

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Направление: 02.03.02 «Фундаментальная информатика и информационные технологии»
ООП: Программирование и информационные технологии

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Предмет: прикладные задачи построения современных вычислительных систем

Тема задания: создание программы для моделирования систем распознавания по лицам

Выполнил: Докиенко Денис Александрович 17.Б-13-пу
Фамилия И. О. номер группы

Санкт-Петербург
2021

Содержание

Содержание	2
Цель работы	3
Разработанная программа	3
Признаки и параметры	5
Кросс-валидация	12
Голосование	15
Выводы	17
Ссылки	18

Цель работы

Целью данной работы является изучение признаков, используемых для описания изображений с лицами, написание программного решения для классификации лиц на основе этих признаков, проведение сравнительного анализа результатов классификации.

Разработанная программа

Для выполнения практической работы была разработана программа со следующим интерфейсом (Рис. 1):

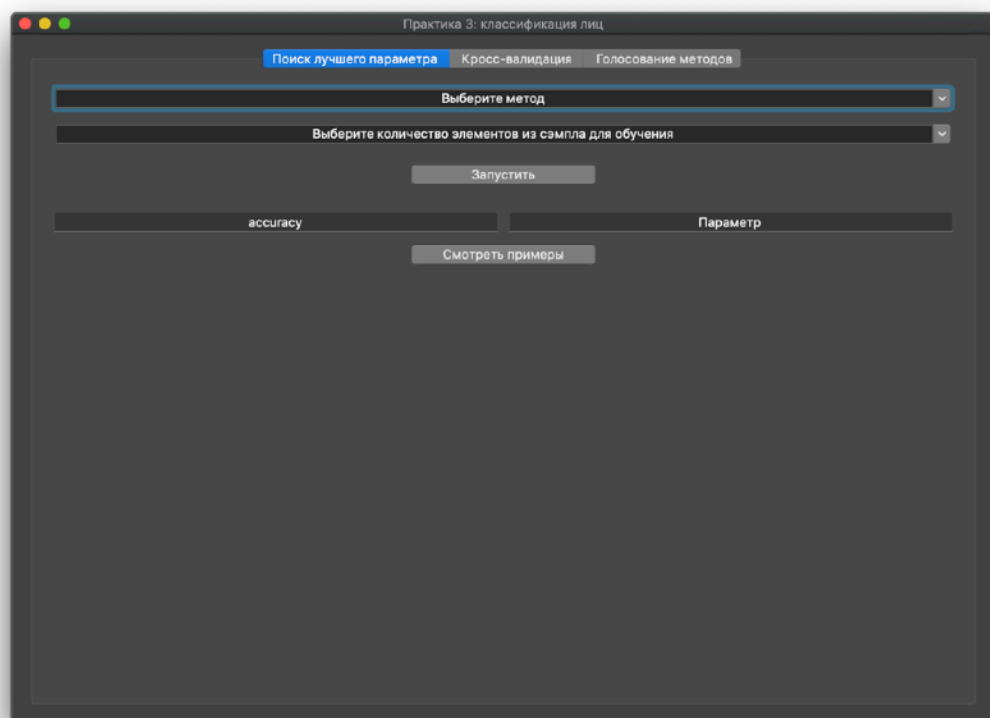


Рис. 1 (а) — Поиск лучшего параметра и вывод примеров

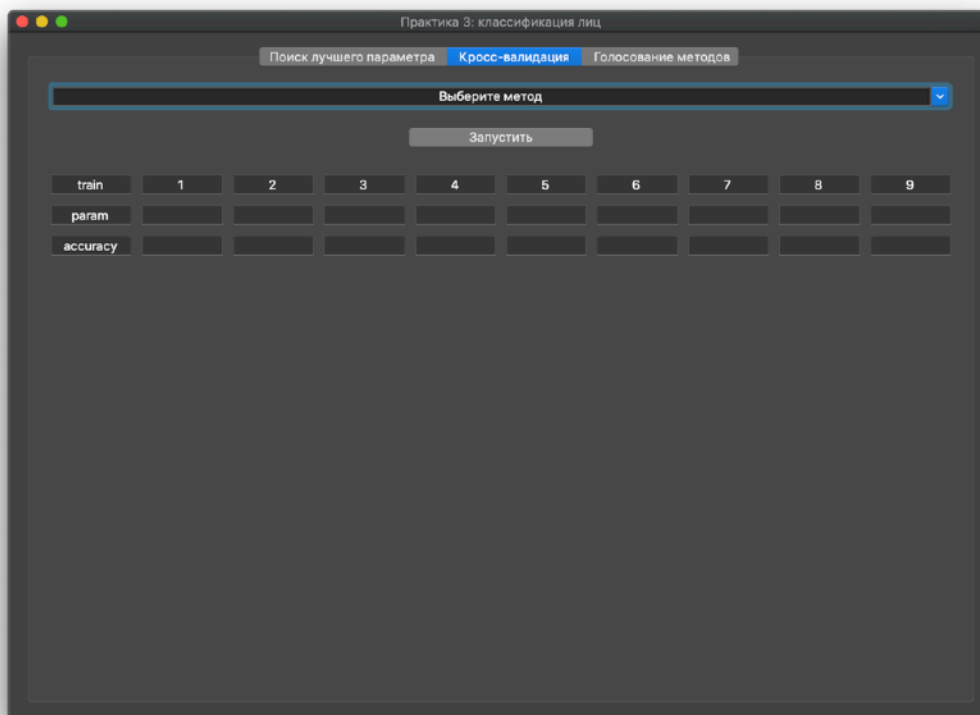


Рис. 1 (b) — Кросс-валидация

Интерфейс состоит из трех разделов — в данной практической работе нас интересуют только первые два, последний понадобится для четвертой практической работы. Графический интерфейс разработан с использованием фреймворка PyQt5, бэкенд разработан на Python 3.9. Ссылка на код в репозитории GitHub приведена в соответствующем разделе в конце работы [1].

Для работы был выбран открытый датасет, содержащийся в sklearn - «The ORL Database of Faces» (fetch_olivetti_faces). В нем содержится 40 классов, в каждом классе — по 10 изображений одного человека, отличающиеся поворотом лица и эмоциями. Фотографии сделаны при достаточном освещении, выровнены и нормализованы (яркости пикселей находятся в пределах $[0, 1]$). Фотографии имеют размер 64x64. Подробнее можно узнать из официальной документации sklearn. [2]

В данной работе нас интересуют три функциональности программы:

- поиск лучшего параметра для признака по введенному размеру training-датасета,
- вывод примеров лиц по заданному параметру и признаку,
- кросс-валидация по размеру training-датасета при выбранном признаке.

Под размером training-датасета в данном случае подразумевается такое количество элементов (от 1 до 9) одного класса, которое будет использоваться для подбора параметров признака.

Про параметры и признаки поговорим подробнее далее.

Признаки и параметры

1. Гистограмма

Данный метод на основе полученного изображения вычисляет вектор, состоящий из количества пикселей изображения в соответствующем интервале яркости.

Параметром данного признака является количество таких интервалов.

Приведем несколько примеров изображений с различными параметрами (Рис. 2-4):

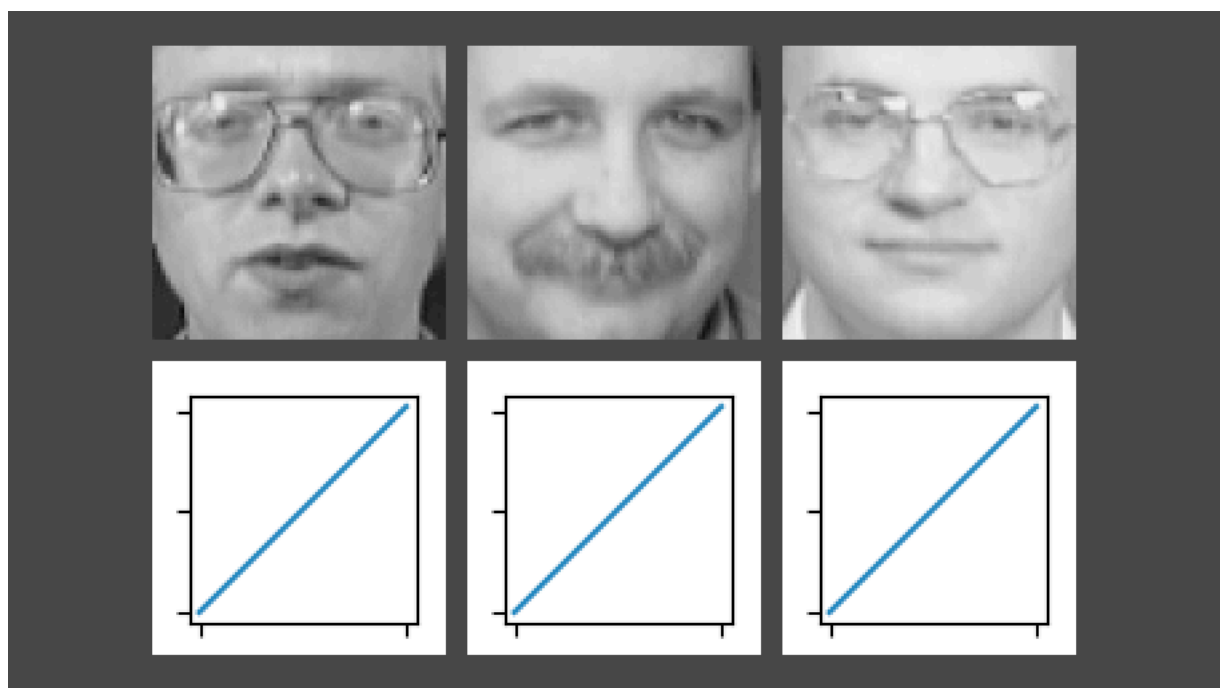


Рис. 2 — Два интервала

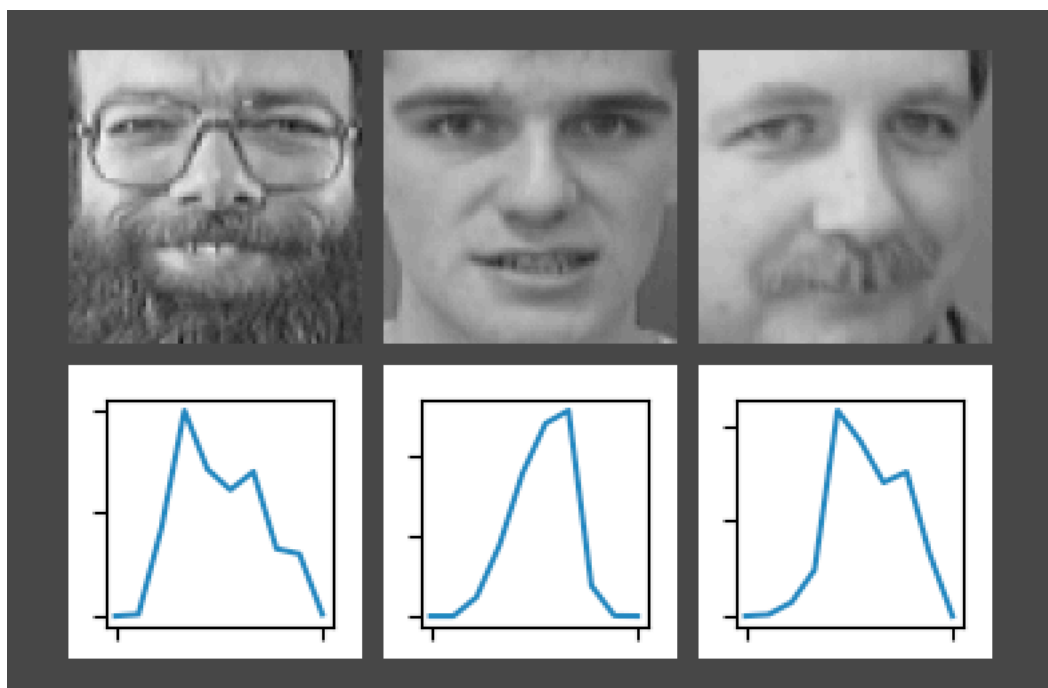


Рис. 3— Десять интервалов

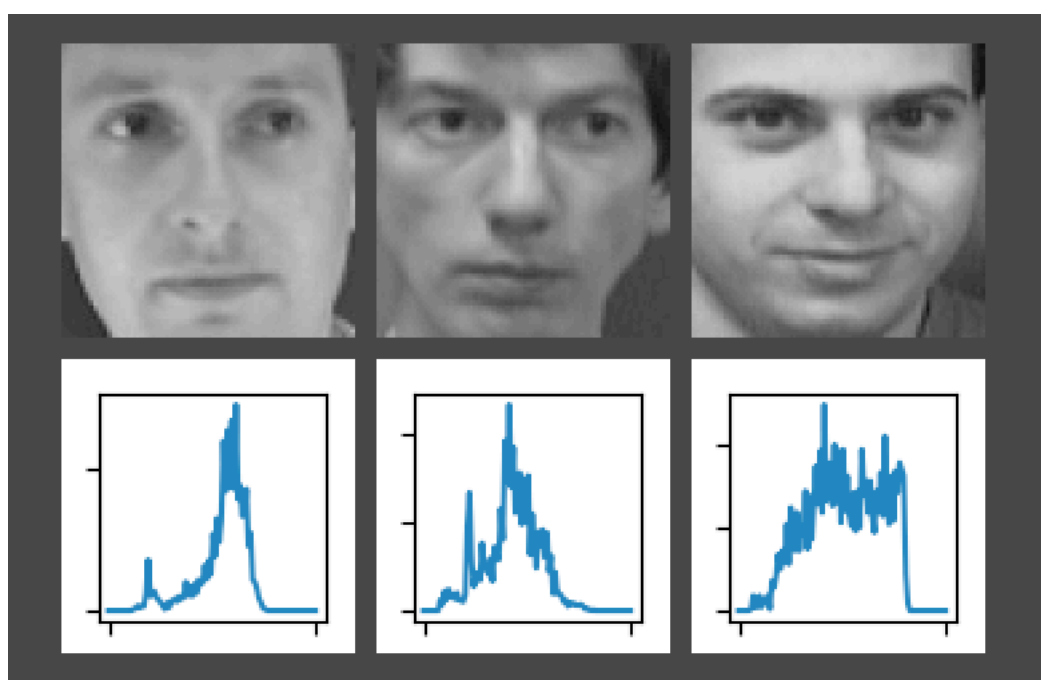


Рис. 4 — Сто интервалов

Как видно, гистограмма для разных людей даже на малых значениях параметра выявляет основные тенденции изображения. Это может значить то, что сильное увеличение параметра не будет сильно увеличивать качество классификации.

2. DFT

Данный признак заключается в выполнении преобразования Фурье над введенным изображением. При этом оставляется левая подматрица, размерность которой и является перебираемым параметром.

Приведем несколько примеров изображений с различными параметрами (Рис. 5-7):

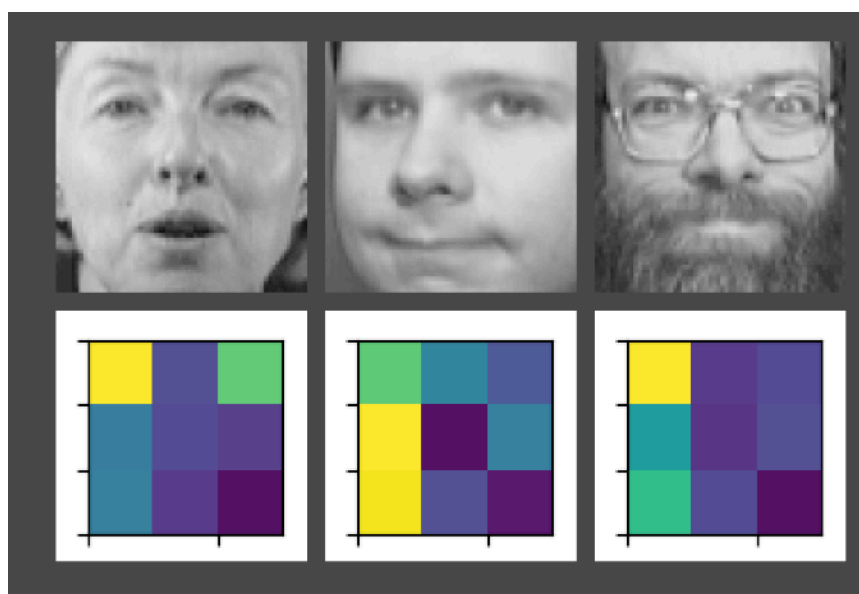


Рис. 5 — Подматрица размера 4

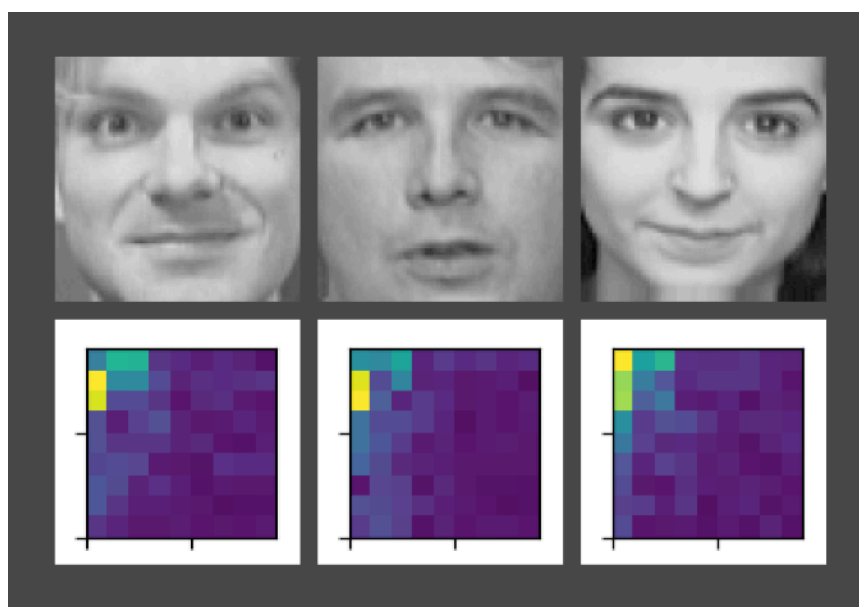


Рис. 6 — Подматрица размера 10

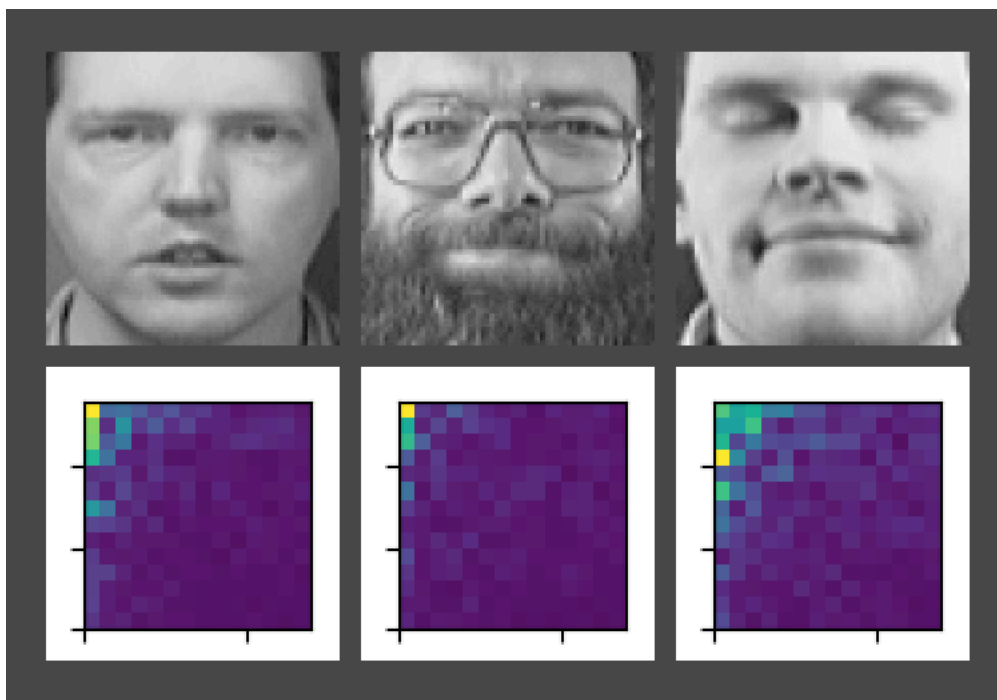


Рис. 7 — Подматрица размера 15

Как видно из примеров, оптимальное значение параметра будет не сильно большое (максимум, вероятно, — 30 или 40), поскольку при увеличении размера рассматриваемой матрицы новые пиксели содержат все меньше информации.

3.DCT

DCT — дискретное косинусное преобразование. Это преобразование тесно связано с дискретным преобразованием Фурье и является гомоморфизмом его векторного пространства.

Параметр, аналогично DFT — размерность левой подматрицы

Приведем несколько примеров изображений с различными параметрами (Рис. 8-10):

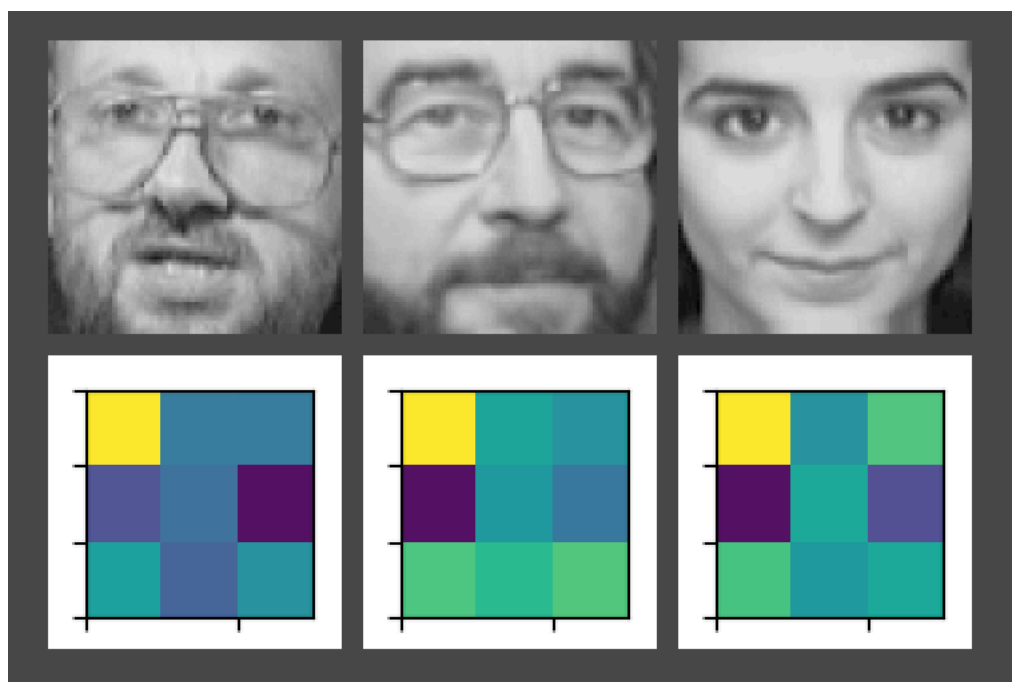


Рис. 8 — Подматрица размера 4

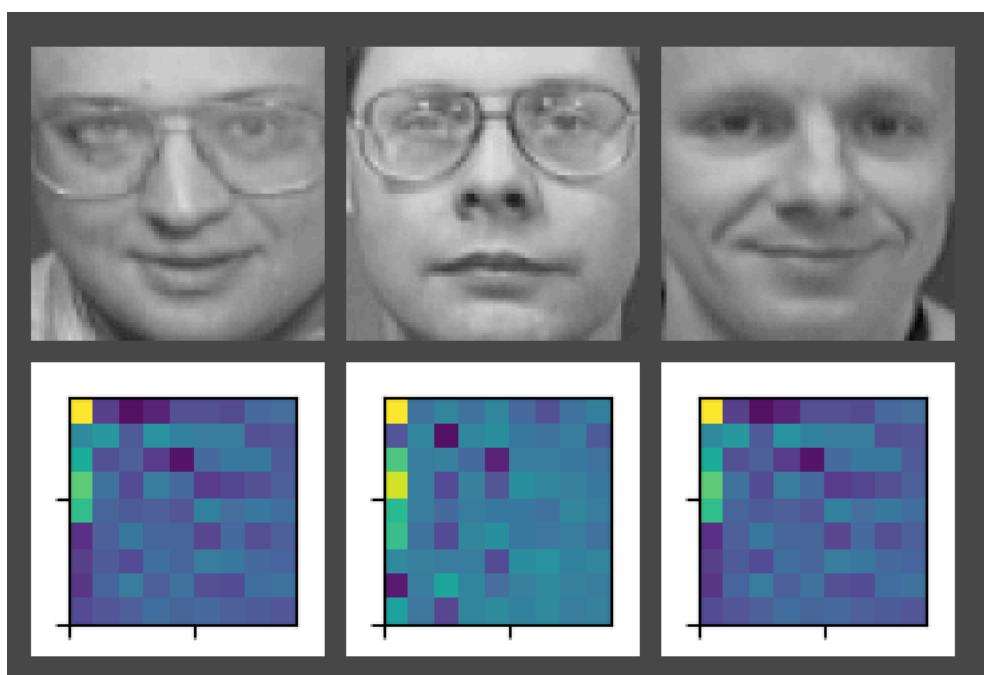


Рис. 9 — Подматрица размера 10

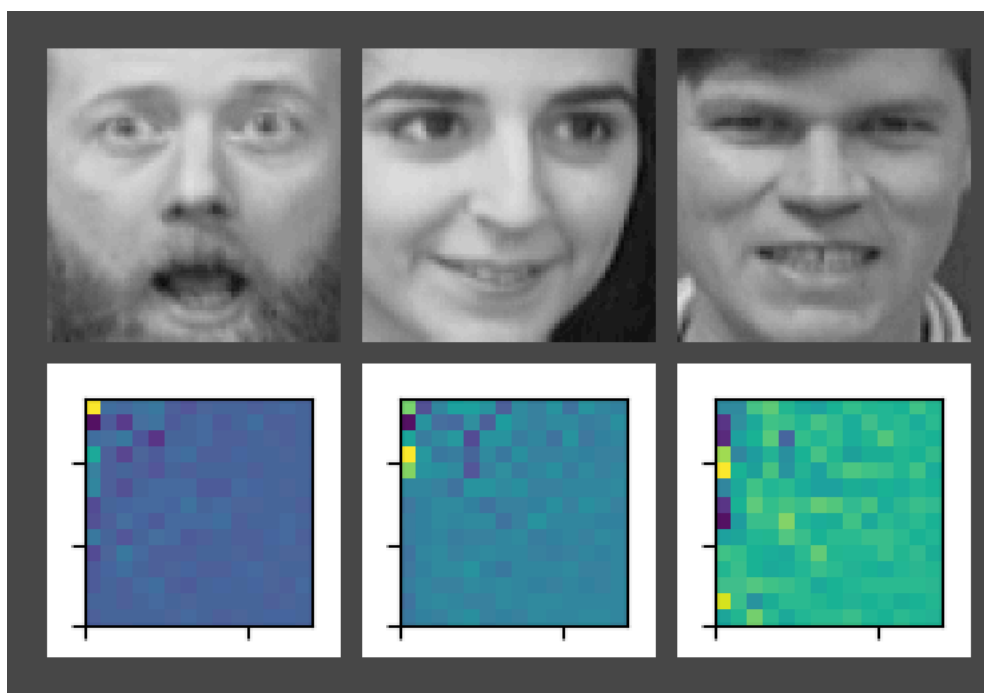


Рис. 10 — Подматрица размера 15

Аналогично DFT оптимальное значение параметра не больше, вероятно, 30 или 40.

4. Градиент

Вычисляется расстояние между двумя зеркально расположенными полосами шириной ≥ 1 пикселя, скользящими по исходному изображению лица. Расстояние на каждом этапе скольжения можно рассматривать как градиент яркости между зеркально расположенными полосами.

Параметром является размер указанного выше скользящего окна.

Приведем несколько примеров изображений с различными параметрами (Рис. 11-13):

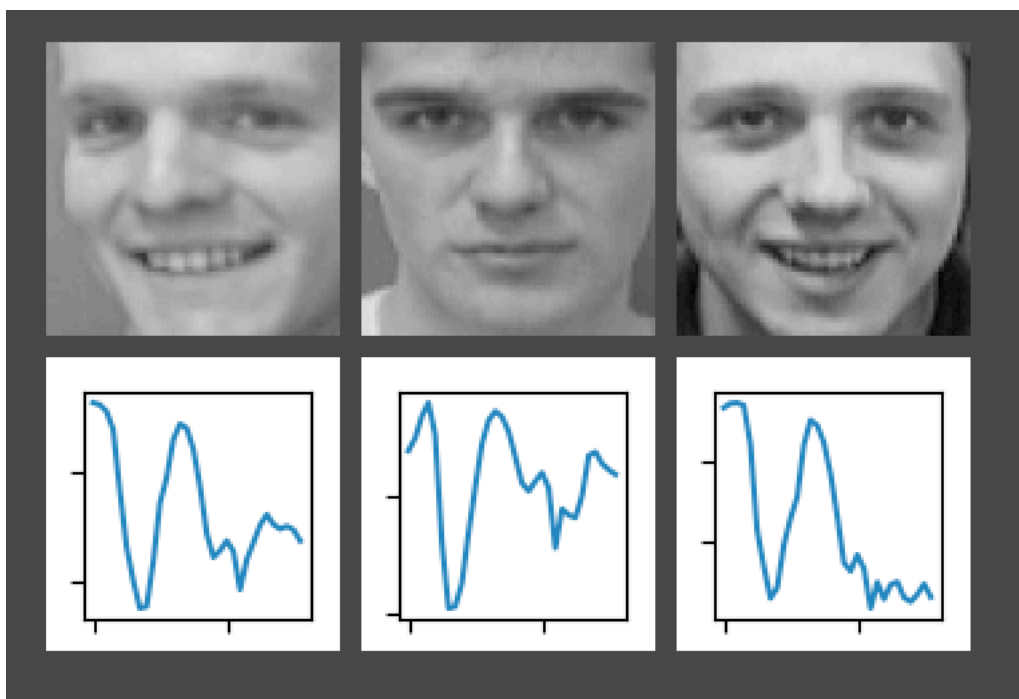


Рис. 11 — Окно размера 2

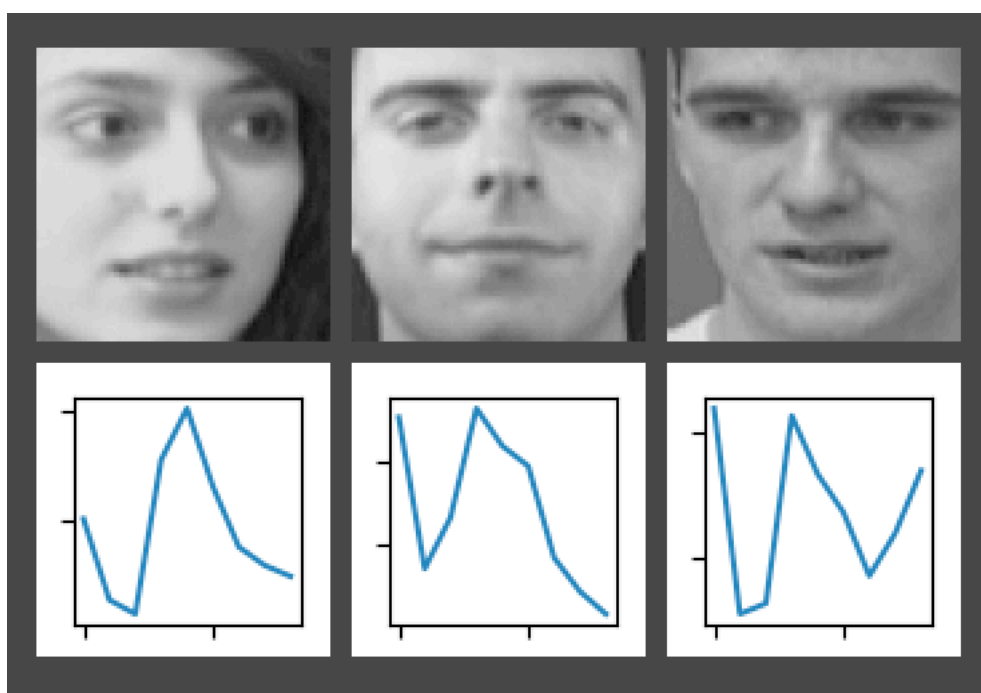


Рис. 12 — Окно размера 7

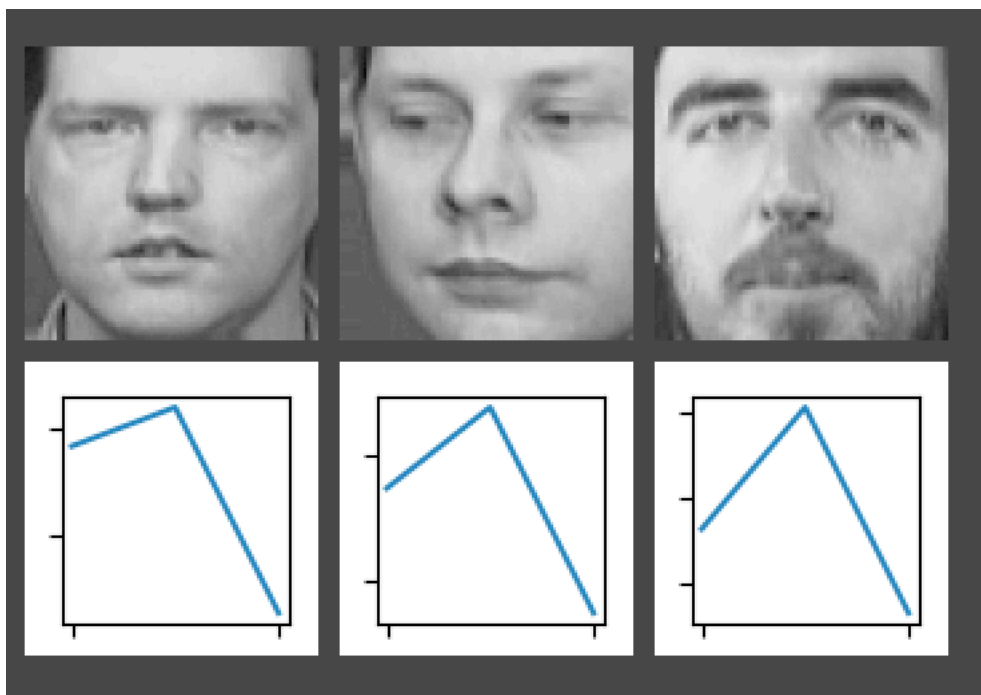


Рис. 13 — Окно размера 20

По примерам видно, что при увеличении параметра гладкость графика уменьшается, а следовательно график теряет информативность. Значит оптимальный параметр будет маленьким (до 20).

5. Scale

Данный метод преобразует исходное изображение в изображение с меньшей размерностью.

Параметром здесь является коэффициент преобразования.

Кросс-валидация

Итак, перейдем к подбору оптимальных параметров. Поскольку кросс-валидация — это фактически расширенный функционал поиска оптимального параметра для конкретного размера training-выборки, то мы можем сразу перейти к рассмотрению именно кросс-валидации.

Далее результаты будут представлены в виде скриншотов (Рис. 14-18).

1. Гистограмма

Результаты кросс-валидации представлены на следующем рисунке (Рис. 14):

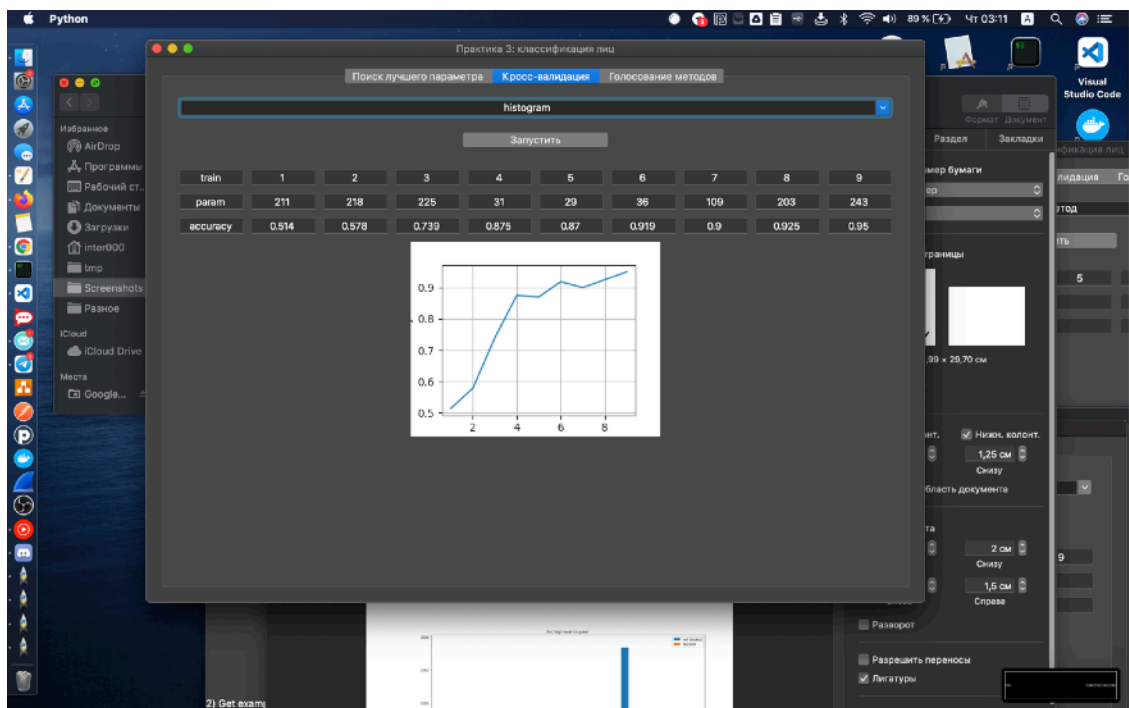


Рис. 14 — Кросс-валидация для гистограммы

2. DFT

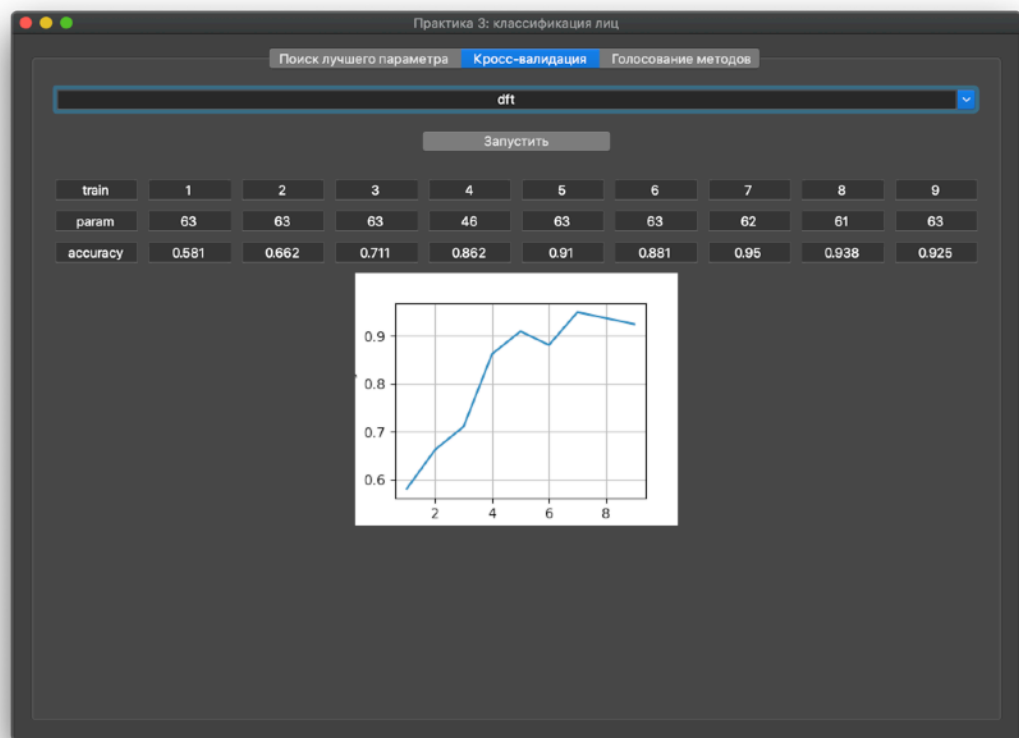


Рис. 15 — Кросс-валидация для DFT

3. DCT

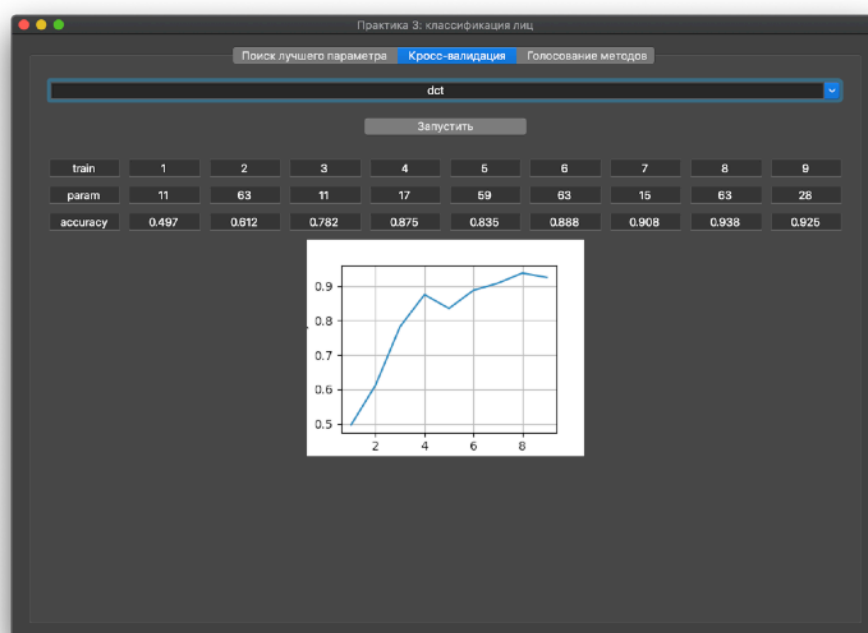


Рис. 16 — Кросс-валидация для DCT

4. Градиент

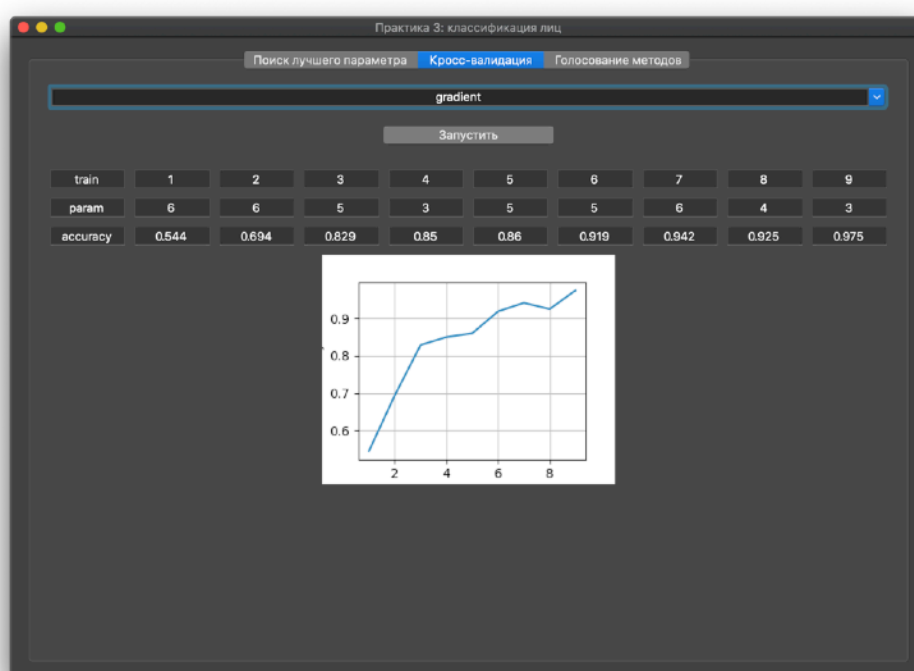


Рис. 17 — Кросс-валидация для градиента

5. Scale

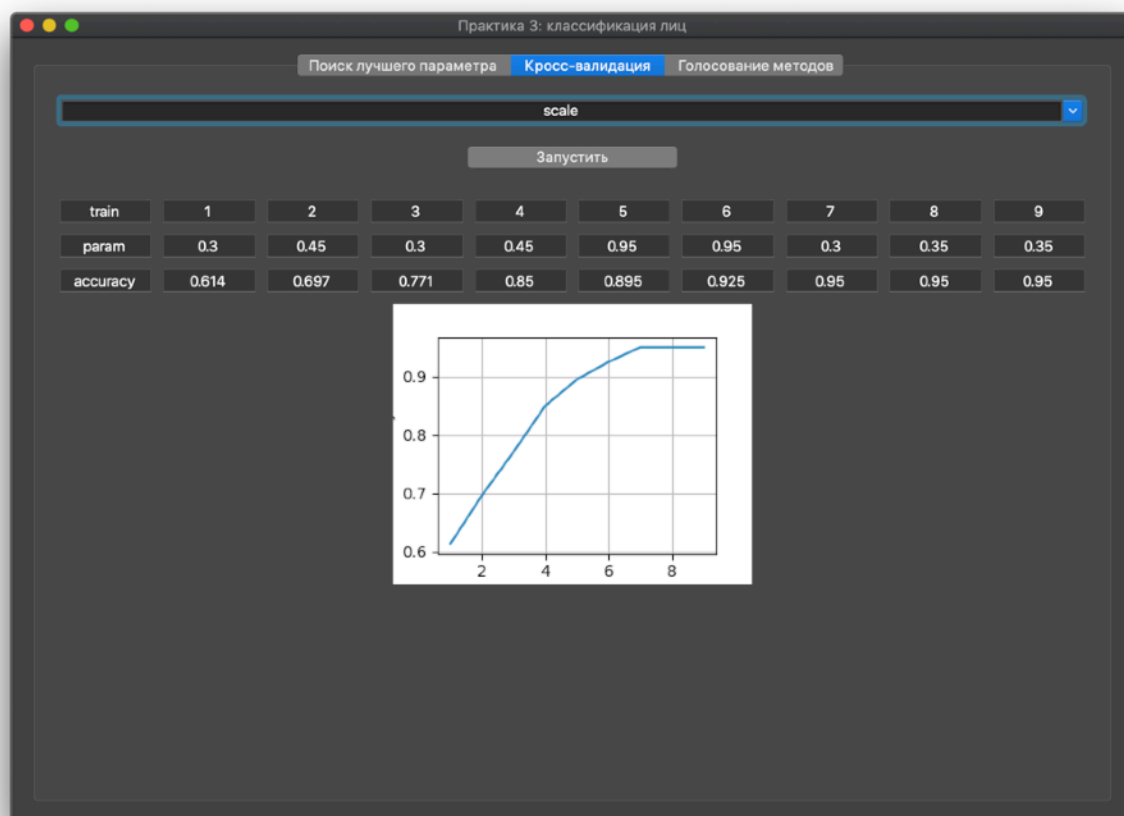


Рис. 18 — Кросс-валидация для Scale

Итак, мы рассмотрели результаты кросс-валидации и теперь можем перейти к рассмотрению голосования.

Голосование

Как можно улучшить лучшие результаты методов? Логичной мыслью будет объединить их результаты в лучший мета-алгоритм. Тогда каждый алгоритм будет голосовать за решением и, вероятно, алгоритмы будут компенсировать недостатки друг друга.

Рассмотрим, как это реализована в приложении (Рис. 19):

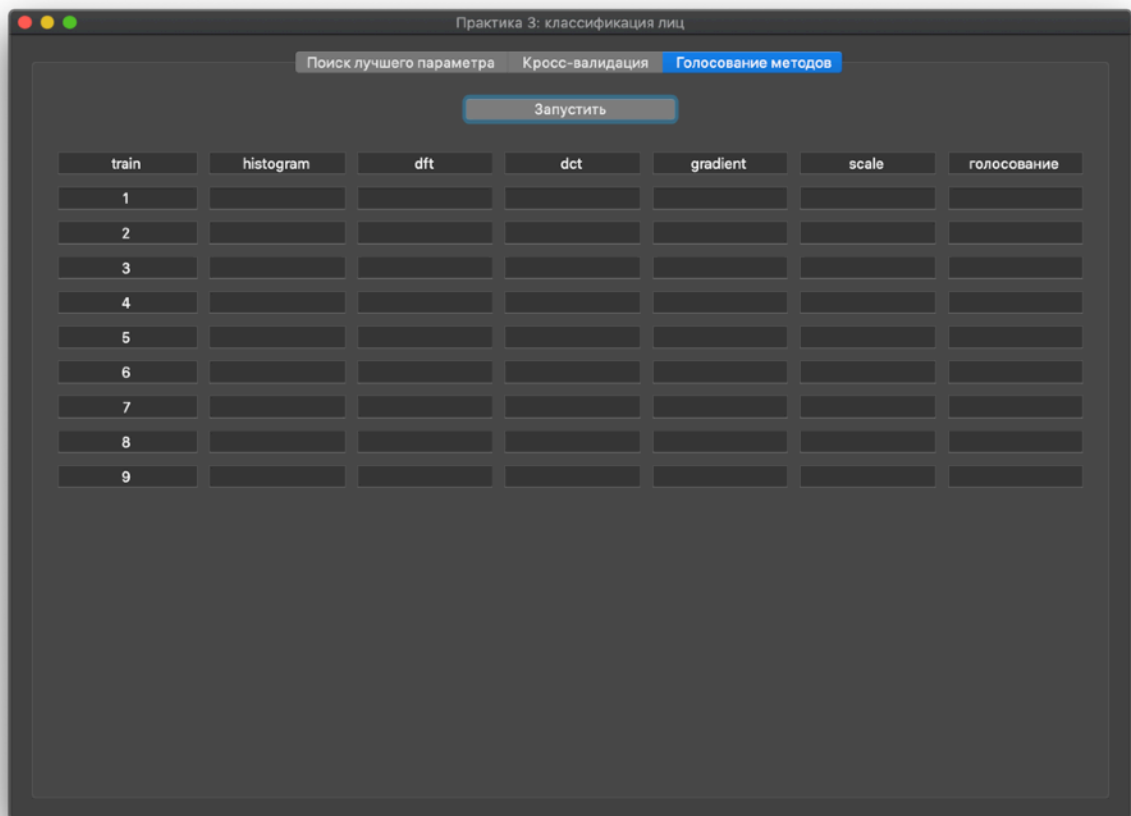


Рис. 19 — Реализация голосования алгоритмов

Если запустить ее, то для каждого значения размера train-выборки будет подобран наиболее подходящий параметр (аналогично кросс-валидации).

Тогда получатся следующие значения: (Рис. 20)

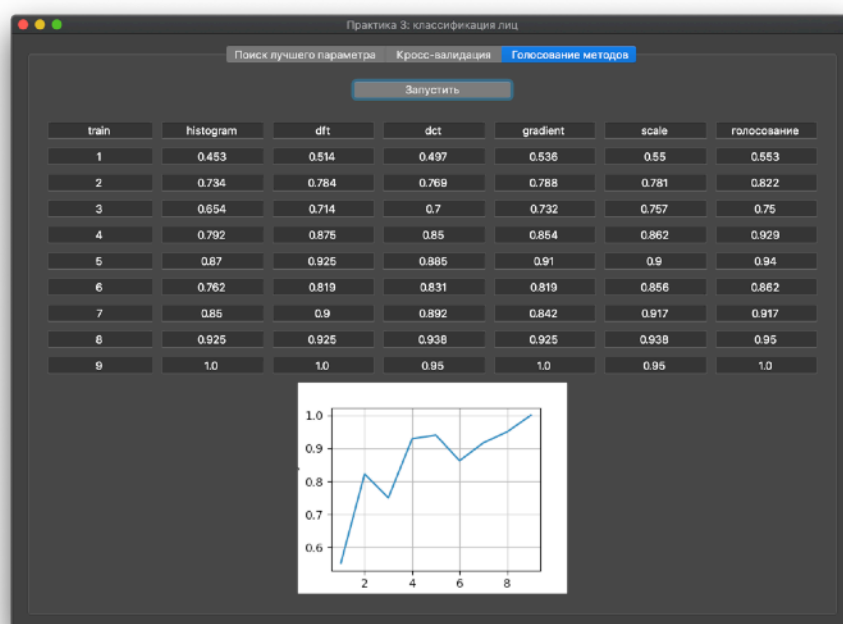


Рис. 20 — Результаты голосования методов

Выводы

Итак, мы получили итоговую точность для разных объёмов train-выборок при голосовании. И действительно в среднем намного эффективней использовать именно подход голосования. Конечно, скорость вычислений при этом сильно падает, но результаты как правило того стоят

Ссылки

[1] Исходный код в открытом доступе [Электронный ресурс] // URL:
<https://github.com/sncodeGit/faces/tree/main/2>

[2] Документация sklearn с информацией об используемом датасете
[Электронный ресурс] // URL: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_olivetti_faces.html