

МГУ им.Ломоносова
Факультет ВМК кафедра ММП

ЛАБОРАТОРНАЯ РАБОТА №2

Изучение и освоение методов сравнения формы изображений
в рамках курса "Обработка и распознавание изображений"

Батшева Анастасия 317 группа

Москва
2021

Содержание

| | | |
|----------|--|-----------|
| 1 | Формулировка задания | 2 |
| 2 | Входные данные | 2 |
| 3 | Описание метода и экспериментальная часть | 3 |
| 3.1 | Сегментация | 3 |
| 3.2 | Векторизация | 5 |
| 3.3 | Кластеризация | 9 |
| 4 | Итоги | 12 |
| 5 | Программная реализация | 13 |
| 6 | Выводы | 13 |

1 Формулировка задания

Разработать и реализовать программу для классификации изображений ладоней, обеспечивающую:

1. Ввод и отображение на экране изображений в формате .tif
2. Сегментацию изображений на основе точечных и пространственных преобразований
3. Генерацию признаков описаний формы ладоней на изображениях
4. Вычисление меры сходства ладоней
5. Кластеризацию изображений

Нужно разработать и реализовать алгоритм, входом которого являются эти изображения, а выходом – описание признаков формы, попарные расстояния, кластеры изображений.

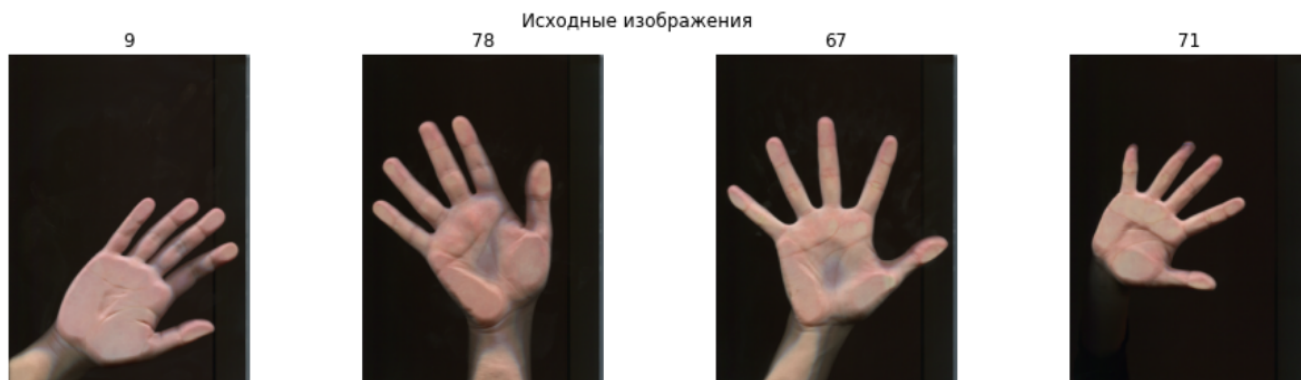
Конкретизация пунктов задания для класса **Expert**:

1. Найти на изображении ладони точки в кончиках и основаниях пальцев
2. Визуализировать результат для экспертного контроля
3. Найти для каждой ладони 3 наиболее похожих изображения и представить результат в виде таблицы «имя образца – имена ближайших соседей»
4. Определить число людей, чьи ладони представлены в изображениях, и составить списки ладоней для каждого, т.е. провести кластеризацию изображений в виде таблицы «Персона No – имена изображений ладоней»

2 Входные данные

В качестве исходных данных прилагается набор из 99 цветных изображений ладоней разных людей, полученных с помощью сканера, в формате 510 × 702 с разрешением 72 dpi.

Примеры входных изображений представлены на рисунках:



3 Описание метода и экспериментальная часть

Поскольку решаются три задачи: **сегментация**, **векторизация** и **кластеризация**, то методы описываются последовательно.

3.1 Сегментация

Сегментация включает в себя четыре основных пункта:

1. Вычисление масок для некоторой подвыборки изображений аналитическим путем:
2. Аугментация полученных пар изображение-маска до необходимого для сети размера
3. Тренировка сети
4. Применение сети ко всем имеющимся изображениям

Размерем каждый пункт более подробно:

Вычисление масок:

1. Перевод в полутоновое изображение
2. Бинаризация для выделения области руки по среднему
3. Бинаризация с подобранным порогом (0.85 - квантиль) для "отрезания" кисти от самой руки (почти на каждом изображении возле перехода от кисти к руке видно затемнение из-за сгиба)
4. Area opening и clear border для очистки от лишних краевых эффектов и случайных пятен
5. Поиск выпуклой оболочки ладони (для заполнения больших пустот в маске)
6. Совмещение имеющихся бинаризаций.



На выходе мы получаем корректные маски для десяти изображений:



Важное уточнение: загрузка картинок происходила НЕ по их номерам в директории, а по порядку. Тем не менее везде (если не сказано обратное), где есть подписи изображений в виде номеров - эти номера соответствуют **ИСТИННЫМ** названиям изображений.

Аугментация или создание датасета:

Каждый сэмпл (пара изображение-маска) подвергается трансформациям:

1. Поворот на -180, -135, -90, -45, 0, 45, 90, 135 - итого 8 вращений
2. Сдвиг по горизонтали и вертикали от -40 до 40 с шагом в 20 - итого по 5 на каждую ось = 25
3. Изменение яркости в половину - 2

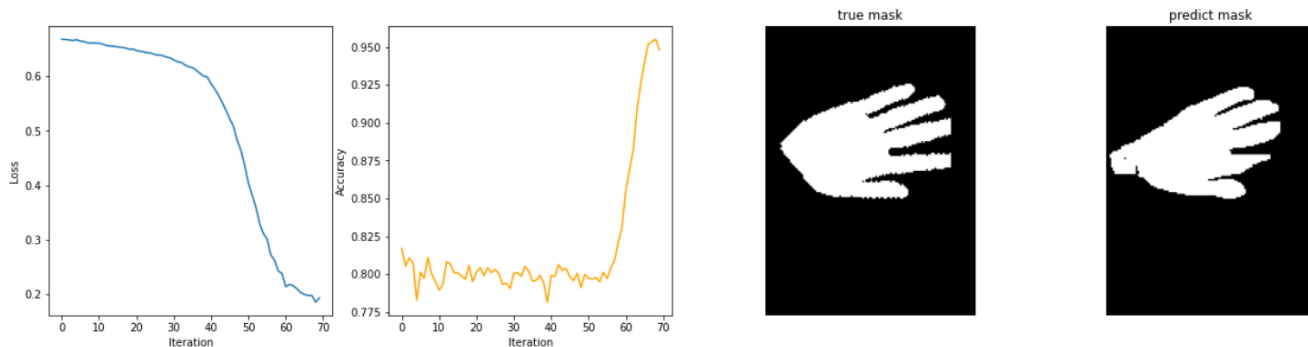
В результате на одно изображение из исходных приходится 400 новых.
Объем датасета = **4000** картинок.

Выборка сэмплов из обучающего датасета



Процесс обучения сети:

На датасете тренируется обычный UNet с предобученным vgg13. По батчам длины 16. Классика. В целом, в тренировке нет ничего необычного и нового. Оптимизатор - Adam, Ошибка - BinaryLogLoss.



Применение сети:

Итоги нейросетевой сегментации:



Видно, что результат соответствует ожиданиям и дает великолепное качество.

3.2 Векторизация

В задании было предложено выделить особую линию пальцев, однако это несколько смущает: одна и та же рука может в таком случае соответствовать двум абсолютно различным векторам, если на фотографиях человек будет по-разному раздвигать пальцы (чем больше угол, тем больше становится каждая линия-гипотенуза в ломаной). Поэтому было принято решение разработать более точный подход по тому же принципу, в результате включающий в себя такие параметры как:

1. Длина и ширина каждого пальца - итого 10
2. Площадь и периметр ладони - итого 2
3. Распределение интенсивности цвета для каждого канала в пределах от 0.2 до 0.8 включительно - итого $7 * 3 = 21$

Разберем типы параметров подробнее:

Вычисление длины и ширины:

1. Определение области ладони без пальцев с помощью erosion и delation: пальцы, как самые тонкие объекты ладони, будут "вытравлены" эрозией раньше, чем плоская часть посередине
2. Определение оставшихся связных областей - пальцев (разумеется, внутри еще разные очистки от ненужных атрибутов, таких как порог по площади)
3. Для каждой связной области (пальца) встроенными методами skimage ищется аппроксимирующий эллипс, а так же его радиусы.
4. Итоговые значения записываются в словарь признаков данного изображения.



Однако после 2 пункта стоит сделать лирическое отступление: некоторые пальцы оказались склеены, потому в алгоритм подсчета параметров пальцев был добавлен алгоритм разделения. Алгоритм просто изучает область склеенных пальцев на исходном изображении, на котором они, очевидно, делимы.



Итоги вычисления длины и ширины для экспертного контроля:



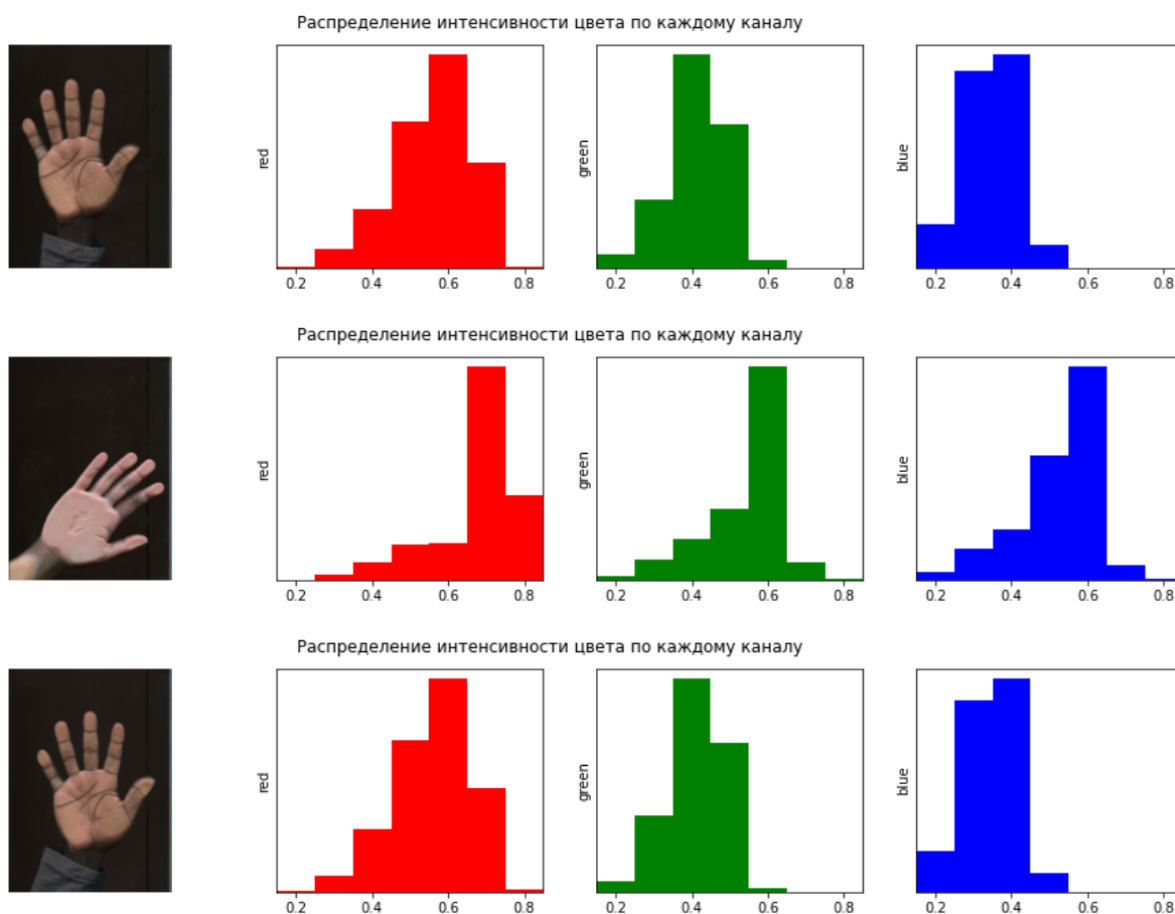
Однако на самом деле длины и ширины пальцев мало для качественной кластеризации, ведь руки, различные по общей ширине и длине или цвету, всё равно окажутся одинаковыми с точки зрения алгоритма.

Поэтому помимо **длины** и **ширины** пальцев были добавлены **периметр**, **площадь** и **параметры спектра** ладони. Ведь, действительно, человек при классификации скорее обратит внимание на цвет ладони (во всяком случае я именно так распознавала одинаковые).

Распределение интенсивности цветов:

Каждый новый цветовой признак обозначается как **c-d**, где **c** = **r** для красного канала, **g** для зеленого и **b** для синего, а **d** лежит в пределах от 0.2 до 0.8 включительно с шагом в 0.1 и означает долю пикселей интенсивности **d** в нужном канале. Уточню, что все цвета лежат в пределах от 0 до 1 по значению.

Можно предположить и легко убедиться, что гистограммы распределения цветов будут сильно отличаться у различных изображений, и быть схожими у изображений одной руки.



Результаты векторизации:

Итоговые вектора, пронумерованные в ПОРЯДКЕ ЗАГРУЗКИ изображений:

| | l_1 | w_1 | l_2 | w_2 | l_3 | w_3 | l_4 | w_4 | l_5 | w_5 | ... | g_0.6 | g_0.7 | g_0.8 | b_0.2 | b_0.3 | b_0.4 | b_0.5 | b_0.6 | b_0.7 | b_0.8 |
|-----------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 0 | 124.84 | 36.33 | 128.85 | 38.52 | 153.39 | 40.29 | 156.64 | 41.56 | 164.33 | 42.46 | ... | 0.581 | 0.047 | 0.001 | 0.018 | 0.071 | 0.114 | 0.281 | 0.482 | 0.033 | 0.001 |
| 1 | 131.64 | 39.03 | 134.87 | 52.54 | 150.71 | 43.62 | 158.96 | 46.22 | 159.78 | 45.77 | ... | 0.198 | 0.008 | 0.000 | 0.007 | 0.076 | 0.314 | 0.516 | 0.087 | 0.000 | 0.000 |
| 2 | 132.04 | 36.37 | 134.31 | 47.14 | 154.36 | 38.99 | 156.93 | 42.96 | 163.99 | 43.93 | ... | 0.568 | 0.091 | 0.004 | 0.002 | 0.016 | 0.117 | 0.599 | 0.242 | 0.024 | 0.000 |
| 3 | 82.43 | 31.90 | 107.05 | 35.43 | 115.04 | 32.26 | 129.47 | 36.00 | 130.26 | 35.11 | ... | 0.542 | 0.064 | 0.005 | 0.024 | 0.074 | 0.151 | 0.469 | 0.265 | 0.016 | 0.001 |
| 4 | 108.70 | 34.43 | 126.30 | 44.16 | 141.98 | 38.78 | 148.92 | 44.65 | 153.09 | 43.24 | ... | 0.548 | 0.134 | 0.001 | 0.007 | 0.025 | 0.083 | 0.401 | 0.458 | 0.026 | 0.000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 94 | 105.76 | 37.12 | 128.79 | 35.30 | 152.31 | 38.27 | 160.23 | 42.11 | 163.25 | 39.24 | ... | 0.556 | 0.054 | 0.001 | 0.007 | 0.055 | 0.157 | 0.551 | 0.213 | 0.016 | 0.000 |
| 95 | 105.79 | 54.61 | 115.46 | 36.50 | 143.87 | 45.60 | 149.69 | 43.49 | 156.20 | 48.31 | ... | 0.020 | 0.000 | 0.000 | 0.088 | 0.406 | 0.455 | 0.051 | 0.000 | 0.000 | 0.000 |
| 96 | 135.66 | 40.66 | 142.19 | 44.79 | 162.88 | 43.82 | 171.33 | 41.08 | 174.68 | 43.49 | ... | 0.182 | 0.001 | 0.000 | 0.038 | 0.143 | 0.349 | 0.429 | 0.041 | 0.000 | 0.000 |
| 97 | 127.52 | 34.91 | 141.18 | 45.47 | 148.74 | 36.90 | 161.38 | 43.19 | 162.81 | 39.78 | ... | 0.638 | 0.076 | 0.000 | 0.001 | 0.020 | 0.114 | 0.569 | 0.292 | 0.004 | 0.000 |
| 98 | 119.24 | 56.83 | 129.24 | 40.82 | 163.84 | 44.63 | 164.99 | 42.55 | 175.22 | 43.82 | ... | 0.346 | 0.054 | 0.001 | 0.025 | 0.180 | 0.248 | 0.406 | 0.133 | 0.008 | 0.000 |

l - длина каждого пальца, w - его ширина

Важно учесть, что длины располагаются по возрастанию, так как были заранее отсортированы (параллельно с параметрами ширины), чтобы избежать ошибок при разном расположении рук.

3.3 Кластеризация

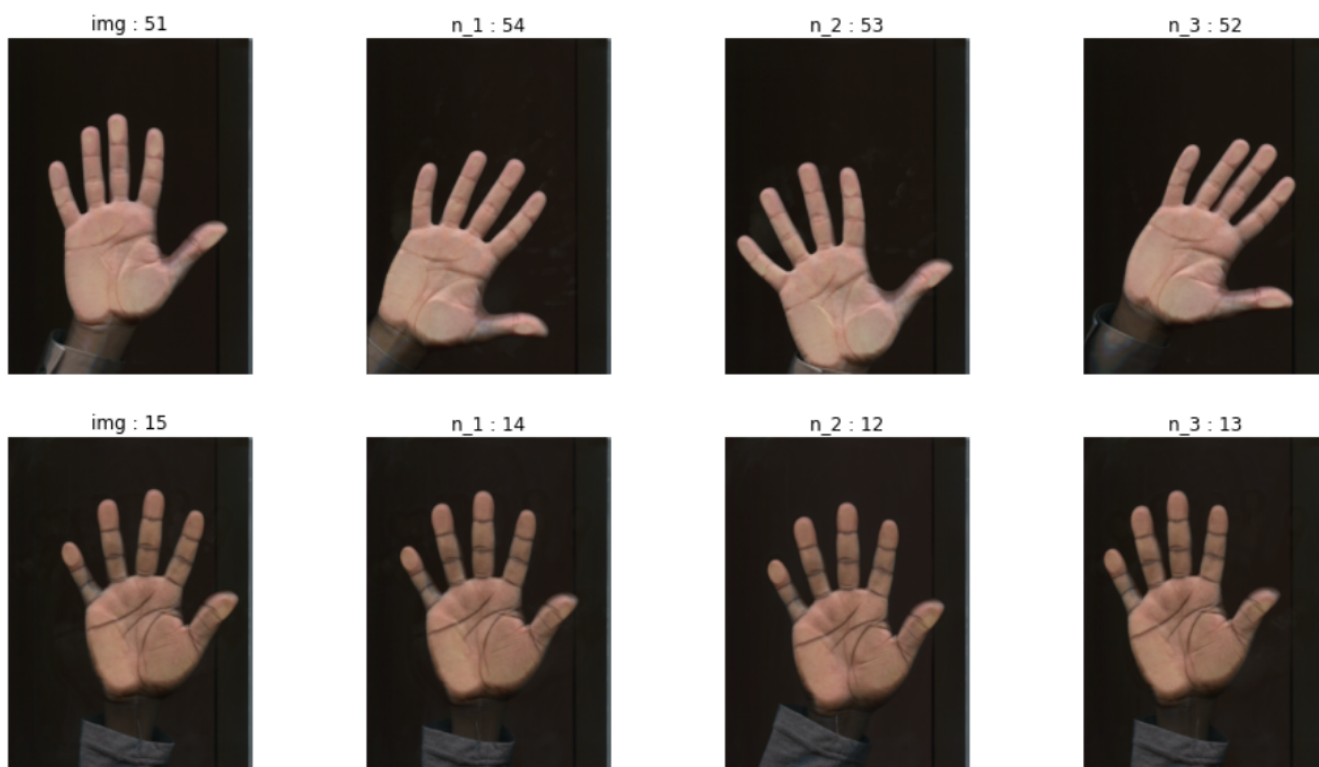
На данном этапе были вычислены ближайшие соседи (3) и общие соседи по кластеру.

Поиск 3 ближайших соседей:

1. Классические преобразования исходной аблицы признаков: шкалирование и нормализация для равновесности данных - получение матрицы **X**
2. Вычисление попарных евклидовых расстояний между объектами-изображениями (точнее их векторами) - получение матрицы **dist**
3. Сортировка расстояний по возрастанию и отбор 3 первых, исключая 0-го (расстояние между одним и тем же объектом = 0) - получение матрицы **neighbors-3**

Результаты в виде таблицы изображений на следующей странице.

| 0 - 20 | | | | | 20 - 40 | | | | | 40 - 60 | | | | | 60 - 80 | | | | | 80 - 100 | | | | |
|--------|-----|-----|-----|--|---------|-----|-----|-----|--|---------|-----|-----|-----|--|---------|-----|-----|-----|--|----------|-----|-----|-----|--|
| name | n_1 | n_2 | n_3 | | name | n_1 | n_2 | n_3 | | name | n_1 | n_2 | n_3 | | name | n_1 | n_2 | n_3 | | name | n_1 | n_2 | n_3 | |
| 9 | 10 | 11 | 8 | | 97 | 37 | 6 | 144 | | 142 | 82 | 34 | 141 | | 95 | 113 | 92 | 112 | | 91 | 145 | 6 | 49 | |
| 78 | 77 | 76 | 29 | | 46 | 79 | 77 | 51 | | 28 | 29 | 26 | 27 | | 55 | 54 | 52 | 53 | | 57 | 56 | 67 | 11 | |
| 67 | 56 | 86 | 57 | | 49 | 150 | 71 | 152 | | 24 | 37 | 31 | 8 | | 157 | 155 | 96 | 99 | | 31 | 127 | 23 | 112 | |
| 71 | 150 | 152 | 49 | | 111 | 93 | 109 | 96 | | 79 | 77 | 46 | 76 | | 53 | 54 | 51 | 55 | | 150 | 49 | 71 | 152 | |
| 37 | 39 | 41 | 6 | | 127 | 126 | 122 | 123 | | 1 | 4 | 3 | 2 | | 129 | 128 | 66 | 31 | | 113 | 112 | 114 | 118 | |
| 12 | 15 | 14 | 13 | | 81 | 29 | 26 | 12 | | 112 | 113 | 114 | 118 | | 146 | 144 | 41 | 37 | | 86 | 67 | 56 | 88 | |
| 4 | 3 | 1 | 2 | | 82 | 20 | 13 | 15 | | 51 | 54 | 53 | 52 | | 8 | 11 | 67 | 24 | | 155 | 157 | 109 | 35 | |
| 77 | 78 | 79 | 76 | | 90 | 2 | 50 | 1 | | 15 | 14 | 12 | 13 | | 54 | 55 | 52 | 51 | | 118 | 120 | 112 | 122 | |
| 152 | 151 | 50 | 71 | | 22 | 23 | 120 | 112 | | 141 | 138 | 64 | 82 | | 50 | 152 | 151 | 71 | | 114 | 112 | 113 | 118 | |
| 145 | 151 | 152 | 49 | | 52 | 55 | 54 | 51 | | 17 | 18 | 21 | 19 | | 34 | 71 | 152 | 50 | | 39 | 37 | 88 | 151 | |
| 13 | 14 | 15 | 12 | | 106 | 107 | 105 | 88 | | 20 | 82 | 15 | 14 | | 29 | 28 | 26 | 27 | | 60 | 150 | 41 | 107 | |
| 122 | 127 | 126 | 118 | | 128 | 129 | 66 | 19 | | 92 | 112 | 96 | 93 | | 11 | 9 | 8 | 57 | | 66 | 128 | 129 | 19 | |
| 21 | 19 | 18 | 17 | | 41 | 146 | 150 | 37 | | 96 | 93 | 92 | 112 | | 5 | 6 | 7 | 41 | | 3 | 4 | 1 | 2 | |
| 135 | 71 | 152 | 49 | | 144 | 146 | 41 | 150 | | 35 | 109 | 127 | 126 | | 138 | 141 | 64 | 13 | | 88 | 86 | 106 | 39 | |
| 6 | 5 | 37 | 97 | | 64 | 63 | 26 | 82 | | 99 | 7 | 128 | 66 | | 76 | 78 | 77 | 79 | | 65 | 57 | 56 | 6 | |
| 27 | 26 | 29 | 28 | | 63 | 64 | 28 | 29 | | 23 | 22 | 113 | 112 | | 107 | 105 | 106 | 60 | | 14 | 15 | 13 | 12 | |
| 10 | 9 | 11 | 6 | | 124 | 126 | 123 | 127 | | 7 | 99 | 5 | 60 | | 47 | 71 | 49 | 152 | | 26 | 27 | 29 | 28 | |
| 126 | 127 | 124 | 122 | | 68 | 53 | 120 | 46 | | 109 | 35 | 126 | 127 | | 16 | 18 | 51 | 21 | | 56 | 57 | 67 | 86 | |
| 105 | 107 | 106 | 60 | | 151 | 152 | 150 | 50 | | 19 | 21 | 18 | 51 | | 120 | 118 | 112 | 22 | | 36 | 27 | 12 | 15 | |
| 93 | 96 | 92 | 112 | | 18 | 19 | 21 | 17 | | 2 | 1 | 4 | 3 | | 123 | 124 | 127 | 126 | | | | | | |



Результат очень хорош. Радует глаз. Остальные шедевры в ноутбуке.

Определение кластеров:

Сначала были предприняты попытки задействовать встроенные алгоритмы из sklearn (из модуля cluster), однако без знания об истинной разметке или хотя бы истинного числа классов, нормально кластеризовать не получалось, поэтому было решено действовать самостоятельно. Было видно, что алгоритм качественно ищет ближайшего соседа, поэтому метод кластеризации был основан на данных только об одном ближайшем соседе. Строился **граф**, после чего выяснялась его связность.

Результат - **27** кластеров, т.е 27 персон приложило руку к сканеру с точки зрения алгоритма.

Результаты в виде таблицы и изображений:

0 - 9

| name | | | | | | |
|------|-----|-----|-----|-----|-----|--------|
| 0 | 8 | 9 | 10 | 11 | | |
| 1 | 76 | 77 | 46 | 78 | 79 | |
| 2 | 65 | 67 | 86 | 56 | 57 | 88 |
| 3 | 34 | 135 | 71 | 47 | 49 | 150 60 |
| 4 | 24 | 97 | 37 | 39 | | |
| 5 | 12 | 13 | 14 | 15 | | |
| 6 | 1 | 2 | 3 | 4 | 90 | |
| 7 | 145 | 50 | 151 | 152 | 91 | |
| 8 | 127 | 122 | 123 | 124 | 126 | 31 |

10 - 19

| name | | | | | |
|------|-----|-----|-----|----|-------|
| 9 | 16 | 17 | 18 | 19 | 21 |
| 10 | 5 | 6 | | | |
| 11 | 26 | 27 | 36 | | |
| 12 | 105 | 106 | 107 | | |
| 13 | 96 | 93 | 111 | | |
| 14 | 81 | 28 | 29 | | |
| 15 | 82 | 20 | 142 | | |
| 16 | 22 | 23 | | | |
| 17 | 68 | 51 | 52 | 53 | 54 55 |

20 - 27

| name | | | |
|------|-----|-----|-----------|
| 18 | 128 | 129 | 66 |
| 19 | 144 | 41 | 146 |
| 20 | 64 | 63 | |
| 21 | 112 | 113 | 114 92 95 |
| 22 | 138 | 141 | |
| 23 | 35 | 109 | |
| 24 | 99 | 7 | |
| 25 | 155 | 157 | |
| 26 | 120 | 118 | |

img: 8



img: 9



img: 10



img: 11



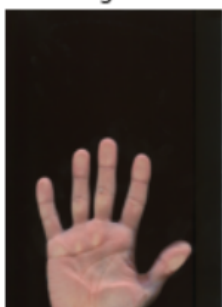
img: 76



img: 77



img: 46



img: 78



img: 79



4 Итоги

Как видно из примеров, качество методов сегментации, векторизации и кластеризации очень высоко. Экспертная оценка автора совпала с результатами кластеризации практически во всех случаях. Ошибки возникают на очень схожих изображениях, поэтому для идеальной кластеризации необходимо было бы учитывать еще какие-то факторы вроде формы ладони, возможно, ее рисунка или числа морщинок. Однако и на данном этапе алгоритм можно уверенно назвать качественным.

Единственное место, в котором была задействована сеть, - это сегментация самих ладоней. Несмотря на однородность фона, методы бинаризации не приводили к успеху, так как параметр приходилось бы подбирать под каждое изображение непосредственно, так как разные люди по-разному нажимали на сканер, осветляя некоторые области ладоней или, наоборот, затемняя их. Unet же с достаточной степенью точности справилась с сегментацией, верно предсказав очертания любой ладони вне зависимости от яркости областей на изображении.

Результат работы алгоритма для ВСЕХ изображений можно найти в конце ноутбука.

5 Программная реализация

Архив содержит 4 библиотеки, 1 ноутбук и 3 сета картинок.

Библиотеки:

1. **utils** - содержит функции загрузки и предобработки изображений, классы для датасетов нейросети
2. **net** - содержит реализацию сети UNet
3. **netprocess** - содержит функции процессов обучения и постобработки результатов сегментации
4. **showing** - содержит функции визуализации результатов работы функций из предыдущих библиотек

Сеты картинок:

1. **training** - содержит исходные изображения
2. **train-img** - содержит изображения из тренировочного датасета
3. **train-mask** - содержит маски изображений из тренировочного датасета

6 Выводы

В данном отчете представлены результаты лабораторной работы № 2 для уровня сложности **Expert**. Обработка производилась с помощью как нейронных, так и чисто математических методов. Было задействовано множество функций библиотек `skimage` и `pytorch`, а так же функций собственного производства. Учитывая многогранность и сложность подхода, качество результата и отчета, искренне надеюсь на высокую оценку проверяющих.