

LBPFaceRecognizer (на основе функций LBPН).

Теперь, когда мы знаем немного больше о распознавании лиц и LBPН, давайте пойдем дальше и посмотрим шаги алгоритма:

1. **параметры:** LBPН использует 4 параметра:

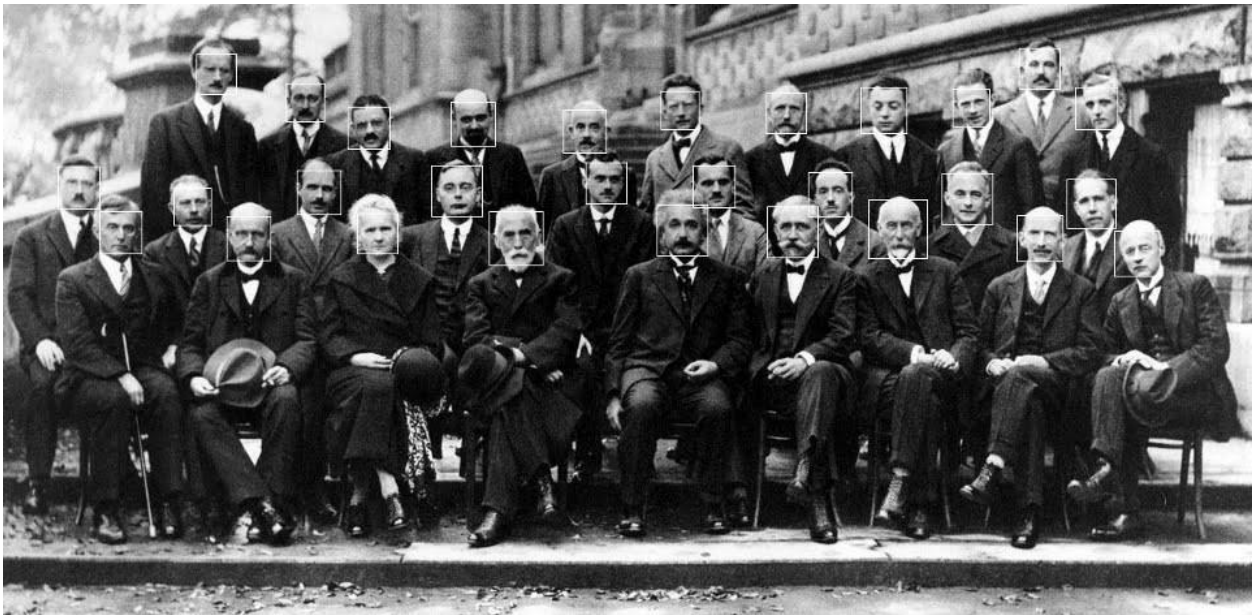
- **Радиус:** радиус используется для построения кругового локального двоичного шаблона и представляет радиус вокруг центрального пикселя. Обычно он равен 1.
- **соседи:** количество точек выборки для построения кругового локального двоичного шаблона. Помните: чем больше точек выборки вы включите, тем выше вычислительные затраты. Обычно он установлен на 8.
- **Сетка X:** количество клеток в горизонтальном направлении. Чем больше ячеек, тем мельче сетка, тем выше размерность результирующего вектора признаков. Обычно он установлен на 8.
- **Сетка Y:** количество клеток в вертикальном направлении. Чем больше ячеек, тем мельче сетка, тем выше размерность результирующего вектора признаков. Обычно он установлен на 8.

Не беспокойтесь о параметрах прямо сейчас, вы поймете их после прочтения следующих шагов.

**2.Обучение Алгоритму**Во-первых, нам нужно обучить алгоритм. Для этого нам нужно использовать набор данных с изображениями лиц людей, которых мы хотим узнать. Нам также необходимо установить идентификатор (это может быть номер или имя человека) для каждого изображения, поэтому алгоритм будет использовать эту информацию для распознавания входного изображения и выдачи вам выходных данных. Изображения одного и того же человека должны иметь одинаковый идентификатор После того, как обучающий набор уже построен, давайте посмотрим на вычислительные шаги LBPН.

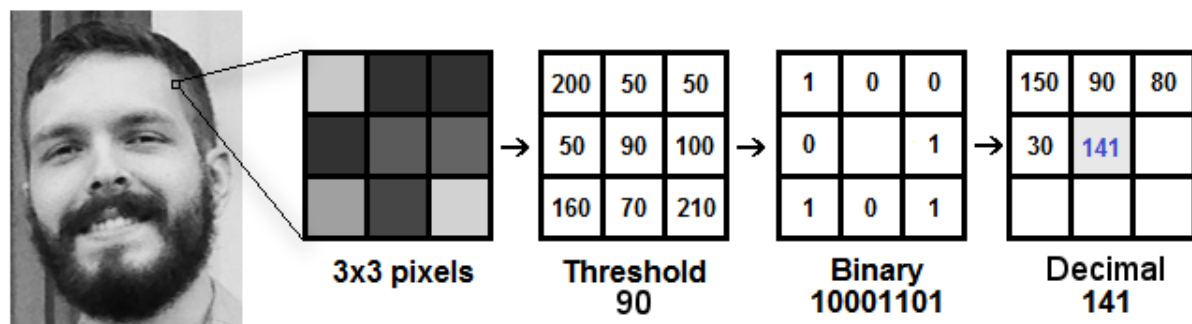
**3. Применение операции LBP:** Первый вычислительный шаг LBPН состоит в создании промежуточного изображения, которое лучше описывает исходное изображение, выделяя черты лица. Для этого алгоритм использует концепцию скользящего окна, основанную на параметрах **радиуса** также **соседи**,

Изображение ниже показывает эту процедуру:



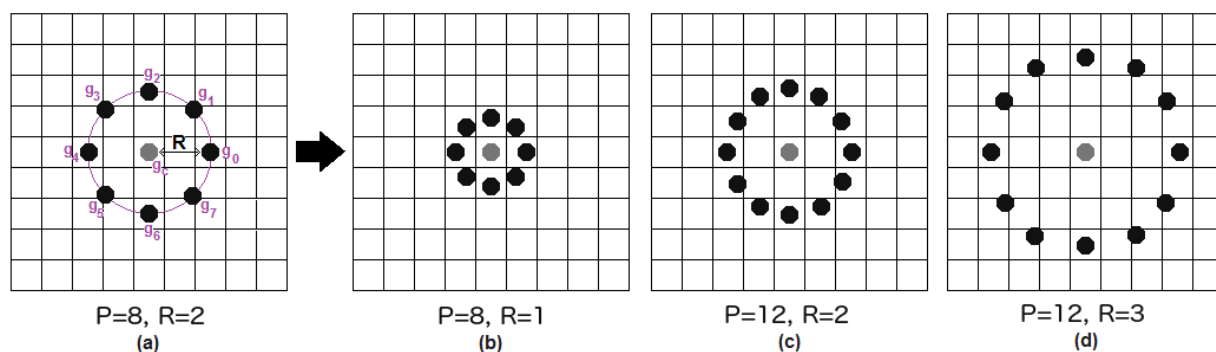
Основываясь на изображении выше, давайте разберем его на несколько небольших шагов, чтобы мы могли легко понять его:

- Предположим, у нас есть изображение лица в оттенках серого.
- Мы можем получить часть этого изображения в виде окна размером  $3 \times 3$  пикселя.
- Он также может быть представлен в виде матрицы  $3 \times 3$ , содержащей интенсивность каждого пикселя ( $0 \sim 255$ ).
- Затем нам нужно взять центральное значение матрицы, которая будет использоваться в качестве порога.
- Это значение будет использоваться для определения новых значений из 8 соседей.
- Для каждого соседа центрального значения (порога) мы устанавливаем новое двоичное значение. Мы устанавливаем 1 для значений, равных или превышающих порог, и 0 для значений ниже порогового.
- Теперь матрица будет содержать только двоичные значения (без учета центрального значения). Нам нужно объединить каждое двоичное значение из каждой позиции из матрицы строка за строкой в новое двоичное значение (например, 10001101). Примечание: некоторые авторы используют другие подходы для объединения двоичных значений (например, по часовой стрелке), но конечный результат будет таким же.
- Затем мы преобразовываем это двоичное значение в десятичное значение и устанавливаем его в центральное значение матрицы, которое фактически является пикселем от исходного изображения.
- В конце этой процедуры (процедура LBP) у нас есть новое изображение, которое лучше отражает характеристики исходного изображения.
- **Заметка:** Процедура LBP была расширена для использования другого числа радиусов и соседей, она называется Circular LBP.



Это можно сделать с помощью **билинейная интерполяция**, Если некоторая точка данных находится между пикселями, она использует значения из 4 ближайших пикселей (2x2), чтобы оценить значение новой точки данных.

**4. Извлечение гистограмм** Теперь, используя изображение, созданное на последнем шаге, мы можем использовать **Сетка Ха** также **Сетка Y** Параметры для разделения изображения на несколько сеток, как видно на следующем изображении:



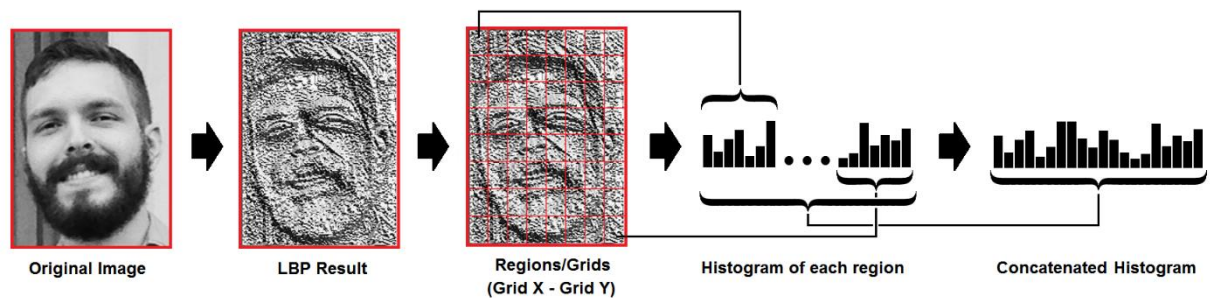
Основываясь на изображении выше, мы можем извлечь гистограмму каждого региона следующим образом:

- Поскольку у нас есть изображение в оттенках серого, каждая гистограмма (из каждой сетки) будет содержать только 256 позиций (0 ~ 255), представляющих вхождения каждой интенсивности пикселей.
- Затем нам нужно объединить каждую гистограмму, чтобы создать новую и большую гистограмму. Предположим, что у нас есть сетки 8x8, мы будем иметь  $8 \times 8 \times 256 = 16,384$  позиции в финальной гистограмме. Итоговая гистограмма представляет характеристики изображения исходного изображения.

Алгоритм LBPn в значительной степени это.

**5. Выполнение распознавания лица** На этом этапе алгоритм уже обучен. Каждая созданная гистограмма используется для представления каждого изображения из набора обучающих данных. Итак, с учетом входного изображения, мы снова выполняем шаги для этого нового изображения и создаем гистограмму, которая представляет изображение.

- Таким образом, чтобы найти изображение, которое соответствует входному изображению, нам просто нужно сравнить две гистограммы и вернуть изображение с ближайшей гистограммой.
- Мы можем использовать различные подходы для сравнения гистограмм (рассчитать расстояние между двумя гистограммами), например: **Евклидово расстояние, хи-квадрат, абсолютная величина** и т. д. В этом примере мы можем использовать евклидово расстояние (которое довольно известно) на основе следующей формулы:



- Таким образом, вывод алгоритма - это идентификатор изображения с ближайшей гистограммой. Алгоритм также должен возвращать рассчитанное расстояние, которое можно использовать как **уверенность**. Измерение **Заметка**: не обманывайте себя по поводу названия «достоверность», так как более низкая достоверность лучше, потому что это означает, что расстояние между двумя гистограммами ближе.
- Затем мы можем использовать порог и «достоверность», чтобы автоматически оценить, правильно ли алгоритм распознал изображение. Можно предположить, что алгоритм успешно распознается, если достоверность ниже заданного порогового значения.

## Выводы

- LBPН - один из самых простых алгоритмов распознавания лиц.
- Он может представлять локальные особенности на изображениях.
- Можно получить отличные результаты (в основном в контролируемой среде).
- Он устойчив к монотонным преобразованиям серой шкалы.
- Это обеспечивается [OpenCV](#) библиотекой (библиотека с открытым исходным кодом Computer Vision).