

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ
УПРАВЛЕНИЯ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
на тему «Методы распознавания лиц»

Выполнила
студентка 2 курса
группы 21-Б15.ПУ

Павлова Ксения Андреевна

Преподаватель
Щеголева Надежда Львовна

Санкт-Петербург
2022

СОДЕРЖАНИЕ

Цель.....	3
Задачи.....	3
Ход работы	
Описание методов экстракции признаков	4
Блок-схема общего алгоритма.....	5
Примечания к блок-схеме общего алгоритма.....	6
Описание функций алгоритма.....	6
Сравнение методов.....	8
Контрольный пример	12
Заключение	14
Список литературы	16

Цель: изучить различные методы экстракции признаков (ЭП).

Задачи:

1. Изучить принцип работы методов ЭП;
2. Реализовать эти методы с помощью языка программирования Python;
3. Выяснить, как влияет на работу алгоритмов изменение определенных параметров;
4. Сравнить различные методы ЭП.

Ход работы.

Описание методов экстракции признаков.

I. Метод гистограмм.

Вычисляется гистограмма яркости исходного изображения, где по абсциссе располагаются диапазоны яркости пикселей, а по ординате количество пикселей. Таким образом, гистограмма иллюстрирует, сколько пикселей соответствует тому или иному диапазону. Значения гистограммы сводятся в соответствующий вектор признаков.

II. Метод случайных точек.

Случайным образом выбираются отдельные пиксели исходного изображения, равномерно распределенные по нему, и сводятся в вектор яркостных признаков.

III. Метод сжатия.

Изображение «сжимается» в n раз: для каждого квадрата из n пикселей рассчитывается среднее значение яркостей этих пикселей. Средние значения квадратов сводятся в соответствующий вектор признаков.

Блок-схема общего алгоритма.

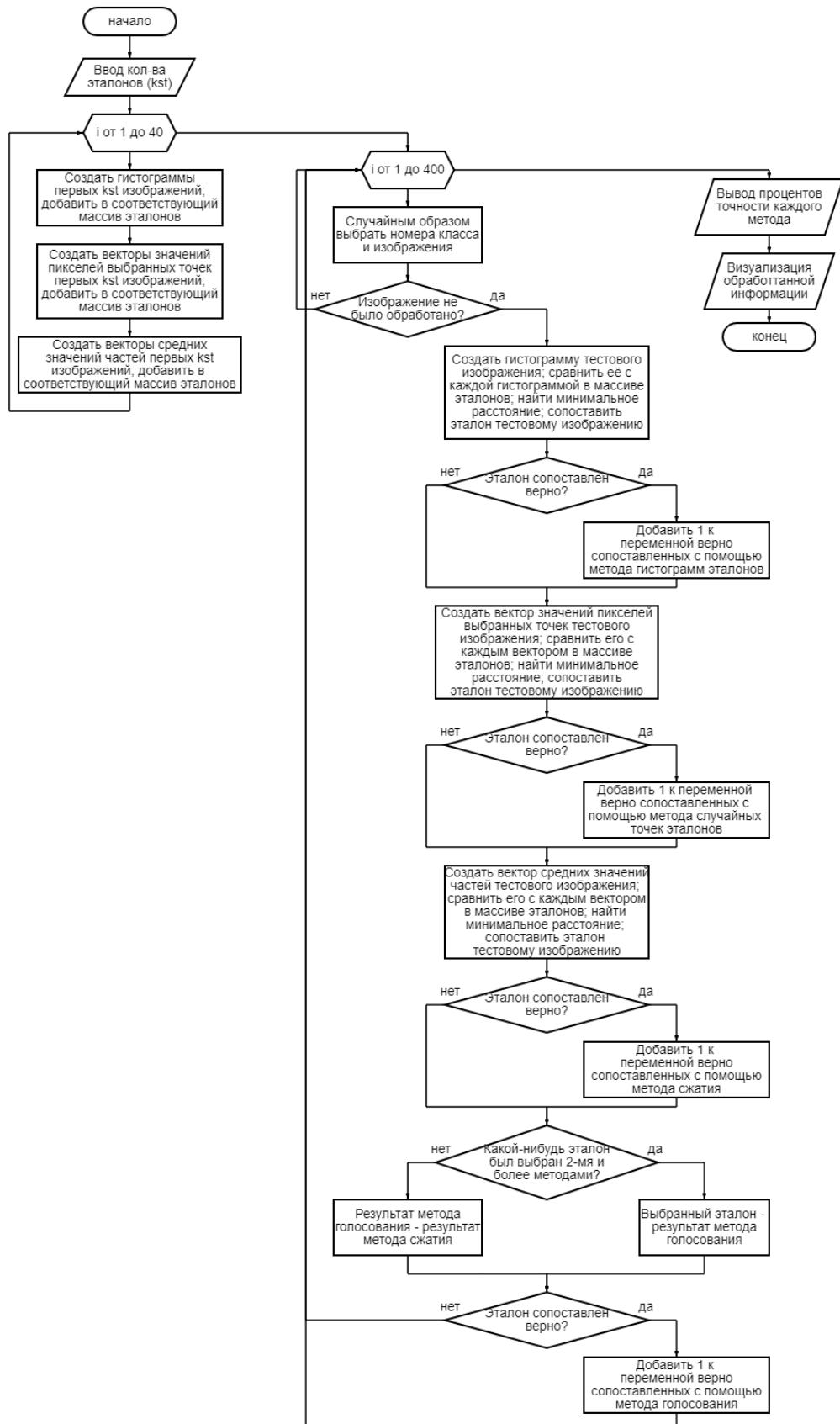


Рис.1. Блок-схема общего алгоритма.

Примечания к блок-схеме общего алгоритма (Рис. 1).

Массив случайно выбранных координат пикселей с равномерным распределением по изображению был построен с помощью стандартной библиотеки «*trandom*» до исполнения основной программы, чтобы уменьшить время выполнения алгоритма.

Результатом метода голосования становится эталон, выбранный хотя бы двумя простыми методами. Если все результаты различны, голосованием выбирается результат, полученный методом сжатия, так как этот метод является наиболее точным.

Процент точности каждого метода рассчитывается как частное правильно выбранных данным методом эталонов и количества обработанных изображений, умноженное на 100, и выводится с точностью до двух знаков после запятой.

Функции получения гистограмм, векторов яркостей пикселей с случайными координатами и векторов средних значений яркостей пикселей в квадратах изображения, а также функции сравнения и визуализации будут описаны далее.

Описание функций.

I. Вычисление гистограмм.

Данная функция реализована с помощью библиотек OpenCV и *pymtrr*, так как последняя позволяет строить гистограммы с помощью встроенной функции, причём количество столбцов в гистограмме можно изменять, а первая позволяет открыть изображение в нужном формате. На вход данной функции подаются два числа – номер класса и номер изображения. На выходе функции возвращается гистограмма в виде вектора, элементами которого являются значения столбцов.

II. Получение векторов яркости пикселей случайных точек.

Данная функция реализована с помощью компонента библиотеки PIL – *Image*, который позволяет получить значение яркости пикселей с определёнными координатами. На вход данной функции подаются два числа

– номер класса и номер изображения. Используя глобальную переменную – массив, случайно выбранных координат пикселей, функция создаёт вектор значений яркости пикселей в этих точках. На выходе возвращается полученный вектор.

III. Получение векторов средних значений яркости пикселей в квадратах.

Данная функция реализована с помощью компонента библиотеки PIL – Image, который позволяет получить значение яркости пикселей с определёнными координатами. На вход данной функции подаются два числа – номер класса и номер изображения. В функции изображение делится на m квадратов, а затем для каждого квадрата вычисляется среднее значение яркости пикселей в нём, которое записывается в вектор. На выходе возвращается полученный вектор.

IV. Сравнение.

На вход данной функции подаются 2 вектора одинаковой длины. Для каждой пары элементов с одинаковыми индексами вычисляется модуль разности, который прибавляется к общей сумме расстояния между векторами. На выходе функции возвращается конечная общая сумма – расстояние, которое далее записывается в соответствующий определённому методу массив. Из этого массива выбирается минимальное расстояние для сопоставления тестового изображения эталону.

V. Визуализация.

Данный блок программы реализован с помощью библиотек OpenCV, PIL, matplotlib.pyplot, numpy и tkinter. Первые две используются для открытия изображений в нужных форматах и считывания с них определённой информации (размер изображения, яркость пикселей и т.п.), а также для преобразования изображений (для метода случайных точек и метода сжатия). Библиотеки matplotlib.pyplot и numpy позволяют визуализировать метод гистограмм, а библиотека tkinter – выводить группу изображений (тестовое,

искомый эталон, эталоны, найденные каждым из методов и векторы признаков) для каждого из тестов в одном окне.

Сравнение методов.

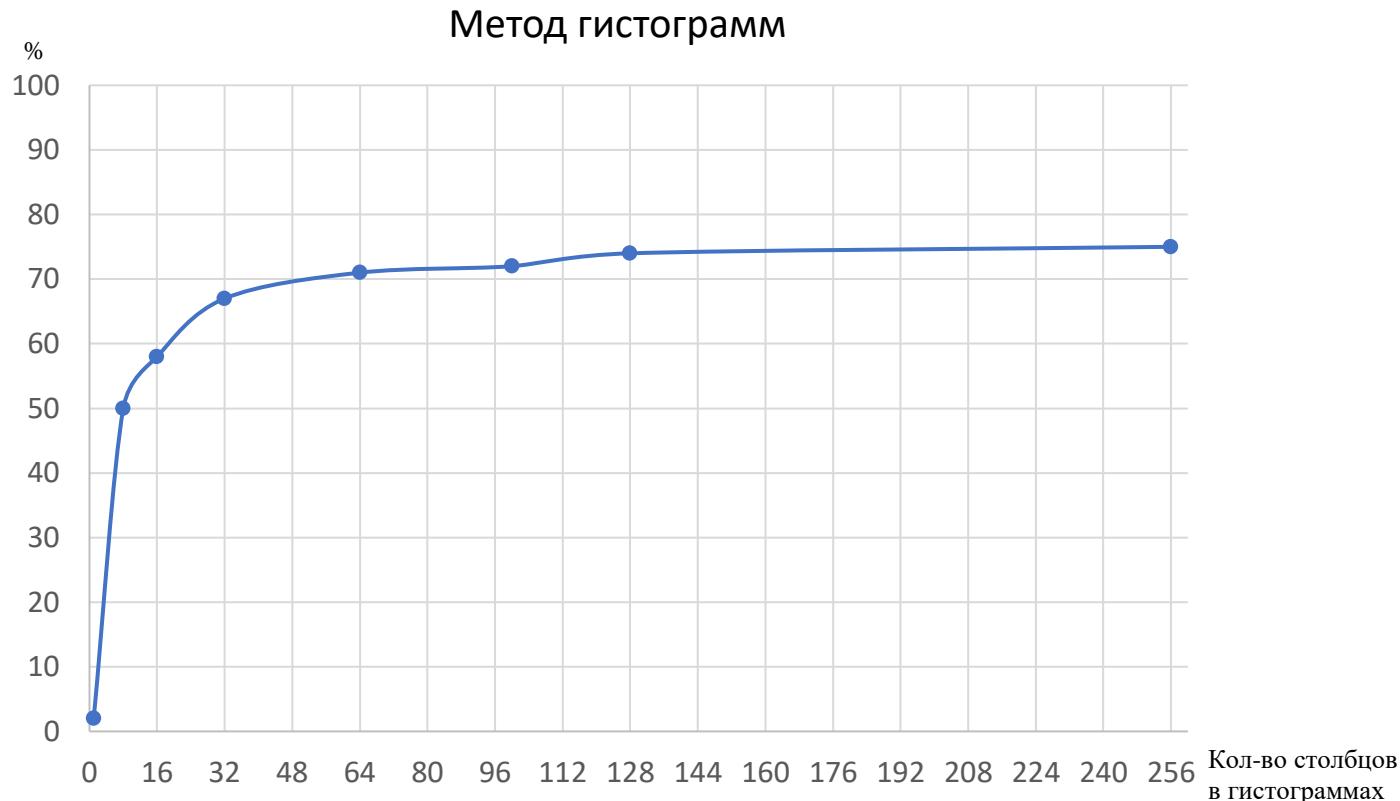


Рис.2. Точность метода гистограмм в зависимости от количества столбцов.

При увеличении количества столбцов в гистограммах и константных значениях других параметров (количество эталонов – по три в каждом классе) точность метода гистограмм сначала резко возрастает, а затем увеличивается менее, чем на 1% (рис. 2), однако при этом время выполнения программы увеличивается более, чем в 2 раза, поэтому выбрано 64 столбца в гистограммах изображения.

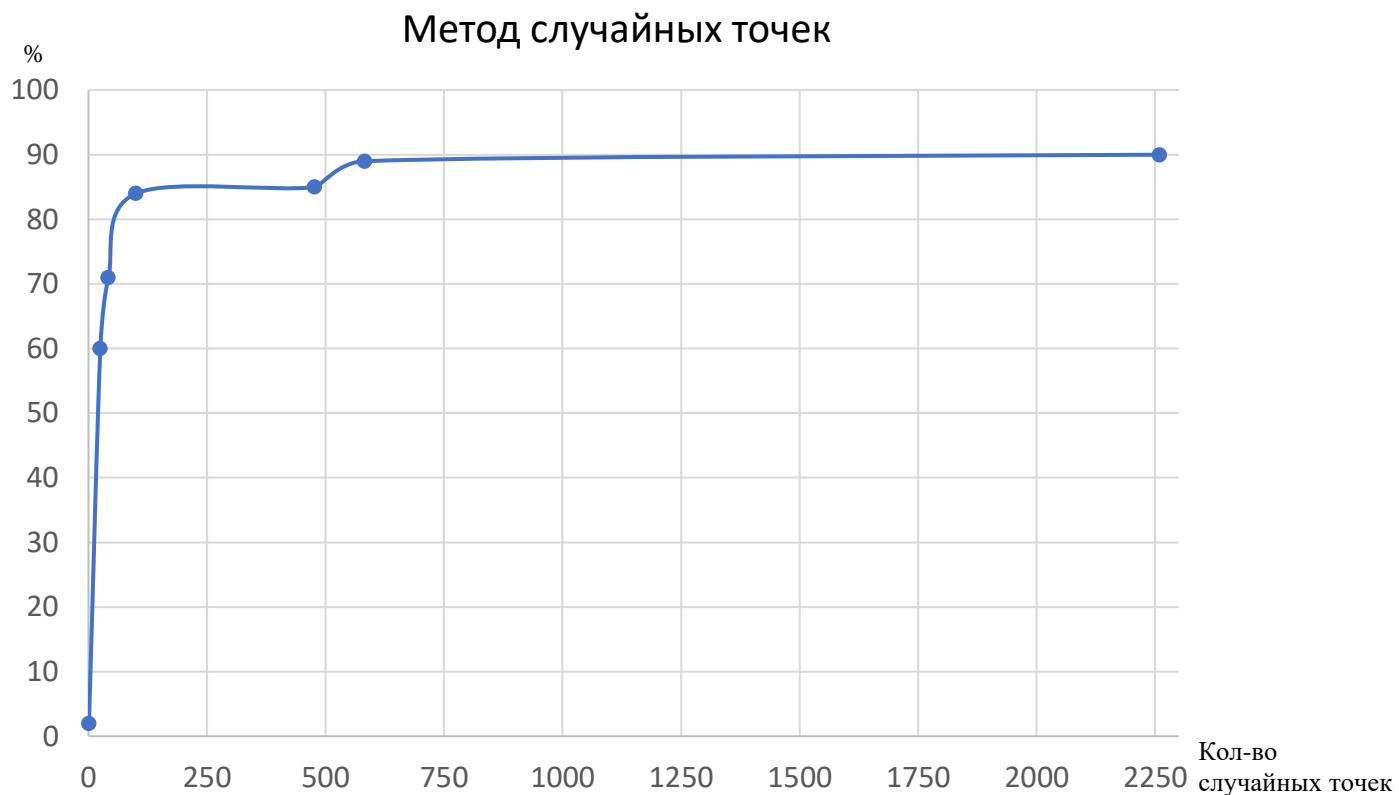


Рис.3. Точность метода случайных точек в зависимости от их количества.

При увеличении количества случайных точек и константных значениях других параметров (количество эталонов – по три в каждом классе) точность метода случайных точек сначала резко возрастает, а затем увеличивается менее, чем на 1% (рис. 3), однако при этом время выполнения программы увеличивается в 3 раза, поэтому выбирается по 580 случайных пар координат точек в изображениях для составления вектора признаков.

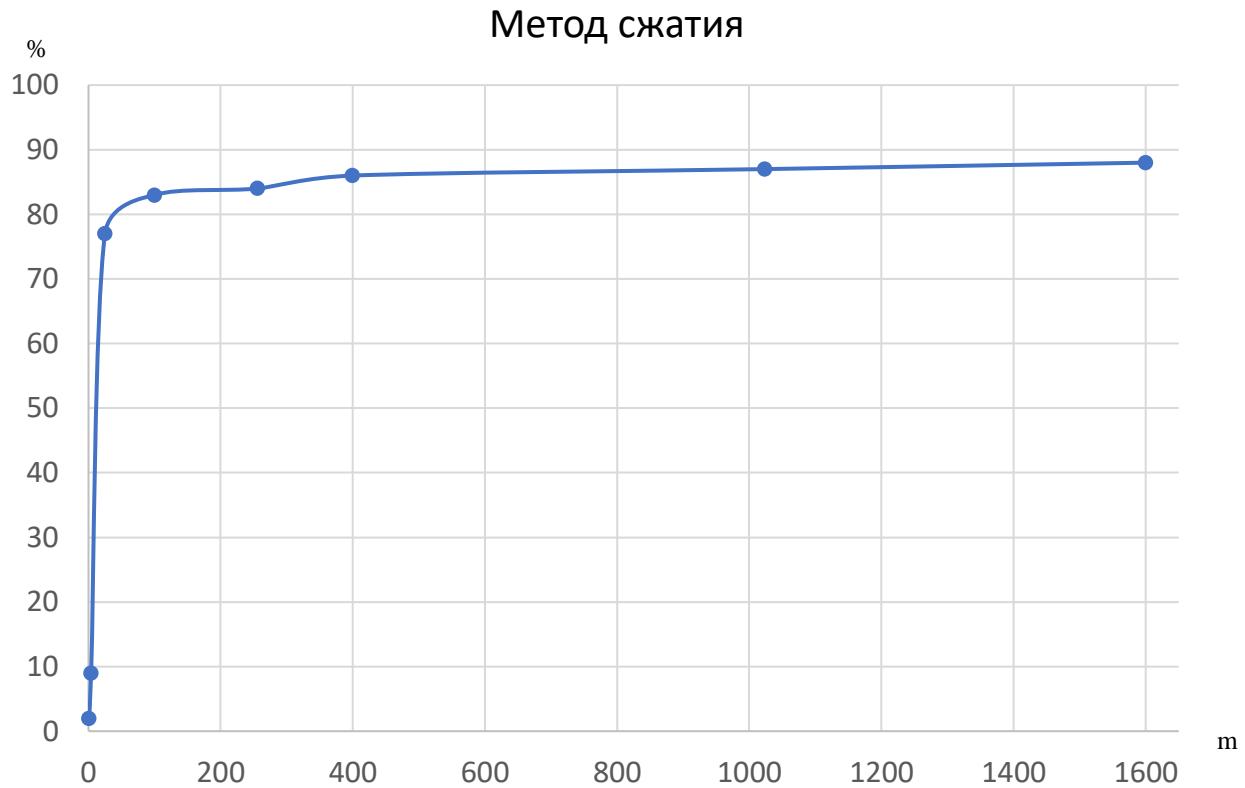


Рис.4. Точность метода сжатия в зависимости от m .

При увеличении m (количество квадратов, на которые делится изображение) и константных значениях других параметров (количество эталонов – по три в каждом классе) точность метода сжатия сначала резко возрастает, а затем увеличивается менее, чем на 2% (рис. 4), однако при этом время выполнения программы увеличивается в 3,5 раза, поэтому выбирается деление на 400 квадратов.

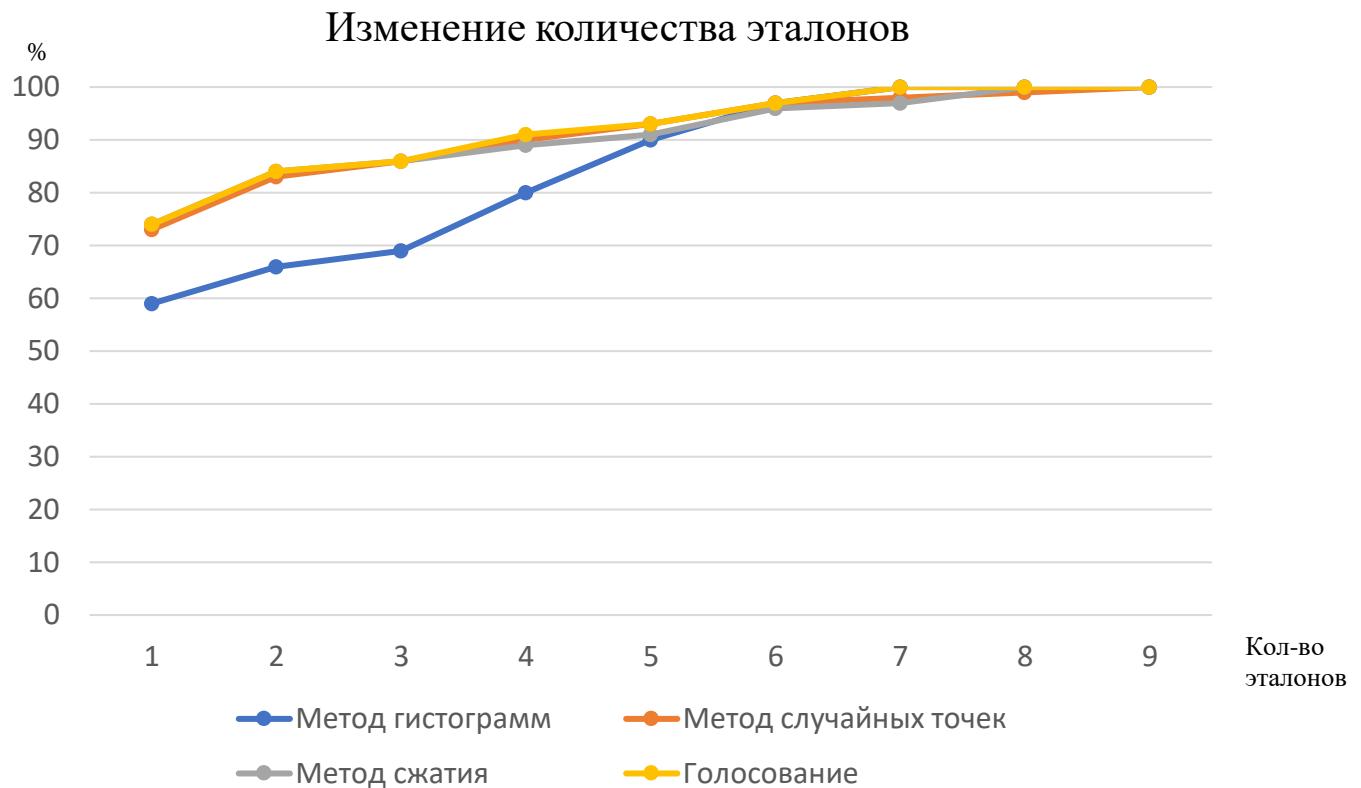


Рис.5. Точность методов в зависимости от количества эталонов.

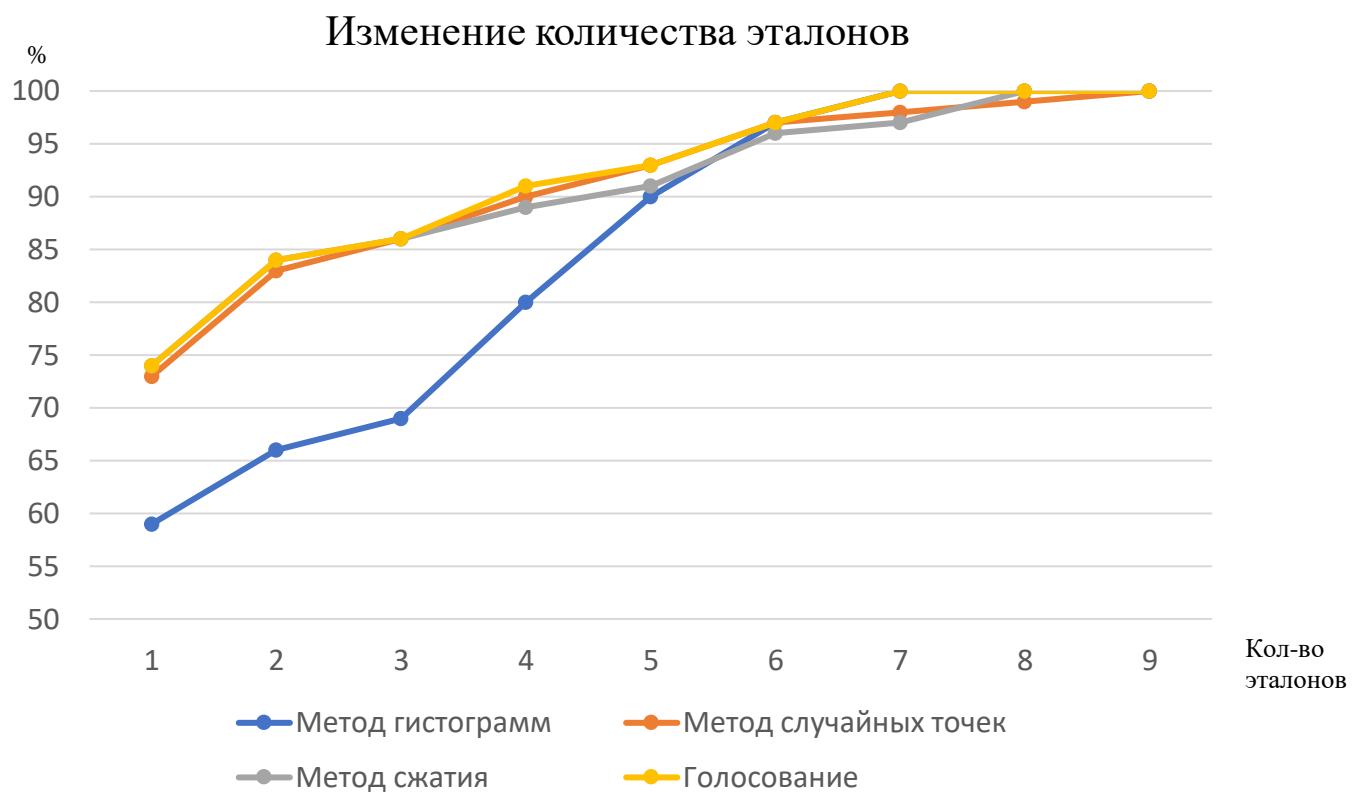


Рис.6. Точность методов в зависимости от количества эталонов
(увеличенный масштаб).

При увеличении количества эталонов в каждом классе и константных значениях параметров различных методов точность алгоритма увеличивается в соответствии с представленными графиками (рис. 5 и рис. 6). Это связано с тем, что на большем количестве эталонов человек находится в большем разнообразии положений, а значит найти близкое к тестовому изображению легче.

Контрольный пример.

В данном разделе представлен пример работы данной программы.

Количество эталонов в каждом классе равно 4, количество столбцов в методе диаграмм – 64, количество случайных неповторяющихся точек – 580, количество квадратов – 400.

Количество запусков – 400, но обработано 314 тестовых изображений, так как при новом запуске проверяется не было ли изображение обработано раньше, в противном случае алгоритм переходит к следующему запуску.

В основном алгоритм выдаёт верный результат (рис. 7). Возникающие ошибки различных методов чаще всего объясняются тем, что изображения действительно похожи: либо на них изображены люди со схожими чертами лица, либо освещение в соответствующих участках изображений совпадает.

Вывод результата работы алгоритма – рис. 8.



Рис. 7. Визуализация обработанных данных.

The screenshot shows the PyCharm IDE interface. The project tree on the left lists files like faces, gist, mash, rand_points, venv library root, empty.png, hist.py, main.py, point.py, and touch.txt. The main editor window displays the main.py script with code for calculating accuracy of different methods. The Run tab at the bottom shows the command run and the output terminal window displaying the execution results.

```
result_voting = result_compression
answer.append(result_voting)
if result_voting + 1 == folder:
    correct_voting += 1
answers.append(answer)

percent_hist = correct_hist / (len(used) - 40 * kst) * 100
percent_points = correct_points / (len(used) - 40 * kst) * 100
percent_compression = correct_compression / (len(used) - 40 * kst) * 100
percent_voting = correct_voting / (len(used) - 40 * kst) * 100
print("Точность метода гистограмм = {percent_hist:.2f} %")
print("Точность метода случайных точек = {percent_points:.2f} %")
print("Точность метода сжатия = {percent_compression:.2f} %")
print("Точность метода голосования = {percent_voting:.2f} %")
print("Обработано изображений: ", len(used))

for i in range(400):
    if str(folder) + " " + str(fil... : else
```

```
C:\Users\user\PycharmProjects\face_recognition\venv\Scripts\python.exe C:/Users/user/PycharmProjects/face_recognition/main.py
Введите количество эталонов в каждом классе:
|
Точность метода гистограмм = 76.62 %
Точность метода случайных точек = 87.66 %
Точность метода сжатия = 88.96 %
Точность метода голосования = 88.96 %
Обработано изображений: 314

Process finished with exit code 0
```

Рис. 8. Вывод результата работы

Заключение.

В ходе работы изучены и реализованы несколько методов экстракции признаков полутоновых изображений лиц, а также проведен сравнительный анализ работы методов. Из полученных данных сделаны выводы:

- С увеличением количества столбцов в методе гистограмм, количества случайных неповторяющихся точек в методе случайных точек и количества квадратов в методе сжатия увеличивается и точность работы алгоритма, однако начиная с некоторого момента точность перестаёт расти значительно, а время выполнения программы увеличивается;
- Наиболее точным из реализованных методов является метод голосования, который учитывает результат работы каждого из представленных простых методов.

Список литературы.

1. Методы обработки и распознавания изображений лиц в задачах биометрии / Г. А. Кухарев, Е. И. Каменская, Ю. Н. Матвеев, Н. Л. Щеголева; под ред. М. В. Хитрова. – СПб.: Политехника, 2013. – 388 с.: ил.
2. Introducing Python. Modern computing in simple packages / Bill Lobarovic.
3. <https://dmkpress.com/files/PDF/978-5-97060-200-3.pdf>
4. Ссылка на код https://github.com/ksenkap/face_recognition