

Министерство образования и науки Российской Федерации  
Федеральное агентство по образованию Санкт-Петербургский Государственный  
Электротехнический Университет  
Факультет компьютерных технологий и информатики  
Кафедра математического обеспечения ЭВМ

## Исследование и улучшение метода гистограмм

курсовая работа

Работу выполнили студенты гр. 3305:  
Зубарев П.С.  
Свириденко С.В.

Санкт-Петербург 2007

# Оглавление

Введение . . . . .	3
Постановка задачи . . . . .	3
Гистограммный метод . . . . .	4
Вычисление гистограммы . . . . .	4
Применение гистограмм . . . . .	4
Модификации метода гистограмм . . . . .	5
Исследования . . . . .	6
Влияние освещения на гистограмму . . . . .	6
Модификация базы . . . . .	7
Модификация метода . . . . .	8
Текст программ . . . . .	8
Результаты . . . . .	18
Результаты работы на оригинальной базе ORL . . . . .	18
Результаты работы на модифицированной базе ORL . . . . .	19
Выводы . . . . .	19
Список литературы . . . . .	20

# Введение

Многие из нас ассоциируют биометрию или с идентификационными картами, или с контролем доступа на основе считывания отпечатков пальцев, или же с системами распознавания черт лица, используемыми при видео наблюдении, однако редко кто задумывался о действительном значении биометрии. Согласно определению, которое дает Оксфордский толковый словарь, биометрия — это «наука приложения статистических методов к биологическим проявлениям».

Цель биометрических технологий — создать механизм однозначной идентификации объекта, используя статистические измерения определенных характеристик или поведения этого объекта. Такая уникальная биометрическая информация может быть сохранена в электронном виде, а затем извлечена из базы данных и использована для идентификации.

Опознавание человека по лицу сегодня используется как метод идентификации буквально сплошь и рядом, однако многое при этом зависит от человека — например, от охранника, который должен определить, соответствуют ли черты лица фотографии в пропуске. Поистине переворотом стала техника сканирования лица, которая в биометрической индустрии сейчас занимает второе место после сканирования отпечатков пальцев.

Биометрическое опознавание лица, использующее специально разработанное программное обеспечение, избавляет от необходимости присутствия человека при проведении идентификации.

## Постановка задачи

Методы опознавания основаны на преобразовании черт конкретного лица в алгоритмическую модель, которая сравнивается или с фотографии на пропуске, или с содержимым базы фотографических данных. Проще говоря, для каждого образа создается уникальный «пароль», содержащий характеристики черт лица. По лицу человека можно узнать его историю, симпатии и антипатии, болезни, эмоциональное состояние, чувства и намерения по отношению к окружающим. Всё это представляет особый интерес для автоматического распознавания лиц (например, для выявления потенциальных преступников).

Наиболее часто на практике распознавания лиц человека применяют яркостные методы, как способ представления характеристик лица человека. Среди причин применения яркостных методов можно выделить две основные.

Во-первых, яркостные признаки по своей сути представляют любое цифровое изображение и не изменяются при плоских поворотах и изменении размеров.

Во-вторых, яркостные признаки позволяют выделить области с резким перепадом яркости, которые могут соответствовать определенным областям лица. Сами перепады будут находиться на границах этих областей.

Процесс распознавания можно определить следующим образом. Пусть есть несколько разных изображений лиц или образов заданного класса<sup>1</sup>, которых находятся в базе данных. Каждый такой образ можно представить как эталон. В процессе распознавания на вход системе поступает новый образ, который необходимо идентифицировать. Для этого необходимо проверить принадлежность этого образа базе данных: вычисляя либо некоторую меру подобия между новым образом и каждым эталоном из базы, либо меру подобия между некоторой характеристикой нового образа и совместной характеристикой образов в классе.

Новый образ будет принадлежать тому классу, мера подобия с которым будет наивысшей. Однако в случае если на вход системе поступит изображение для которого

---

<sup>1</sup>изображения одного и того же объекта относятся к одному классу

нет эталона в базе, будет произведена неверная идентификация. Для того чтобы отсеять часть неверных решений, можно ввести минимальный порог подобия. В таком случае если наивысшая мера подобия между входным образом и всеми эталонами получилась меньше этого порога, то можно сказать, что изображение не было идентифицировано.

Таким образом весь процесс распознавания можно разложить на следующие этапы:

- детекция области лица на исходном изображении;
- экстракция признаков — представления изображения выделенного лица в форме вектора исходных признаков;
- селекции некоторых признаков из полного набора или редукции исходного пространства признаков;
- сравнения признаков нового образа с признаками эталонов;
- принятие решения о принадлежности этого образа к одному из известных классов.

## Гистограммный метод

Одним из главных этапов процесса распознавания является этап экстракции признаков. Самым простым, но в тоже время и эффективным методом является гистограммный метод, который заключается в вычислении гистограммы яркости исходного изображения и сведения ее значений в вектор исходных (гистограммных) признаков.

### Вычисление гистограммы

Вычисление одномерной гистограммы яркости исходного изображения и сведения её значений в вектор гистограммных признаков заключается в следующем: каждый элемент гистогрaммы  $H(j)$  определяется количеством пикселей исходного изображения, имеющих значения яркости, равное  $j = 0, 1, 2 \dots 255$ .  $H(j)$  — вектор признаков размерностью 256. Порой работать с таким большим вектором признаков не очень удобно, и имеет смысл произвести редукцию признаков к пространству  $DIM$ . При редукции пространства признаков гистограмма  $H(j)$  преобразуется к следующему виду:

$$H(b) = \sum_{j=(b-1) \frac{256}{BIN}}^{b \frac{256}{BIN}} H(j), b = 1, 2, \dots, BIN$$

В случае если исходное изображение и эталонное изображение, имеют различные размеры, то вектор признаков необходимо нормировать:

$$H_{norm}(b) = \frac{H(b)}{MN}, b = 1, 2, \dots, BIN$$

, где  $M, N$  — число строк и число столбцов в исходном изображении.

### Применение гистограмм

Большим преимуществом гистограммного метода является его хорошая стойкость к геометрическим преобразованиям исходной картинки. Так если даже исходное изображение повернуть то его гистограммы останутся прежней.

На рисунке 1 представлено исходное изображение, для которого построена гистограмма (рис. 2). Затем исходное изображение было разрезано на маленькие кусочки, которые были перемешаны внутри изображения (рис. 3). Если сравнить гистограммы исходного (рис. 2) и измененного (рис. 4) изображения, то можно обратить внимание, что они совпадают. Это лишний раз показывает стойкость метода по отношению к геометрическим преобразованиям изображений. Такая особенность гистограммного метода может быть с легкостью применена в криминалистике. Например, если необходимо



Рис. 1: Исходное изображение

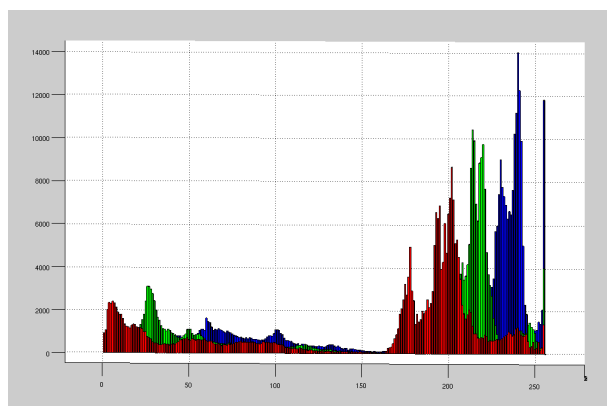


Рис. 2: Гистограмма исходного изображения



Рис. 3: Измененное изображение

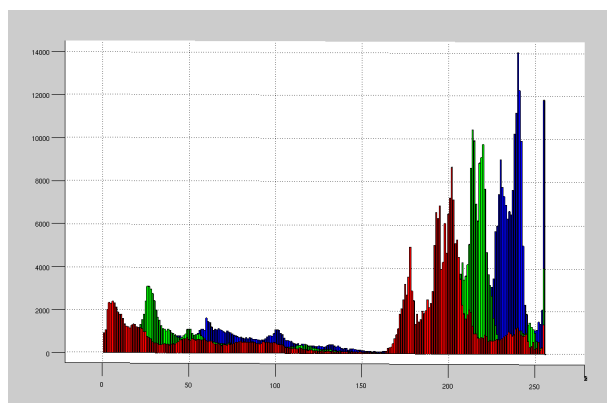


Рис. 4: Гистограмма измененного изображения

идентифицировать личность, когда на входе имеется изображение, порезанное на множество маленьких кусочков.

Помимо всего следует отметить некоторые особенности гистограммного метода. Например, чем больше площадь фона вокруг области лица, тем заметнее различие гистограмм для разных изображений лиц даже при одном и том же фоне. Кроме этого гистограммный метод может применяться в случае, когда на изображениях фон различается несущественно.

Однако следует помнить, что гистограмма — это яркостная характеристика, поэтому два структурно или текстурно одинаковых, но имеющих разную яркость изображения будут иметь в общем случае различные по форме гистограммы: от циклического сдвига по отношению друг к другу, до циклического сдвига и искажений на границах.

## Модификации метода гистограмм

Возможны несколько способов модификации метода гистограмм. Один из видов модификации — метод, направленный на ускорение вычисления вектора исходных признаков. В связи с тем что лицо человека достаточно симметрично, то гистограммы левой и правой части лица практически идентичны, в результате чего можно воспользоваться только одной половиной лица и построить по нему вектор исходных признаков, который по качественным показателям не будет отличаться от вектора признаков, построенного по всему лицу. В случае разделении лица по горизонтали произойдет существенная потеря данных и в таком случае метод будет неработоспособным.

В случае же если у нас в базе данных мало изображений одного класса, с помо-

щью разделения изображения лица на две половинки и отражении каждой из половинок относительно оси разделения, можно получить два эталонных изображения вместо одного, что может существенно улучшить результат распознавания.

## Исследования

### Влияние освещения на гистограмму

Рассмотрим, как меняется гистограмма изображения при освещении и затемнении: В данном случае освещение и затемнение были выполнены в графическом редак-

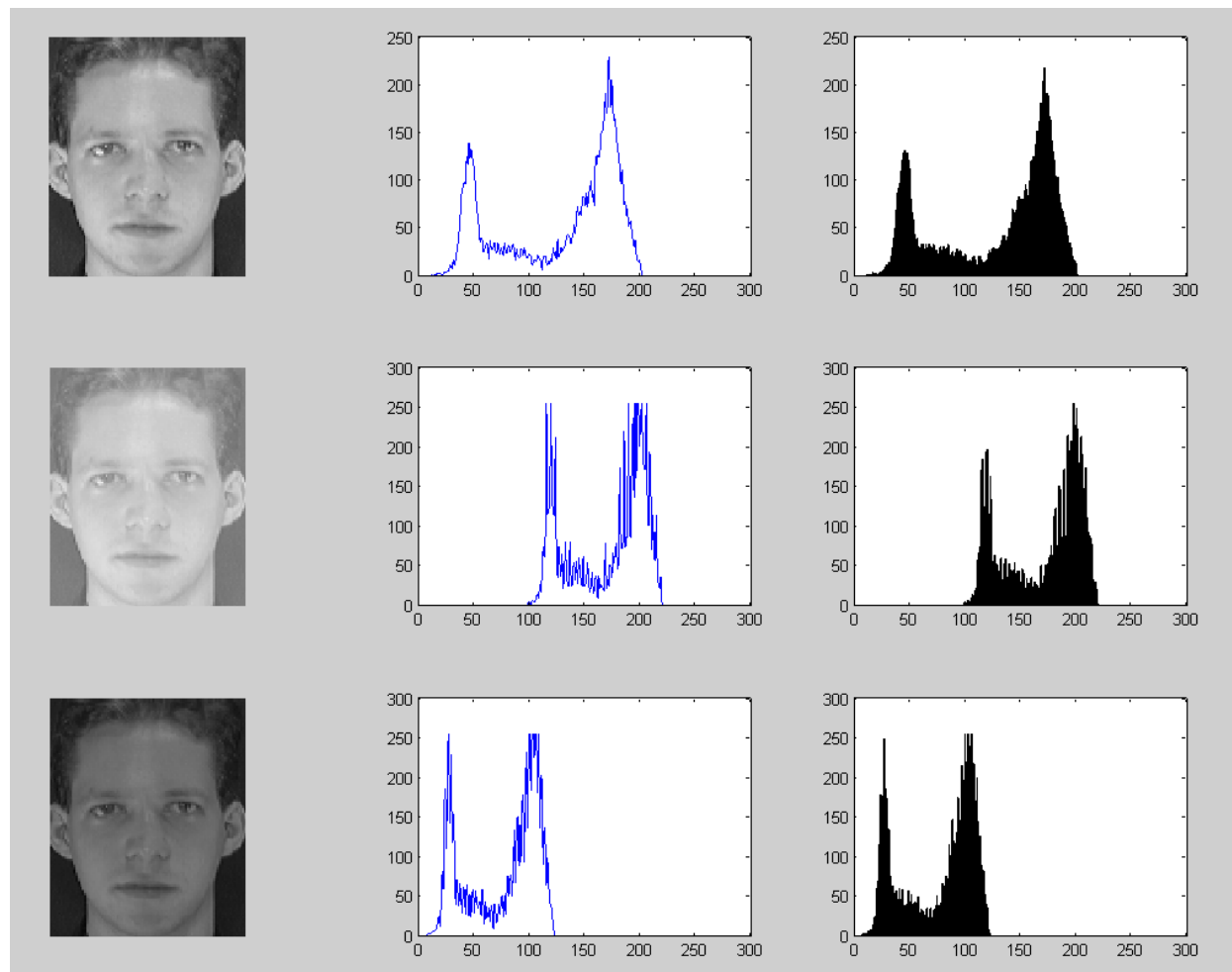


Рис. 5: Влияние освещения/затемнения изображения на характер гистограммы

торе. Мы видим, что гистограмма не просто сдвинулась на определённую величину, но и сжалась по горизонтали. Тем не менее, снимки, сделанные на фотоаппараты и камеры, выглядят отлично от представленных образцов. Всё это приводит нас к мысли, что редакторы несколько «облагораживают» картинку.

Действительно, сравним изображения, осветлённые на одинаковую величину в графическом редакторе и программой на MATLAB:

Здесь некоторая схожесть улавливается, хотя нижняя картинка больше похожа на то, что мы видели бы в реальной жизни. Попробуем сделать ещё большее осветление(рис. 6). Мы видим характерную картину: редактор сжал гистограмму для сохранения читабельности, а MATLAB произвёл более адекватные преобразования, но при этом необратимо «испортил» картинку (пиксели с яркостью, большей максимума, сформировали ярко выраженный горб на правой границе гистограммы).

Также можно заметить, что при большом осветлении происходит группировка

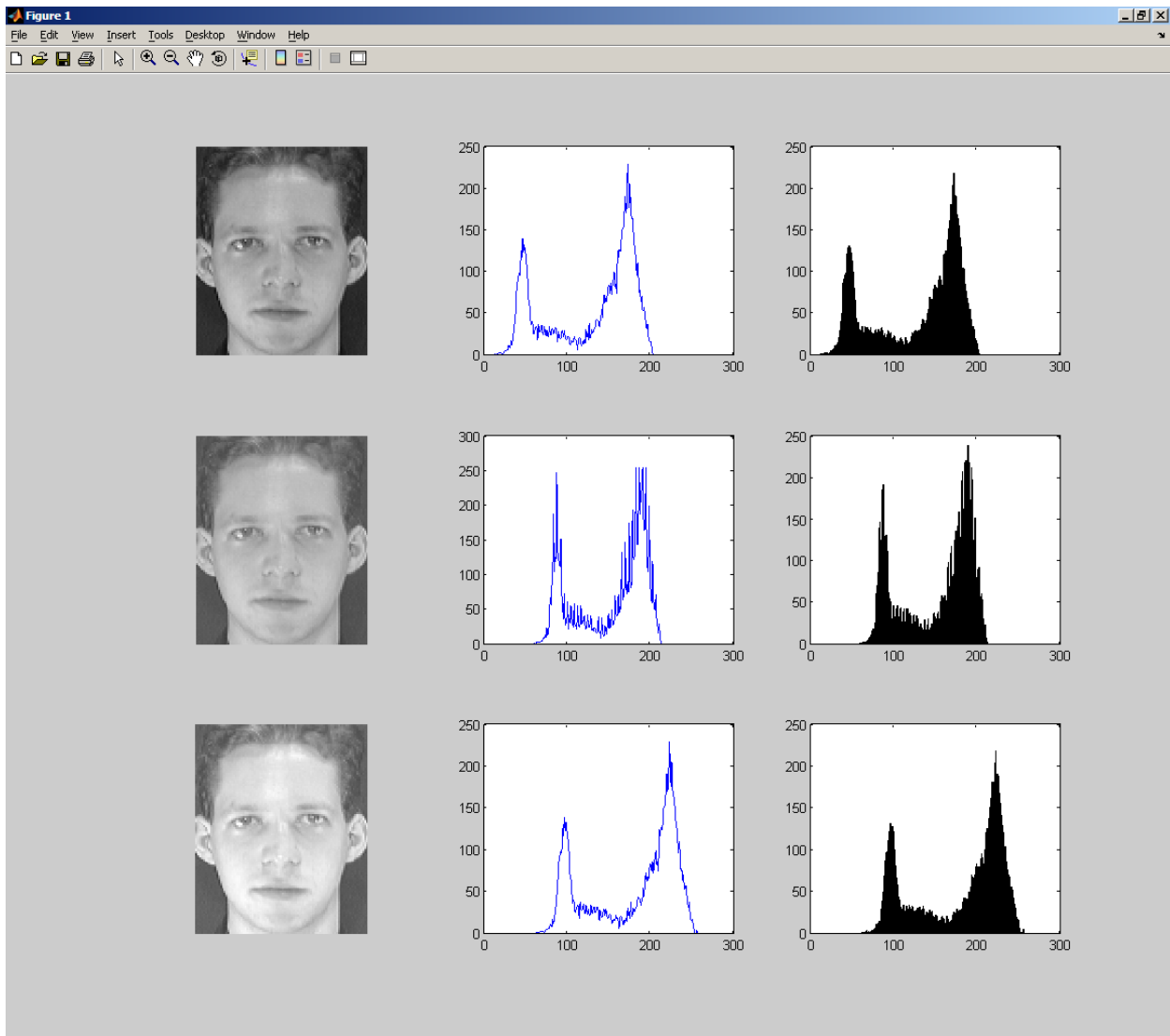


Рис. 6: Осветление +50. Изображения сверху вниз: оригинал, осветление IrfanView, осветление MATLAB

пикселей по правому краю гистограммы, что делает практически невозможным правильное распознавание (т.к. все изображения будут иметь приблизительно одинаковый характер гистограммы, мелкие всплески с максимальным, прижатым к правому краю). Справедливости ради, отметим, что любой другой метод тоже будет иметь большие трудности с распознаванием таких изображений.

## Модификация базы

Нашей задачей будет улучшение гистограммного метода, направленное на распознавание изображений, подвергшихся изменению яркости.

В качестве первой тестовой базы изображений мы будем использовать базу ORL — 40 классов, 10 чёрно-белых изображений  $92 \times 112$ .

В качестве второй мы будем использовать её же, но изображения с чётными номерами будут осветлены на 50 пунктов.

Для того, чтобы выделить различия более явно, мы проведём 2 серии испытаний, для одного тестового изображения в базе, и для двух (один оригинал плюс одно осветлённое изображение).

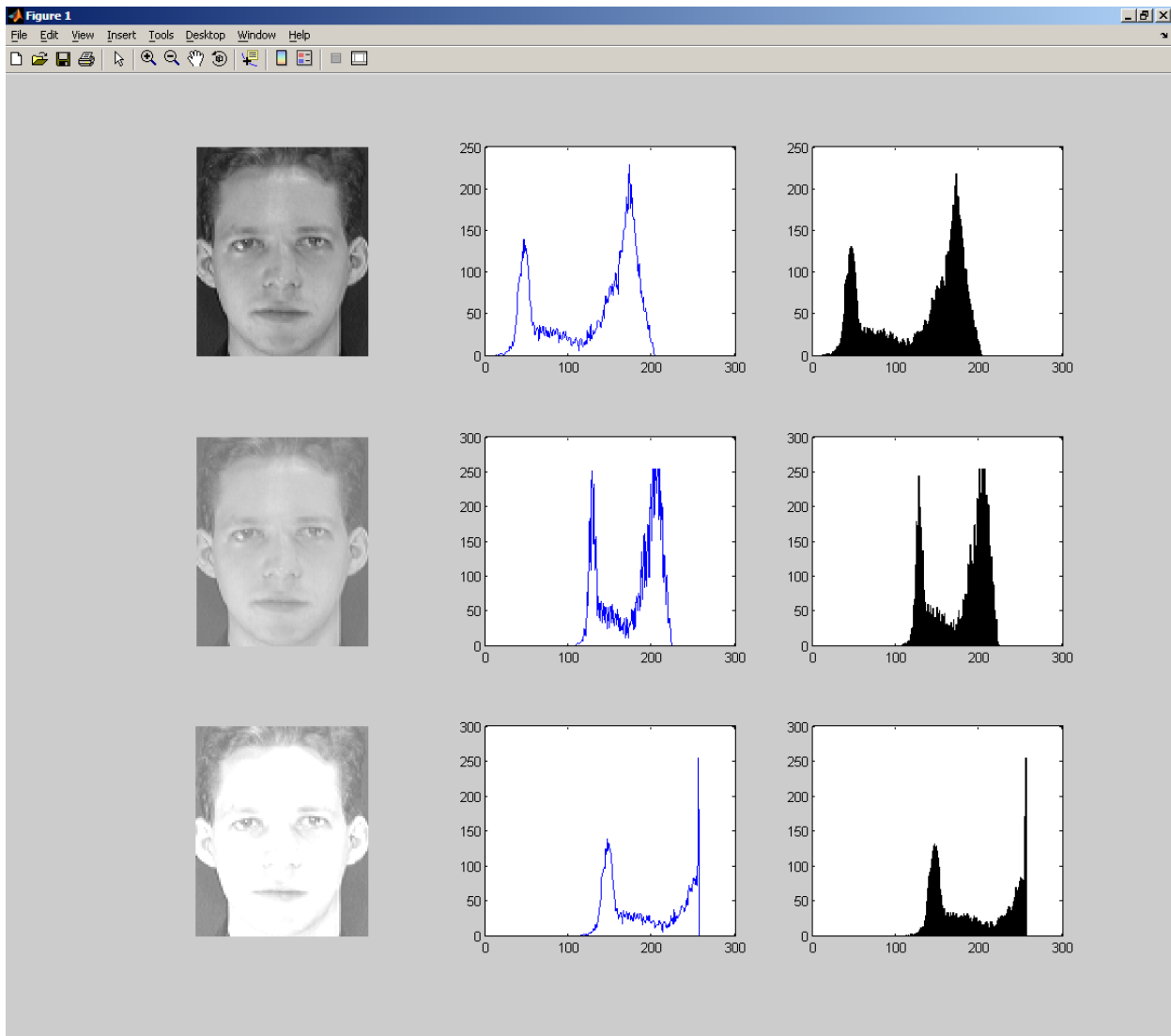


Рис. 7: Осветление +100. Изображения сверху вниз: оригинал, осветление IrfanView, осветление MATLAB

## Модификация метода

Модификация метода будет заключаться в расширении гистограммы тестируемого изображения слева и справа на количество столбцов, равное их количеству в оригинальной гистограмме. Затем гистограмма тестируемого изображения будет перемещаться по эталонной гистограмме, минимизируя метрику L2 отклонения между ними. Таким образом, мы должны верно распознать даже те изображения, которые были осветлены.

## Текст программ

```
% brightness_test.m
% влияние освещения на гистограмму
%

% очистка экрана
clear all; clc;

% префикс пути к картинке
PATH_PREFIX = 'D:\PROJECTS\A-team-777\9\kml\kurs\brightness_test\img\';

% первая
PIC1_PATH = [PATH_PREFIX, '1.jpg'];
```



```

% вторая
PIC2_PATH = [PATH_PREFIX, '2.jpg'];

% третья
PIC3_PATH = [PATH_PREFIX, '3.jpg'];

% количество уровней гистограммы
LEVELS = 256;

% размер блока гистограммы
hist_block_size = uint8( 256 / LEVELS );

% 1-ая картинка

image1 = imread( PIC1_PATH );    % чтение изображения из файла

figure( 1 );
subplot( 3, 3, 1 );
imshow( image1 );    % показываем картинку

full_spectrum1 = uint8( zeros( 1, 256 ) );
for i = 1 : size( image1, 1 )
    for j = 1: size( image1, 2 )
        color = image1( i, j );
        full_spectrum1( color + 1 ) = full_spectrum1( color + 1 ) + 1;
    end;
end;    % расчёт полной гистограммы

figure( 1 );
subplot( 3, 3, 2 );
plot( full_spectrum1 ); % рисуем полную гистограмму

short_spectrum1 = uint8( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum1( i ) = mean( full_spectrum1( 1, ( i - 1 ) * hist_block_size + 1 : i
        * hist_block_size + 1 ) );
end;    % расчёт укороченной гистограммы

figure( 1 );
subplot( 3, 3, 3 );
bar( short_spectrum1 ); % рисуем укороченную гистограмму

% 2-я картинка

image2 = imread( PIC2_PATH );    % чтение изображения из файла

figure( 1 );
subplot( 3, 3, 4 );
imshow( image2 );    % показываем картинку

full_spectrum2 = uint8( zeros( 1, 256 ) );
for i = 1 : size( image2, 1 )
    for j = 1: size( image2, 2 )
        color = image2( i, j );
        full_spectrum2( color + 1 ) = full_spectrum2( color + 1 ) + 1;
    end;
end;    % расчёт полной гистограммы

figure( 1 );
subplot( 3, 3, 5 );
plot( full_spectrum2 ); % рисуем полную гистограмму

short_spectrum2 = uint8( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum2( i ) = mean( full_spectrum2( 1, ( i - 1 ) * hist_block_size + 1 : i
        * hist_block_size + 1 ) );
end;    % расчёт укороченной гистограммы

figure( 1 );
subplot( 3, 3, 6 );
bar( short_spectrum2 ); % рисуем укороченную гистограмму

% 3-я картинка

image3 = imread( PIC3_PATH );    % чтение изображения из файла

```

```

figure( 1 );
subplot( 3, 3, 7 );
imshow( image3 ); % показываем картинку

full_spectrum3 = uint8( zeros( 1, 256 ) );
for i = 1 : size( image3, 1 )
    for j = 1: size( image3, 2 )
        color = image3( i, j );
        full_spectrum3( color + 1 ) = full_spectrum3( color + 1 ) + 1;
    end;
end; % расчёт полной гистограммы

figure( 1 );
subplot( 3, 3, 8 );
plot( full_spectrum3 ); % рисуем полную гистограмму

short_spectrum3 = uint8( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum3( i ) = mean( full_spectrum3( 1, ( i - 1 ) * hist_block_size + 1 : i
        * hist_block_size + 1 ) );
end; % расчёт укороченной гистограммы

figure( 1 );
subplot( 3, 3, 9 );
bar( short_spectrum3 ); % рисуем укороченную гистограмму

% editor_vs_matlab.m
% сравнение освещения при помощи редактора и через MATLAB

% очистка экрана
clear all; clc;

% префикс пути к картинке
PATH_PREFIX = 'D:\PROJECTS\а-team-777\9\kml\kurs\editor_vs_matlab\img\';

% первая
PIC1_PATH = [PATH_PREFIX, '1.jpg'];

% вторая
PIC2_PATH = [PATH_PREFIX, '2.jpg'];

% третья
PIC3_PATH = [PATH_PREFIX, '3.jpg'];

% количество уровней гистограммы
LEVELS = 256;

% освещение/затемнение
COLOR_CHANGE = 50;

% размер блока гистограммы
hist_block_size = uint8( 256 / LEVELS );

% 1-ая картинка — оригинал

image1 = imread( PIC1_PATH ); % чтение изображения из файла

figure( 1 );
subplot( 3, 3, 1 );
imshow( image1 ); % показываем картинку

full_spectrum1 = uint8( zeros( 1, 256 ) );
for i = 1 : size( image1, 1 )
    for j = 1: size( image1, 2 )
        color = image1( i, j );
        full_spectrum1( color + 1 ) = full_spectrum1( color + 1 ) + 1;
    end;
end; % расчёт полной гистограммы

figure( 1 );
subplot( 3, 3, 2 );
plot( full_spectrum1 ); % рисуем полную гистограмму

short_spectrum1 = uint8( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum1( i ) = mean( full_spectrum1( 1, ( i - 1 ) * hist_block_size + 1 : i
        * hist_block_size + 1 ) );
end;

```

```

end;      % расчёт укороченной гистограммы

figure( 1 );
subplot( 3, 3, 3 );
bar( short_spectrum1 ); % рисуем укороченную гистограмму

% 2-я картинка – освещение редактором +50

image2 = imread( PIC2_PATH );      % чтение изображения из файла

figure( 1 );
subplot( 3, 3, 4 );
imshow( image2 );      % показываем картинку

full_spectrum2 = uint8( zeros( 1, 256 ) );
for i = 1 : size( image2, 1 )
    for j = 1: size( image2, 2 )
        color = image2( i, j );
        full_spectrum2( color + 1 ) = full_spectrum2( color + 1 ) + 1;
    end;
end;      % расчёт полной гистограммы

figure( 1 );
subplot( 3, 3, 5 );
plot( full_spectrum2 ); % рисуем полную гистограмму

short_spectrum2 = uint8( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum2( i ) = mean( full_spectrum2( 1, ( i - 1 ) * hist_block_size + 1 : i
        * hist_block_size + 1 ) );
end;      % расчёт укороченной гистограммы

figure( 1 );
subplot( 3, 3, 6 );
bar( short_spectrum2 ); % рисуем укороченную гистограмму

% 3-я картинка – ручное освещение + 50

image3 = imread( PIC1_PATH );      % чтение изображения из файла

full_spectrum3 = uint8( zeros( 1, 256 ) );
for i = 1 : size( image3, 1 )
    for j = 1: size( image3, 2 )
        color = image3( i, j ) + COLOR_CHANGE;
        if color < 0
            color = 0;
        elseif color > 255
            color = 255;
        end;
        full_spectrum3( color + 1 ) = full_spectrum3( color + 1 ) + 1;
        image3( i, j ) = color;
    end;
end;      % расчёт полной гистограммы, коррекция цвета

imwrite( image3, PIC3_PATH ); % запись полученной картинки в файл

figure( 1 );
subplot( 3, 3, 7 );
imshow( image3 );      % показываем полученную картинку

figure( 1 );
subplot( 3, 3, 8 );
plot( full_spectrum3 ); % рисуем полную гистограмму

short_spectrum3 = uint8( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum3( i ) = mean( full_spectrum3( 1, ( i - 1 ) * hist_block_size + 1 : i
        * hist_block_size + 1 ) );
end;      % расчёт укороченной гистограммы

figure( 1 );
subplot( 3, 3, 9 );
bar( short_spectrum3 ); % рисуем укороченную гистограмму

% colors.m
clear all; clc;

```

```

PATH = '/home/zps/test.png'

image = imread( PATH );

red = image( :, :, 1 );
green = image( :, :, 2 );
blue = image( :, :, 3 );

red_s = uint32( zeros( 1, 256 ) );
for i = 1 : size( red, 1 )
    for j = 1: size( red, 2 )
        color = red( i, j );
        red_s( color + 1 ) = red_s( color + 1 ) + 1;
    end;
end;

green_s = uint32( zeros( 1, 256 ) );
for i = 1 : size( green, 1 )
    for j = 1: size( green, 2 )
        color = green( i, j );
        green_s( color + 1 ) = green_s( color + 1 ) + 1;
    end;
end;

blue_s = uint32( zeros( 1, 256 ) );
for i = 1 : size( blue, 1 )
    for j = 1: size( blue, 2 )
        color = blue( i, j );
        blue_s( color + 1 ) = blue_s( color + 1 ) + 1;
    end;
end;

figure(1);
subplot( 1, 1, 1 );
imshow( image );

figure(2);
subplot( 1, 1, 1 );
RRR = [red_s; green_s; blue_s]';
bar3(RRR, 'detached');
colormap([1 0 0;0 1 0;0 0 1]);

PATH = '/home/zps/test2.png'

image = imread( PATH );

red = image( :, :, 1 );
green = image( :, :, 2 );
blue = image( :, :, 3 );

red_s = uint32( zeros( 1, 256 ) );
for i = 1 : size( red, 1 )
    for j = 1: size( red, 2 )
        color = red( i, j );
        red_s( color + 1 ) = red_s( color + 1 ) + 1;
    end;
end;

green_s = uint32( zeros( 1, 256 ) );
for i = 1 : size( green, 1 )
    for j = 1: size( green, 2 )
        color = green( i, j );
        green_s( color + 1 ) = green_s( color + 1 ) + 1;
    end;
end;

blue_s = uint32( zeros( 1, 256 ) );
for i = 1 : size( blue, 1 )
    for j = 1: size( blue, 2 )
        color = blue( i, j );
        blue_s( color + 1 ) = blue_s( color + 1 ) + 1;
    end;
end;

figure(3);
subplot( 1, 1, 1 );
imshow( image );

```

```

figure(4);
subplot( 1, 1, 1 );
RRR = [red_s; green_s; blue_s]';
bar3(RRR, 'detached');
colormap([1 0 0;0 1 0;0 0 1]);

% hist.m
% стандартный гистограммный метод

% очистка
clear all; clc;

% параметры
DB_PATH = 'D:\LETI\9\knil\bases\orl_bmp\s'; % путь к базе
LEVELS = 256; % количество уровней в гистограмме
K = 40; % количество классов изображений
L = 3; % количество изображений в базе
Q = 10; % общее количество изображений в классе
no_gui = 1; % не выводить надоедливые окна, печатать результат в консоль в конце работы

% инициализация
rec_rate = 0; % количество распознанных верно изображений
rec_percent = 0; % процент распознанных верно изображений
takt = 0; % номер шага
hist_block_size = uint32( 256 / LEVELS ); % размер шага в гистограмме
image_db = []; % база изображений
size_db = []; % база размеров изображений
full_spectrum_db = []; % база полных гистограмм
short_spectrum_db = []; % база коротких гистограмм

% построение баз
for class = 1 : K
    for file = 1 : L
        % чтение из файла
        image = imread( [DB_PATH, num2str( class ), '\', num2str( file ), '.bmp'] );

        % добавление изображения
        image_db = [image_db image( : )];

        % добавление размера
        vec = size( image );
        size_db = [size_db vec'];

        % рисуем исходное изображение
        if no_gui == 0
            figure( 1 );
            subplot( 1, 3, 1 );
            imshow( image );
        end;

        % считаем полную гистограмму
        full_spectrum = uint32( zeros( 1, 256 ) );
        for i = 1 : size( image, 1 )
            for j = 1: size( image, 2 )
                color = image( i, j );
                full_spectrum( color + 1 ) = full_spectrum( color + 1 ) + 1;
            end;
        end;

        % добавляем полную гистограмму в базу
        full_spectrum_db = [full_spectrum_db full_spectrum( : )];

        % рисуем полную гистограмму
        if no_gui == 0
            figure( 1 );
            subplot( 1, 3, 2 );
            plot( full_spectrum );
        end;

        % считаем короткую гистограмму
        short_spectrum = uint32( zeros( 1, LEVELS ) );
        for i = 1 : LEVELS
            short_spectrum( i ) = mean( full_spectrum( 1, ( i - 1 ) * hist_block_size + 1 :
                i * hist_block_size ) );
        end;
    end;
end;

```

```

% добавляем короткую гистограмму в базу
short_spectrum_db = [short_spectrum_db short_spectrum( : )];

% рисуем короткую гистограмму
if no_gui == 0
    figure( 1 );
    subplot( 1, 3, 3 );
    bar( short_spectrum );
end;

% ждёмс
if no_gui == 0
    pause( 0.04 );
end;
end;

% производим распознавание
for class = 1 : K
    for file = L + 1 : Q
        % следующий шаг
        takt = takt + 1;

        % чтение изображения из файла
        image = imread( [DB_PATH, num2str( class ), '\', num2str( file ), '.bmp'] );

        % рисуем исходное изображение
        if no_gui == 0
            figure( 2 );
            subplot( 2, 3, 1 );
            imshow( image );
        end;

        % считаем полную гистограмму
        full_spectrum = uint32( zeros( 1, 256 ) );
        for i = 1 : size( image, 1 )
            for j = 1: size( image, 2 )
                color = image( i, j );
                full_spectrum( color + 1 ) = full_spectrum( color + 1 ) + 1;
            end;
        end;

        % рисуем полную гистограмму
        if no_gui == 0
            figure( 2 );
            subplot( 2, 3, 2 );
            plot( full_spectrum );
        end;

        % считаем короткую гистограмму
        short_spectrum = uint32( zeros( 1, LEVELS ) );
        for i = 1 : LEVELS
            short_spectrum( i ) = mean( full_spectrum( 1, ( i - 1 ) * hist_block_size + 1 :
                i * hist_block_size ) );
        end;

        % рисуем короткую гистограмму
        if no_gui == 0
            figure( 2 );
            subplot( 2, 3, 3 );
            bar( short_spectrum );
        end;

        % выравниваем вектор в столбец
        vector = short_spectrum( : );

        % клонируем вектор до размера базы
        cloned_vector = kron( vector, uint32( ones( 1, K * L ) ) );

        % рассчитываем и сортируем расстояние по метрике L2
        delta = sum( ( short_spectrum_db - cloned_vector ).^2 );
        [ value, index ] = sort( delta );

        % рисуем распознанное изображение и его гистограммы ниже исходного изображения
        im_vector = image_db( :, index( 1 ) );
        size_vector = size_db( :, index( 1 ) );

```

```

rec_image = reshape( im_vector, size_vector' );

if no_gui == 0
    figure( 2 );
    subplot( 2, 3, 4 );
    imshow( rec_image );

    figure( 2 );
    subplot( 2, 3, 5 );
    full_sp_vector = full_spectrum_db( :, index( 1 ) );
    plot( full_sp_vector' );

    figure( 2 );
    subplot( 2, 3, 6 );
    short_sp_vector = short_spectrum_db( :, index( 1 ) );
    bar( short_sp_vector' );
end;

% ищем класс распознанного изображения
class_new = fix( ( index( 1 ) - 1 ) / L ) + 1;
if class == class_new
    rec_rate = rec_rate + 1;
end;

% пересчитываем процент распознавания
rec_percent = ( rec_rate / takt ) * 100;

if no_gui == 0
    subplot( 2, 3, 4 );
    title( [ 'Recognition: ', num2str( rec_percent ) ] );
end;

% ждёмс
if no_gui == 0
    pause( 0.04 );
end;
end;
end;

% вывод результата
rec_percent

% hist2.m
% модификация гистограммного метода

% очистка
clear all; clc;

% параметры
DB_PATH = 'D:\LETI\9\kml\bases\orl_bmp\s'; % путь к базе
LEVELS = 256; % количество уровней в гистограмме
K = 40; % количество классов изображений
L = 3; % количество изображений в базе
Q = 10; % общее количество изображений в классе
no_gui = 1; % не выводить надоедливые окна, печатать результат в консоль в конце работы

% инициализация
rec_rate = 0; % количество распознанных верно изображений
rec_percent = 0; % процент распознанных верно изображений
takt = 0; % номер шага
hist_block_size = uint32( 256 / LEVELS ); % размер шага в гистограмме
image_db = []; % база изображений
size_db = []; % база размеров изображений
full_spectrum_db = []; % база полных гистограмм
short_spectrum_db = []; % база коротких гистограмм

% построение баз
for class = 1 : K
    for file = 1 : L
        % чтение из файла
        image = imread( [DB_PATH, num2str( class ), '\', num2str( file ), '.bmp'] );

        % добавление изображения
        image_db = [image_db image( : )];

        % добавление размера
        vec = size( image );
    end
end

```

```

size_db = [size_db vec'];

% рисуем исходное изображение
if no_gui == 0
    figure( 1 );
    subplot( 1, 3, 1 );
    imshow( image );
end;

% считаем полную гистограмму
full_spectrum = uint32( zeros( 1, 256 ) );
for i = 1 : size( image, 1 )
    for j = 1: size( image, 2 )
        color = image( i, j );
        full_spectrum( color + 1 ) = full_spectrum( color + 1 ) + 1;
    end;
end;

% добавляем полную гистограмму в базу
full_spectrum_db = [full_spectrum_db full_spectrum( : )];

% рисуем полную гистограмму
if no_gui == 0
    figure( 1 );
    subplot( 1, 3, 2 );
    plot( full_spectrum );
end;

% считаем короткую гистограмму
short_spectrum = uint32( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum( i ) = mean( full_spectrum( 1, ( i - 1 ) * hist_block_size + 1 :
        i * hist_block_size ) );
end;

% добавляем короткую гистограмму в базу
short_spectrum_db = [short_spectrum_db short_spectrum( : )];

% рисуем короткую гистограмму
if no_gui == 0
    figure( 1 );
    subplot( 1, 3, 3 );
    bar( short_spectrum );
end;

% ждёмс
if no_gui == 0
    pause( 0.04 );
end;
end;
end;

% производим распознавание
for class = 1 : K
    for file = L + 1 : Q
        % следующий шаг
        takt = takt + 1;

        % чтение изображения из файла
        image = imread( [DB_PATH, num2str( class ), '\', num2str( file ), '.bmp'] );

        % рисуем исходное изображение
        if no_gui == 0
            figure( 2 );
            subplot( 2, 3, 1 );
            imshow( image );
        end;

        % считаем полную гистограмму
        full_spectrum = uint32( zeros( 1, 256 ) );
        for i = 1 : size( image, 1 )
            for j = 1: size( image, 2 )
                color = image( i, j );
                full_spectrum( color + 1 ) = full_spectrum( color + 1 ) + 1;
            end;
        end;
    end;
end;

```



```

% рисуем полную гистограмму
if no_gui == 0
    figure( 2 );
    subplot( 2, 3, 2 );
    plot( full_spectrum );
end;

% считаем короткую гистограмму
short_spectrum = uint32( zeros( 1, LEVELS ) );
for i = 1 : LEVELS
    short_spectrum( i ) = mean( full_spectrum( 1, ( i - 1 ) * hist_block_size + 1 :
        i * hist_block_size ) );
end;

% рисуем короткую гистограмму
if no_gui == 0
    figure( 2 );
    subplot( 2, 3, 3 );
    bar( short_spectrum );
end;

% выравниваем вектор в столбец
vector = short_spectrum( : );

% расширенный нулями вектор
ext_vector = [ uint32( zeros( LEVELS, 1 ) ); vector; uint32( zeros( LEVELS, 1 ) ) ];

% минимальное значение расстояния, индекс и смещение
min_value = uint32( intmax ) * 2;
min_index = 0;
best_offset = 0;

% сдвиг
for offset = 1 : LEVELS * 2 + 1
    % выбираем рабочую часть
    work_vector = ext_vector( offset : offset + LEVELS - 1 );

    % копируем вектор до размера базы
    cloned_vector = kron( work_vector, uint32( ones( 1, K * L ) ) );

    % рассчитываем и сортируем расстояние по метрике L2
    delta = sum( ( short_spectrum_db - cloned_vector ).^2 );
    [ value, index ] = sort( delta );

    % выбираем
    if value( 1 ) < min_value
        min_value = value( 1 );
        min_index = index( 1 );
        best_offset = offset;
    end;
end;

% рисуем распознанное изображение и его гистограммы ниже исходного изображения
im_vector = image_db( :, min_index );
size_vector = size_db( :, min_index );
rec_image = reshape( im_vector, size_vector' );

if no_gui == 0
    figure( 2 );
    subplot( 2, 3, 4 );
    imshow( rec_image );

    figure( 2 );
    subplot( 2, 3, 5 );
    full_sp_vector = full_spectrum_db( :, min_index );
    plot( full_sp_vector' );

    figure( 2 );
    subplot( 2, 3, 6 );
    short_sp_vector = short_spectrum_db( :, min_index );
    bar( short_sp_vector' );
end;

% ищем класс распознанного изображения
class_new = fix( ( min_index - 1 ) / L ) + 1;
if class == class_new

```

```

        rec_rate = rec_rate + 1;
    end;

    % пересчитываем процент распознавания
    rec_percent = ( rec_rate / takt ) * 100;

    if no_gui == 0
        subplot( 2, 3, 4 );
        title( [ 'Recognition: ', num2str( rec_percent ) ] );
    end;

    % ждёмс
    if no_gui == 0
        pause( 0.04 );
    end;
end;

% вывод результата
rec_percent

% light.m
% освещение базы

% очистка
clear all; clc;

% параметры
DB_PATH = 'D:\LETI\9\kml\bases\orl_bmp\s'; % путь к базе
NEW_DB_PATH = 'D:\LETI\9\kml\bases\orl_bmp_test\s'; % путь к новой базе
K = 40; % количество классов изображений
Q = 10; % общее количество изображений в классе
COLOR_CHANGE = 50;

% построение баз
for class = 1 : K
    for file = 1 : Q
        image = imread( [DB_PATH, num2str( class ), '\', num2str( file ), '.bmp'] );

        if rem( file, 2 ) == 0 % освещаем только чётные изображения
            for i = 1 : size( image, 1 )
                for j = 1: size( image, 2 )
                    color = image( i, j ) + COLOR_CHANGE;
                    if color < 0
                        color = 0;
                    elseif color > 255
                        color = 255;
                    end;
                    image( i, j ) = color;
                end;
            end;
            % расчёт полной гистограммы, коррекция цвета
        end;

        imwrite( image, [NEW_DB_PATH, num2str( class ), '\', num2str( file ), '.bmp'] ); %
        запись полученной картинки в файл
    end;
end;

```

## Результаты

### Результаты работы на оригинальной базе ORL

Ориганальная база содержит 40 классов по 10 изображений, из которых было взято 1 изображение в базу, 9 тестовых изображений.

	классический метод	модифицированный метод
32 столбца	58,6111	55,0000
64 столбца	55,8333	53,3333
128 столбцов	54,1667	52,5000
256 столбцов	55,0000	52,2222

Оригинальный метод демонстрирует распознавание примерно **55%**, с отклонениями в пределах **3%** в зависимости от размерности используемой гистограммы. Модифицированный метод показывает распознавание на уровне **53%** с отклонениями в **2%**.

Мы наблюдаем небольшое снижение распознавания около **2%** за счёт неверной идентификации изображений с похожим характером гистограммы, но различным яркостным сдвигом.

*40 классов, 2 изображения в базе, 8 тестовых изображений*

	классический метод	модифицированный метод
32 столбца	66,5625	64,6875
64 столбца	63,1250	61,8750
128 столбцов	62,1875	60,9375
256 столбцов	63,1250	60,9375

Добавление второго тестового изображения ощутимо поднимает результативность классического метода (примерно до **64%** с отклонением в **3%**). Модифицированный метод снова уступает с показателем в **62%** и отклонением в **2%**.

## Результаты работы на модифицированной базе ORL

*40 классов, 1 изображение в базе, 9 тестовых изображений*

	классический метод	модифицированный метод
32 столбца	23,6111	54,1667
64 столбца	23,0556	52,7778
128 столбцов	22,5000	52,7778
256 столбцов	23,0556	52,0000

На модифицированной базе впервые проявляется преимущество модифицированного метода. В то время как классический метод на осветлённых изображениях продемонстрировал показатели **23±2%**, модифицированный смог удержать показатель на уровне **52±2%**, то есть обеспечил более чем 2-кратное преимущество. Результат объясним, если вспомнить, что в данном тесте изображение в базе только (оригинал), что делает почти невозможным для классического метода распознавание осветлённых чётных по номерам изображений.

*40 классов, 2 изображения в базе, 8 тестовых изображений*

	классический метод	модифицированный метод
32 столбца	52,5000	62,5000
64 столбца	51,5625	60,9375
128 столбцов	49,6875	61,2500
256 столбцов	50,9375	61,2500

Теперь в базу входят 2 изображения, один оригинал и одно осветлённое, что должно уравнивать шансы методов. Тем не менее, модифицированный метод с показателем **61±1%** демонстрирует практически 10-процентное преимущество над классическим с **51±2%** распознавания.

## Выводы

В данной работе нам удалось улучшить показатели распознавания гистограммного метода на базах, имеющих изображения с разной степенью яркости.

Особенно эффективна наша модификация метода при существенных ограничениях на базу (1 изображение), хотя и при нескольких изображениях в базе даёт ощутимый эффект.

В качестве минусов модификации следует отметить небольшое (порядка 2%) ухудшение по сравнению с оригинальным методом на базах, имеющих приблизительно одинаковый уровень яркости.

## Список литературы

1. Кухарев Г. А. Биометрические системы: Методы и средства идентификации личности человека. — СПб.: Политехника, 2001.
2. Журнал о безопасности бизнеса и личности «БДИ», №5 2004 г. ; №3 2005 г.