## Московский Государственный Университет

# Генерация признаков формы объектов на изображении

Выполнил: Курцев Д.В.

Группа: 317

Факультет Вычислительной математики и кибернетики Кафедра Математических методов прогнозирования

## 1 Постановка задачи

Необходимо разработать и реализовать программу для классификации изображений ладоней, обеспечивающую:

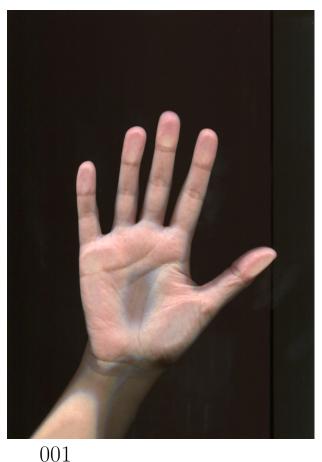
- Ввод и отображение на экране изображений ладоней в формате TIF;
- Сегментацию изображений на основе точечных и пространственных преобразований;
- Определение «позы» ладоней;
- Определение «линии пальцев» ладоней.

Поза ладони определяется по расположению сомкнутых пальцев. Пальцы нумеруются от 1 до 5, начиная с большого против часовой стрелки. Поза ладони описывается кодом 1\*2\*3\*4\*5, где значок «\*» обозначает «-», если пальцы разомкнуты, или «+», если они прижаты друг к другу.

Линия пальцев – ломаная линия, соединяющая точки на кончиках пальцев (tips) с точками в основаниях пальцев (valleys). Пример такой линии представлен на рисунке, она имеет 9 вершин и 8 звеньев.

## 2 Описание данных

В качестве исходных данных прилагается набор из 67 цветных изображений ладоней разных людей, полученных с помощью сканера, в формате 489×684 с разрешением 72 dpi. На изображениях представлены сканы левых рук. Задача состоит в определении позы ладоней и линии пальцев на основе выделения и анализа формы ладоней.





017

#### Ход работы и программная реализация 3

В качестве программного инструмента будем использовать библиотеки OpenCV и MediaPipe.

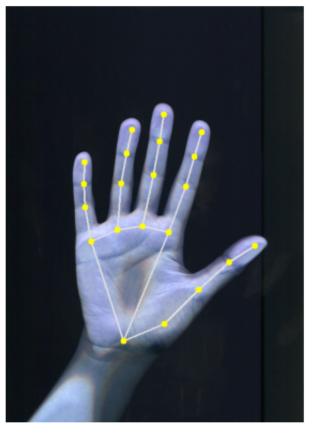
Для начала считаем наше изображение в RGB формате с помощью функции cv2.imread, которая на вход принимает путь до картинки.

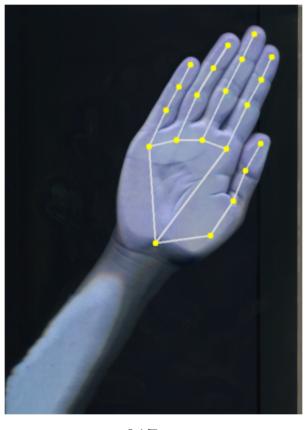
Далее для решения задачи, необходимо построить скелет ладони. Для это воспользуемся готовыми методами класса *Hands* из библиотеки MediaPipe. MediaPipe Hands использует конвейер машинного обучения, состоящий из нескольких моделей, работающих вместе: модель обнаружения ладони, которая работает с полным изображением и возвращает ориентированную ограничительную рамку. Модель ориентира руки, которая работает с обрезанной областью изображения, определенной детектором ладони, и возвращает высококачественные 3D-ключевые точки руки. Более подробно про работу данного метода можно прочитать здесь [1].

С помощью функции process получим наш результат. И далее,

воспользовавшись методом  $multi\_hand\_landmarks$ , проитерируемся и получим некоторые точки скелета. if-блок (в функции  $get\_tips\_and\_valleys$ ) проверяет были ли вообще найдены ладони (любой не пустой объект является истинным). Далее циклом for мы «перебираем» объект с набором этих точек. Второй for — перебираем сами точки из набора.

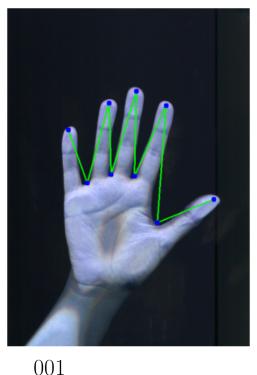
Таким образом, на выходе получим скелетное представление ладони с некоторыми ключевыми точками:





001

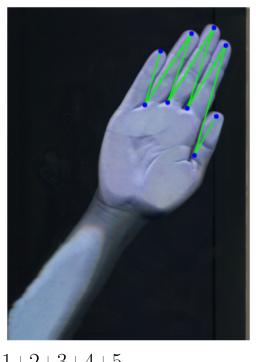
Как можно заметить, верхние точки находятся чуть ниже, чем tips, которые необходимо получить. Поэтому мы сдвинем их на четверть по вектору пальца вверх. Таким образом, мы попадём практически в кончик пальца. Чтобы получить valleys, возьмём нижние точки у основания пальцев. Точно так же сдвинем их вверх на треть по вектору. Для получения итоговой точки усредним данные значения. Далее соединим все точки в ломанную линию и выведем готовый ответ. Для отрисовки линий использовалась функция cv2.line, а для точек cv2.circle.





1 017

Чтобы определить сжаты пальцы или размокнуты, будем смотреть на угол, образованный ломанной у их основания. Эмипирически лучшим значением оказалось 22. Если угол меньше 22°, то пальцы прижаты друг к другу, иначе они разомкнуты.





1+2+3+4+5 1-2+3-4-5

Заметим, что программа верно распознаёт позу ладони(её код), а также линии пальцев на изображени.

Для удобства пользователя предыдущие пошаговые действия были обёрнуты в функцию process\_image. Она на вход принимает 3 аргумента. Путь до директории, в которой лежат изображения, имя файла с картинкой и имя файла, в который будет записан результат. Кодировка и координаты точек отдельно никуда не выводятся, а сразу же записываются в файл с ответом.

## 4 Экспериметы

Стоит отметить, что сначала предпринимались собственные попытки реализации различных методов генерации признаков. Скелеты зачастую получались очень плохими. А методы поиска valleys, основанные на дефектах выпуклой оболочки контура, не работают с сжатыми пальцами. Поэтому автор использовал помощь нейросетевых архитектур.

Продемонстрируем работу нашей программы на нескольких изображениях:



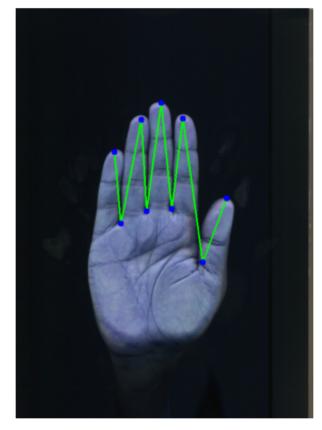




1+2-3+4-5

Видно, что она всегда верно распознаёт линии пальцев, т.е. верно находит все tips и valleys и соединяет их. И практически во всех данных изображениях даёт верную кодировку ладони, кроме нескольких случаев. Проблемным местом в определении позы ладони является связка указательного и большого пальцев. Продемонстрируем несколько примеров, где модель ошибается:



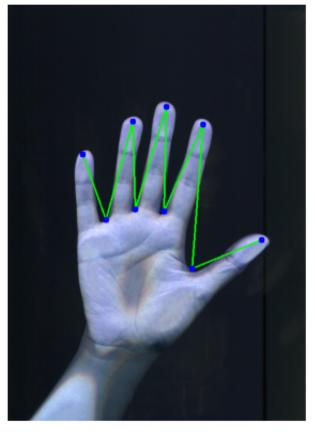


1-2+3-4+5

1-2+3+4+5

На левом изображении угол между большим и указательным пальцами равен 24°, а на правой 28°. Вероятно, такая ошибка связана с тем, что кончик большого пальца плохо прижат к указательному. Тем самым угол становится больше, чем нужно и не удаётся, чётко отследить его позицию.

Так же в датасете есть несколько изображений, на которых сложно сказать прижаты пальцы или нет. Приведём несколько примеров:





1-2-3+4-5 1-2+3+4-5

Скорее всего на левом изображении безымянный и средний пальцы разомкнуты. Но угол между ними получился всего 18°. Эта ошибка может быть связана с тем, что эти пальцы находятся слишком близко друг к другу, поэтому программа их объединяет. Похожая ситуация наблюдается на правой картинке. Непонятно указательный и средний разомкнуты или нет. С одной стороны между ними есть просвет, с другой они очень близки. Если всё же наша модель ошибается в таких ситациях, в будущем придётся искать дополнительные методы для проверки позы ладони. Была произведена попытка смотреть на расстояния между кончиками пальцев, но она не увенчалась успехом...

### 5 Выводы

Задача была во многом решена благодаря современным методам обработки изображений. В этом помогли библиотеки языка программирования *Python OpenCV* и *MediaPipe*.

Для успешного решения поставленной задачи необходимо строить скелет изображения. Рассматривалось несколько способов его реализации на основе библиотек OpenCV и scikit-image. Но данные алгоритмы показывали очень плохое качество, поэтому автор использовал более продвинутые методы, уже реализованные в библиотеке MediaPipe.

Получилось полностью верно определить линию пальцев и практически во всех случаях точно распознать кодировку ладни. Ошибки с позой могут быть связаны, как с методом реализации, так и с не совсем качественными фотографиями. Не на всех изображениях удаётся точно определить разомкнуты пальцы или нет. Для этого необходимо иметь больше примеров с разметкой.

## 6 Литература

```
[1]: \verb|https://google.github.io/mediapipe/solutions/hands.html|
```

[2]: https://www.section.io/engineering-education/creating-a-fi

[3]: https://tproger.ru/articles/pishem-sistemu-raspoznavanija-