногда результат оказывается "смещенным". В целом, подход с использованием скелета показал достойные результаты.