

# Содержание

<b>1</b>	<b>Метод главных компонент</b>	<b>3</b>
1.1	Определения . . . . .	3
1.2	Алгоритм метода главных компонент . . . . .	3
1.3	Применение метода главных компонент для распознавания лиц	5
<b>2</b>	<b>Метод опорных векторов</b>	<b>7</b>
<b>3</b>	<b>Алгоритм Виолы-Джонса</b>	<b>8</b>
<b>4</b>	<b>Метод гибкого сравнения на графах</b>	<b>14</b>

# 1 Метод главных компонент

## 1.1 Определения

*Собственный вектор матрицы* — ненулевой вектор, умножение квадратной матрицы на который дает тот же вектор с числовым коэффициентом, называемым собственным значением [?].

*Ковариационная матрица вектора* — квадратная симметрическая матрица, элементами которой являются ковариации — меры связанности всех возможных пар компонент вектора. Ковариация имеет отрицательное значение, если большие значения одной компоненты соответствуют меньшим значениям другой, и наоборот. Ковариация имеет положительное значение, если большие значения одной компоненты соответствуют большим значениям другой, и аналогично для меньших значений. Ковариация компоненты вектора самой с собой равна дисперсии [?]. На (1.1) представлен пример ковариационной матрицы для трех компонент, ковариация обозначена как  $\text{Cov}(X_1, X_2)$ , дисперсия обозначена как  $\text{Var}(X_1, X_2)$ .

$$\Sigma = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \text{Cov}(X_1, X_3) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \text{Cov}(X_2, X_3) \\ \text{Cov}(X_3, X_1) & \text{Cov}(X_3, X_2) & \text{Var}(X_3) \end{bmatrix} \quad (1.1)$$

## 1.2 Алгоритм метода главных компонент

Метод главных компонент — один из статистических методов уменьшения размерности данных, т. е. числа параметров, которые заданы переменными и необходимы для описания объекта. Такие методы по возможности сохраняют наибольшее количество информации и общую структуру данных [?, ?]. Далее будет изложен алгоритм уменьшения размерности из  $n$  в  $k$ :

- 1) метод главных компонент очень чувствителен к статистическим выбросам и неточностям в данных, поэтому перед применением необходима стандартизация данных — приведение данных к единому мас-

штабу [?, ?]. Это можно сделать вычитанием среднего из каждой переменной и делением полученного на стандартное отклонение [?]:

$$Z_i = \frac{X_i - \bar{X}_i}{\sigma_{X_i}}. \quad (1.2)$$

- 2) для уменьшения размерности данных необходимо найти переменные, которые сильно зависят от значений друг друга, т. е. скоррелированные переменные. Это важно, поскольку такие переменные могут содержать излишнюю информацию, от которой можно было бы избавиться с минимальными потерями полезной информации. Чтобы понять, какие переменные скоррелированы, необходимо найти ковариационную матрицу;
- 3) далее необходимо найти главные компоненты — новые переменные, которые получаются после введения новой системы координат таким образом, что каждая следующая главная компонента максимизирует дисперсию данных относительно себя, но не коррелирует с предыдущими главными компонентами, т. е. перпендикулярна им. Таким образом, из  $n$  изначальных переменных получится  $n$  главных компонент, однако последним главным компонентам будет соответствовать наименьшая дисперсия, т. е. наименьшая изменчивость — это означает, что эти главные компоненты содержат наименьшее количество информации и могут быть отброшены с минимальными потерями. Математически направления осей главных компонент — это собственные векторы ковариационной матрицы, а собственные значения — это коэффициенты, отражающие дисперсию, соответствующую этой главной компоненте. Таким образом, полученные собственные значения необходимо упорядочить по убыванию и выбрать  $k$  первых собственных векторов, которые и будут новыми переменными для уменьшения размерности [?, ?, ?];

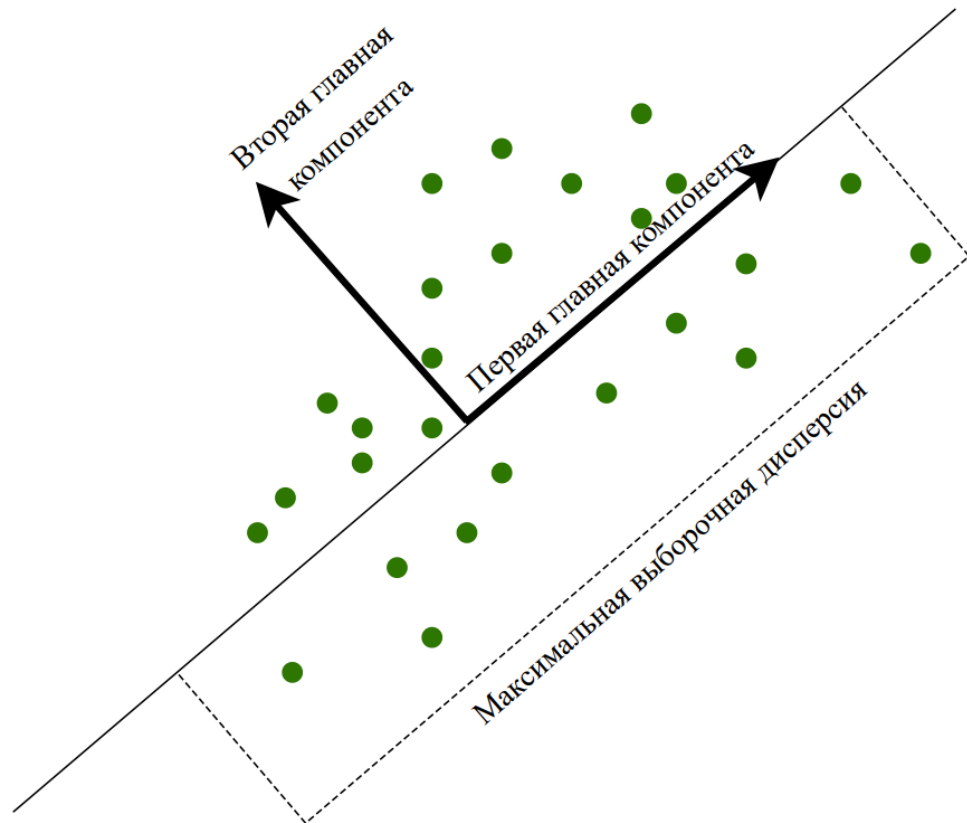


Рисунок 1.1 – Пример ввода первой и второй главных компонент

- 4) На последнем шаге необходимо преобразовать данные, перейдя от исходных переменных к полученным  $k$  главным компонентам. Для этого необходимо умножить транспонированную матрицу исходных данных  $D$  на транспонированную матрицу главных компонент  $PC$  [?].

$$D^T \cdot PC^T \quad (1.3)$$

### 1.3 Применение метода главных компонент для распознавания лиц

Для задачи распознавания человека по изображению лица метод можно применить следующим образом: входные вектора представляют собой отцентрированные (лицо должно находиться в центре изображения и занимать большую его площадь) и приведённые к единому масштабу черно-белые изображения лица  $m$  на  $n$  пикселей. Собственные вектора, вычислен-

ные для всего набора изображений лиц, называются собственными лицами (*eigenfaces*). Метод главных компонент в применении к изображениям лиц так же называют методом собственных лиц. Собственные лица имеют полезное свойство, заключающееся в том, что изображение, соответствующее каждому такому вектору имеет лицеподобную форму [?, ?].

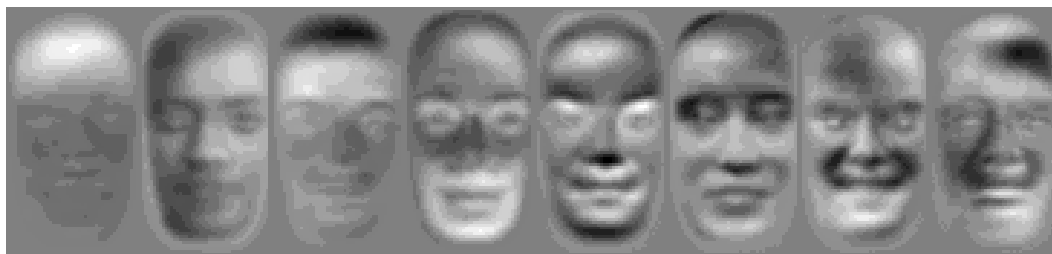


Рисунок 1.2 – Пример собственных лиц [?]

Полученный один раз на обучающей выборке изображений лиц набор собственных векторов используется для кодирования всех остальных изображений лиц, которые представляются взвешенной комбинацией этих собственных векторов [?]. Процесс распознавания заключается в сравнении главных компонент неизвестного изображения с компонентами всех остальных изображений. Для этого обычно применяют какую-либо метрику (простейший случай — Евклидово расстояние) [?].

## 2 Метод опорных векторов

### 3 Алгоритм Виолы-Джонса

Метод Виолы — Джонса — алгоритм, позволяющий обнаруживать и распознавать лица на изображениях и видеопоследовательностях в режиме реального времени [?]. Данный метод основан на использовании следующих принципов:

#### 1) Интегральное представление изображения

Представим исходное изображение в виде матрицы, каждый элемент которой содержит значение интенсивности пикселя. Интегральное представление изображения формируется путем создания новой матрицы, размерность которой совпадает с матрицей исходного изображения. В каждом элементе данной матрицы хранится сумма интенсивностей всех пикселей исходного изображения, находящихся левее и выше текущего пикселя в исходной матрице [?], т. е. элементы новой матрицы рассчитываются по следующей формуле:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j), \quad (3.1)$$

где  $I(i, j)$  — значение элемента матрицы исходного изображения,  $L(x, y)$  — значение элемента матрицы интегрального изображения.

Особенностью интегрального представления является возможность вычисления суммы интенсивностей пикселей внутри произвольных прямоугольных областей [?]. Рассмотрим следующий пример, где искомой областью является прямоугольник  $ABCD$ :

Сумма интенсивностей пикселей, лежащих внутри области  $ABCD$ , будет вычисляться по следующей формуле:

$$F(ABCD) = F(A) + F(D) - F(B) - F(C), \quad (3.2)$$

где  $F(X)$  — суммы интенсивностей пикселей внутри прямоугольной области  $X$ .

Таким образом, для вычисления суммы интенсивностей пикселей внут-

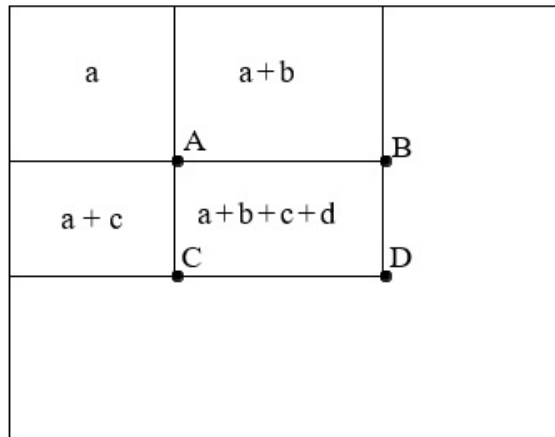


Рисунок 3.1 – Расчет суммы интенсивностей пикселей в прямоугольнике  $ABCD$

ри произвольных прямоугольных областей требуется четыре обращения к матрице интегрального представления изображения.

## 2) Признаки Хаара

Для выделения областей изображения, которые наиболее вероятно содержат нужные объекты, используются признаки Хаара. Признак Хаара — результат сравнения смежных прямоугольных областей изображения путем вычисления разности сумм интенсивностей пикселей соответствующих прямоугольников [?]. В методе Виолы — Джонса используются три вида данных признаков: «два прямоугольника», «три прямоугольника», «четыре прямоугольника» [?] (рис. 3.2 [?]).

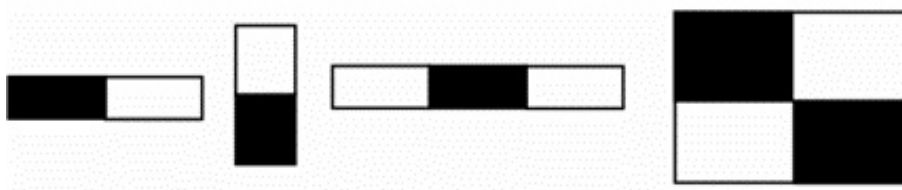


Рисунок 3.2 – Признаки Хаара

Значение каждого признака вычисляется по следующей формуле:

$$F = X - Y, \quad (3.3)$$

где  $X$  - сумма интенсивностей пикселей, находящихся в светлой области признака,  $Y$  - сумма интенсивностей пикселей, находящихся в



темной области признака.

Для вычисления данных сумм интенсивностей пикселей за константное время используется интегральное представление изображения.

### 3) Алгоритм AdaBoost

Для прогнозирования категории, к которой относится входной объект, в машинном обучении используется модель, называемая классификатором [?]. Слабый (или простой) классификатор — это классификатор, который решающий задачу классификации чуть лучше, чем случайное угадывание [?], т. е. вероятность ошибки меньше 50%.

Для повышения точности аналитических моделей используется бустинг (англ. boosting) — последовательная композиция алгоритмов машинного обучения, в которой каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов [?]. В методе Виолы — Джонса в качестве бустинга выбран алгоритм AdaBoost, в ходе выполнения которого формируется сложный классификатор, состоящий из набора простых и имеющих меньшую вероятность ошибки, чем у каждого слабого классификатора по отдельности:

$$a_s(x) = \sum_{i=1}^n b_i * a_i(x), \quad (3.4)$$

где  $x$  — изображение,  $a_s(x)$  — сложный классификатор,  $a_i(x)$  — слабый классификатор,  $b_i$  — весовой коэффициент соответствующего слабого классификатора.

Основные этапы работы алгоритма AdaBoost [?]:

- Инициализация весов объектов обучающего набора одинаковым значением;
- Для каждой итерации алгоритма на основе сравнения ошибок классификации выбирается слабый классификатор, который наилучшим образом разделяет обучающий набор с учетом текущих весов объектов;

- Для каждого выбранного классификатора вычисляется его ошибка и вес, отражающий точность разделения объектов;
- Обновление весов объектов таким образом, чтобы неправильно классифицированные объекты получили больший вес на следующей итерации;
- Формирование сложного классификатора на основе слабых классификаторов и их весов.

Таким образом, в ходе выполнения данного алгоритма путем исключения большинства имеющихся признаков выбираются наиболее подходящие для конкретного объекта, в результате чего формируется классификатор с критическими признаками.

В качестве классификаторов в методе Виолы — Джонса используются вышеописанные признаки Хаара. Для каждого признака выбирается соответствующий порог для бинарной классификации объектов, т. е. для определения того, какие значения признака считаются положительными, а какие — отрицательными. Выбор оптимального порога осуществляется таким образом, чтобы минимизировать ошибку классификации, учитывая веса объектов в процессе обучения алгоритма.

- 4) Каскадная структура классификаторов - последовательное объединение усложняющихся классификаторов в каскадную структуру (рис. 3.3 [?]) [?]. Данная структура повышает скорость обнаружения объектов, фокусируя свою работу на наиболее информативных областях изображения [?]. Это достигается тем, что каскад пытается отсеять как можно больше отрицательных экземпляров на самом раннем этапе: количество вычисленных признаков для анализа отрицательных областей должно быть значительно меньше, чем количество признаков, вычисленных для положительных областей. Следовательно, более сложные классификаторы применяются только к тем областям, которые уже прошли отсев более простыми классификаторами, что экономит ресурсы и улучшает производительность системы.

- 5) Сканирующее окно

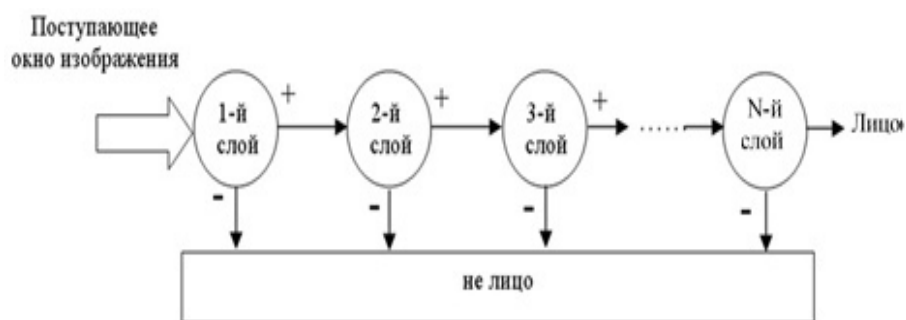


Рисунок 3.3 – Каскадная структура классификаторов

Для выделения областей, в которых могут находиться нужные объекты, используется сканирующее окно — перемещаемая прямоугольная активная область, в которой осуществляется поиск объектов [?]. Данный подход включает в себя пошаговый анализ различных прямоугольных подокон изображения, взятых с различными смещениями и масштабами. Впоследствии для каждой такой области применяются вышеописанные каскады классификаторов, которые оценивают, соответствует ли содержимое подокна определенным критериям.

Таким образом, схему работы метода Виолы — Джонса можно представить в следующем виде:

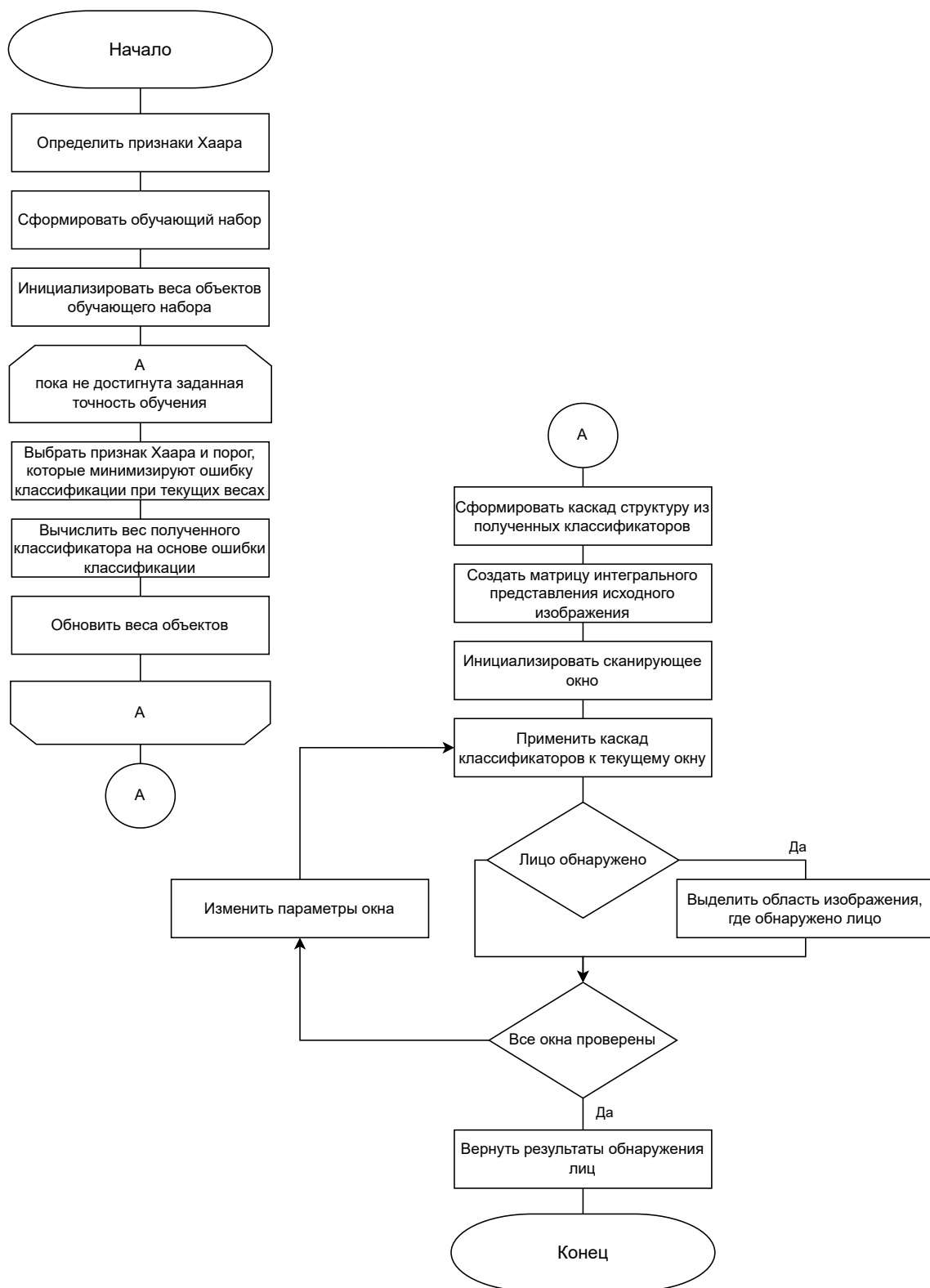


Рисунок 3.4 – Схема работы метода Виолы — Джонса

## 4 Метод гибкого сравнения на графах

Метод гибкого сравнения на графах – метод обработки изображений и распознавания образов, который используется для нахождения соответствия между двумя изображениями, учитывая возможные искажения, изменение масштаба и повороты [?].

Данный метод является одним из способов распознавания лиц. Лица представлены в виде графов со взвешенными вершинами и ребрами[?]. Во время распознавания один из графов — константный(эталонный), в то время как другой изменяется(деформируется) с целью наилучшей подгонки к первому.

В данном методе графы могут представлять собой как прямоугольную решетку (рис. 4.1.а), так и структуру, образованную антропометрическими точками лица(рис. 4.1.б).

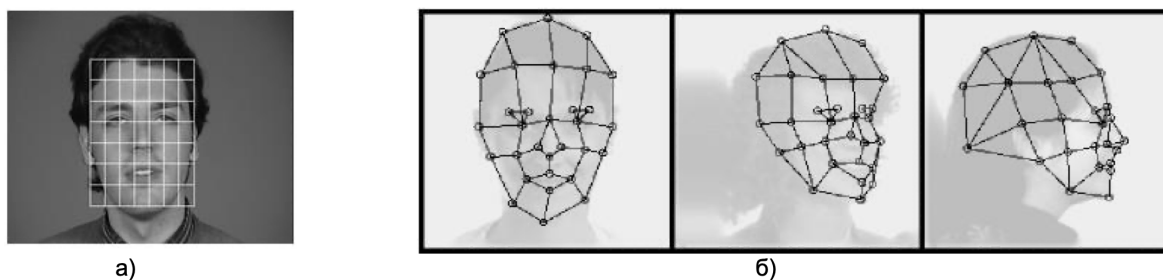


Рисунок 4.1 – Пример структуры графа для распознавания лиц  
а)Регулярная решетка;  
б)Граф на основе антропометрических точек лица.

Частотное содержимое изображения — характеристика, показывающая насколько быстро меняется яркость или цвет в различных частях изображения [?].

Фильтр Габора — фильтр, который анализирует, присутствует ли какое-либо конкретное частотное содержимое в изображении в различных направлениях в области вокруг точки или области анализа. Данный фильтр представляет собой синусоидальную плоскую волну(рис. 4.2).[?]

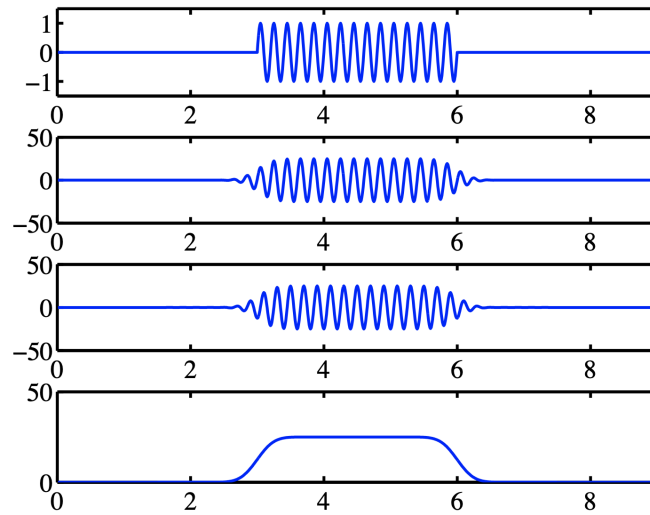


Рисунок 4.2 – Пример синусоидальной плоской волны.

Свертка – операция вычисления нового значения интенсивности, степени яркости светлых пикселей по сравнению с более темными тонами, заданного пикселя, при котором учитываются значения окружающих его соседних пикселей.

В некоторой локальной области вершины графа вычисляют значения путем свертки значений яркости пикселей с набором(рис. 4.3) фильтров Габора(рис. 4.4).

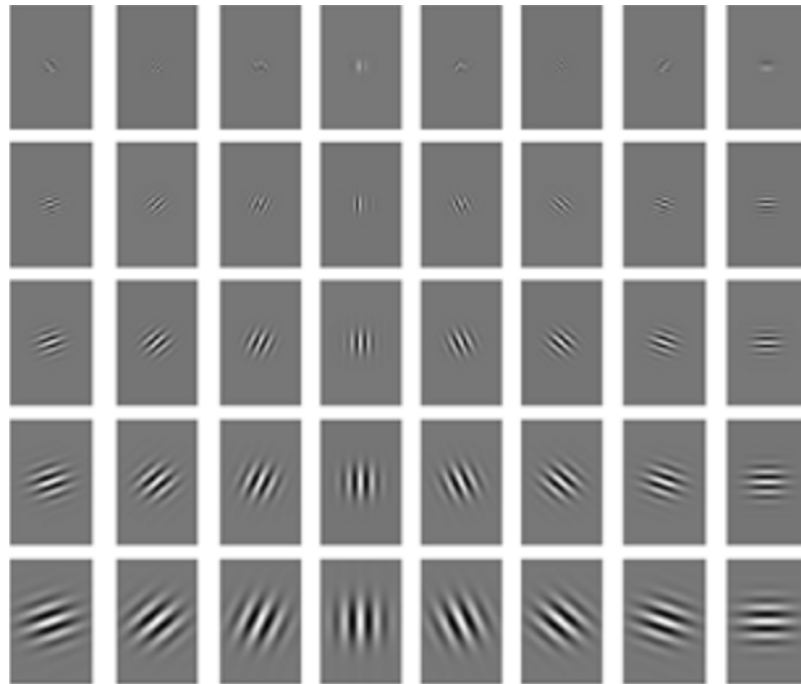


Рисунок 4.3 – Набор фильтров Габора.

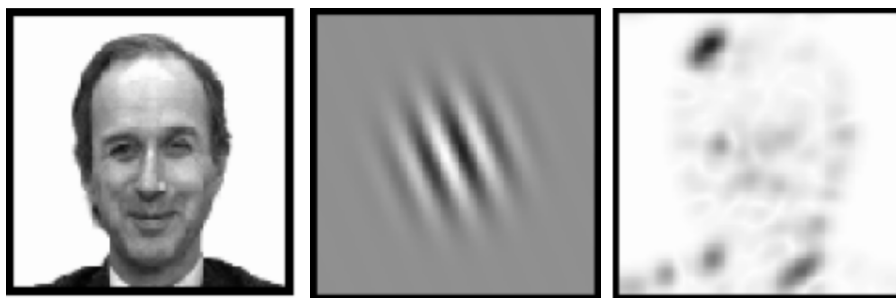


Рисунок 4.4 – Пример свертки изображения лица с фильтрами Габора.

### **Алгоритм распознавания лица:**

- 1) Происходит деформация графа путем смещения каждой из его вершин на некоторое расстояние в различных направлениях относительно его исходного местоположения(рис. 4.5);
- 2) Выбирается такая позиция, при которой разница между значениями в вершине деформируемого графа и соответствующей ей вершине эталонного графа будет минимальной;
- 3) Данная операция выполняется поочередно для всех вершин графа, пока не будет достигнуто наименьшее суммарное различие между признаками этих графов;
- 4) Данная процедура деформации выполняется для всех эталонных лиц, заложенных в базу данных системы. Результат распознавания – эталон с наименьшим суммарным различием.



Рисунок 4.5 – Пример деформации графа.