

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА**

Факультет информатики и систем управления

Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №5
по курсу «Структуры и алгоритмы обработки больших данных»
«Параметры Харалика»

Выполнил:
студент группы ИУ9-21М
Беляев А. В.

Проверил:
Магазов С. С.

Вариант 3

Москва 2019

Цель работы

Научиться классифицировать текстуры методами кластеризации.

1. Дополнить базу изображений 10 изображениями не встречающимися в базе данных не регулярных текстур.



2. Написать программу вычисления вектор параметров Харалика для всех изображений представленных в БД изображений (регулярные и не регулярные структуры).

Ниже приведен код из MathWorks, вычисляющий параметры Харалика

```
function [x] = haralickTextureFeatures(co0cMat, xFeatures)
```

```
% check input
if nargin == 1
    xFeatures = 1 : 14;
end

% check co0cMat for dimensions:
if ~ismatrix(co0cMat)
    error(['\co0cMatInput must be a two dimensional matrix, '...
        'dimensional was %s.',ndims(co0cMat']]);
end
% initialize x
x = zeros(14,1);
% normalize glcm
co0cMat = co0cMat./sum(co0cMat(:));

%% Some pre-calculation:
% columns and rows
if sum(xFeatures == 2) == 1 | ... % Contrast
    sum(xFeatures == 3) == 1 | ... % Correlation
    sum(xFeatures == 4) == 1 | ... % Variance
    sum(xFeatures == 5) == 1 | ... % Inverse Difference Moment
    sum(xFeatures == 6) == 1 | ... % Sum Average
    sum(xFeatures == 7) == 1 | ... % Sum Variance
    sum(xFeatures == 8) == 1 | ... % Sum Entropy
    sum(xFeatures == 10) == 1 | ...% Difference Variance
    sum(xFeatures == 11) == 1 | ...% Difference Entropy
    sum(xFeatures == 14) == 1 % Maximal Correlation Coefficient
sizeco0cMat = size(co0cMat);
[col, row] = meshgrid(1:sizeco0cMat(1),1:sizeco0cMat(2));
end

% average and standarddeviation
if sum(xFeatures == 3) == 1 | ... % correlation
    sum(xFeatures == 10) == 1 % difference variance
    rowMean = sum( row(:).*co0cMat(:) );
    colMean = sum( col(:).*co0cMat(:) );
    rowStd = sqrt( sum( (row(:)-rowMean).^2 .* co0cMat(:) ) );
    colStd = sqrt( sum( (col(:)-colMean).^2 .* co0cMat(:) ) );
end

% sum of rows p_y(i) und sum of columns p_x(j)
if sum(xFeatures == 12) == 1 |...% Information Measures of Correlation I
    sum(xFeatures == 13) == 1|... % Information Measures of Correlation II
    sum(xFeatures == 14) == 1 % Maximal Correlation Coefficient

    rowCo0cMat = sum(co0cMat,2); %sum of rows p_y(i)
    colCo0cMat = sum(co0cMat); %sum of columns p_x(i)
end

% p_x+y
if sum(xFeatures == 6)==1 |... % Sum Average
    sum(xFeatures == 7)==1 |... % Sum Variance
    sum(xFeatures == 8)==1 % Sum Entropy

    start = -(sizeco0cMat(1) -1);
    stop = sizeco0cMat(1) -1;

    % Rotate Matrix 90°
    co0cMat90 = rot90(co0cMat);
```

```

% Initialize p_x+y
p_XplusY = zeros((2*sizeco0cMat(1))-1,1);

k = 1;
for index = start : stop
    p_XplusY(k) = sum( diag(co0cMat90,index) );
    k = k + 1;
end

% Initialize p_x-y
if sum(xFeatures == 10)==1 | ... % Difference Variance
    sum(xFeatures == 11)==1 % Difference Entropy

    start = 1;
    stop = sizeco0cMat(1)-1;

    % Initialize p_XminusY
    p_XminusY = zeros(sizeco0cMat(1),1);
    p_XminusY(1) = sum (diag(co0cMat,0) );

    k = 2;
    for index = start : stop
        p_XminusY(k) = sum( [diag(co0cMat,index);
                             diag(co0cMat,-index)] );
        k = k + 1;
    end
end

%% Haralick Feature Calculations
for f = xFeatures
    switch f
        case 1 % Energy (Angular Second Moment)
            x(1) = sum( co0cMat(:).^2 );

        case 2 % Contrast
            matrix = ( abs(row - col).^2 ) .* co0cMat;
            x(2) = sum( matrix(:) );

        case 3 % Correlation
            zaehler = sum ((row(:) - rowMean) .* ...
                           (col(:) - colMean) .* co0cMat(:));
            denominator = rowStd * colStd;
            x(3) = zaehler/denominator;

        case 4 % Variance
            x(4) = sum( (row(:)-mean(co0cMat(:))).^2 .*co0cMat(:) );

        case 5 % Inverse Difference Moment
            x(5) = sum( co0cMat(:) ./ ( 1+ (row(:)-col(:)).^2 ) );
            case 6 % Sum Average
                x(6) = sum( (2:(2*sizeco0cMat(1)))' .* p_XplusY );

        case 7 % Sum Variance
            x(8) = - sum( p_XplusY(p_XplusY~=0) .* ...
                           log(p_XplusY(p_XplusY~=0)) );
            x(7) = sum( ((2:(2*sizeco0cMat(1)))' - ...
                           x(8)).^2 .* p_XplusY );

        case 8 % Sum Entropy
            if ~x(8) % if it is not calculate in case 7
                x(8) = - sum( p_XplusY(p_XplusY~=0) .* ...
                               log(p_XplusY(p_XplusY~=0)) );
            end
    end

```

```

case 9 % Entropy
    x(9) = - sum( co0cMat(co0cMat~=0) .*...
        log2(co0cMat(co0cMat~=0)) );

case 10 % Difference Variance
    x(10) = sum( ((0:sizeco0cMat(1)-1)' -...
        mean(p_XminusY)).^2 .* p_XminusY);

case 11 % Difference Entropy
    x(11) = - sum( p_XminusY(p_XminusY~=0) .*...
        log(p_XminusY(p_XminusY~=0)) );

case 12 % Information Measures of Correlation I

    x(9) = - sum( co0cMat(co0cMat~=0) .*...
        log2(co0cMat(co0cMat~=0)) );

    % Cut out all zeros:
    logrc = log2( rowCo0cMat*colCo0cMat ); % 256x1 * 1x256
    %Matrixmultiplication
    logrc(logrc == -Inf) = 0; % cut out Inf
    HXY1 = - sum( co0cMat(:).* logrc(:) ); %product of elements
    % between co-occurrence-matrix and the logarithmic matrix
    numerator = x(9) - HXY1;

    % calculate off HX, Entropy of sum of columns
    logc = log2(colCo0cMat);
    logc(logc== -Inf) = 0;
    HX = - sum( colCo0cMat .* logc );

    % calculate off HY, Entropy of sum of columns
    logr = log2( rowCo0cMat );
    logr(logr== -Inf) = 0;
    HY = - sum( rowCo0cMat .* logr );

    % max value
    denominator = max( [HX HY] );
    x(12) = numerator / denominator;

case 13 % Information Measures of Correlation II
    if x(9)
        x(9) = - sum( co0cMat(co0cMat~=0) .*...
            log2(co0cMat(co0cMat~=0)) );
    end
    logrc = log2( rowCo0cMat*colCo0cMat ); % 256x1 * 1x256
    %Matrixmultiplication
    logrc(logrc == -Inf) = 0;
    HXY2 = - sum( sum( (rowCo0cMat * colCo0cMat) .* logrc ) );
    x(13) = ( ( 1 - exp(-2*(HXY2 - x(9))) ) ).^(1/2);

case 14 % Maximal Correlation Coefficient

    % Initialise Q
    Q = zeros(sizeco0cMat(1),sizeco0cMat(2));

    % % Slow version
    %for i = 1 : length(co0cMat(:,1))
    %    for j = 1 : length(co0cMat(1,:))
    %        Q(i,j) = sum(( co0cMat(i,:) .* co0cMat(j,:) )....
    %            ./ ( rowCo0cMat(i) .* colCo0cMat ),'omitnan');
    %    end
    %end

```

```

        for i = 1 : sizeco0cMat(2)
            Q(i,:) = sum( ...
                (repmat(co0cMat(i,:),sizeco0cMat(1),1) .* co0cMat ) ./ ...
                repmat( rowCo0cMat(i) .* colCo0cMat, sizeco0cMat(1),1));
        end
    % cut out nans
    Q(isnan(Q)) = 0;
    eigenvec = eig(Q);
    % Find largest eigenvec and delete
    eigenvec(eigenvec==max(eigenvec))=[];

    % If eigenvec is a zero matrix, no maximum can be found, that
    % would mean, that the second largest eigenvec of course is
    % also 0
    if ~any(eigenvec) || isempty(eigenvec)
        x(14) = sqrt(0);
        continue;
    end
    % Sqrt of second largest eigenvec
    x(14) = sqrt( max(eigenvec) );
    % calculate magnitude of Maximal Correlation Coefficient
    if imag(x(14))
        x(14) = abs(x(14));
    end
end
end

```

Программа для запуска расчетов

```

files = dir('*.png');
IMAGE_NUM = numel(files);
HARALICK_FEAT_NUM = 14;

haralicks = zeros(IMAGE_NUM, HARALICK_FEAT_NUM);
i = 1;
for file = files'
    S=imread(file.name);
    S=rgb2gray(S);
    I= imresize (S, [350 350]);
    glcm=graycomatrix(I,'offset',[-1 1], 'NumLevel', 8, 'Symmetric',true);
    xFeatures = 1:HARALICK_FEAT_NUM;

    j = 1;
    feats = haralickTextureFeatures(glcm, xFeatures);
    while j <= HARALICK_FEAT_NUM
        haralicks(i, j) = feats(j);
        j = j + 1;
    end
    i = i + 1;
end
NUM_CLUSTERS = 3;
%% K-means
% clusterized = kmeans(haralicks, NUM_CLUSTERS);

%% Hierachial
Y = pdist(haralicks);
Z = linkage(Y);
T = cluster(Z, 'maxclust', NUM_CLUSTERS);

disp(T);

```

К 10 изображениям нерегулярных текстур (выше) были добавлены 10 регулярных. Затем была выполнена кластеризация.

Результаты работы программы

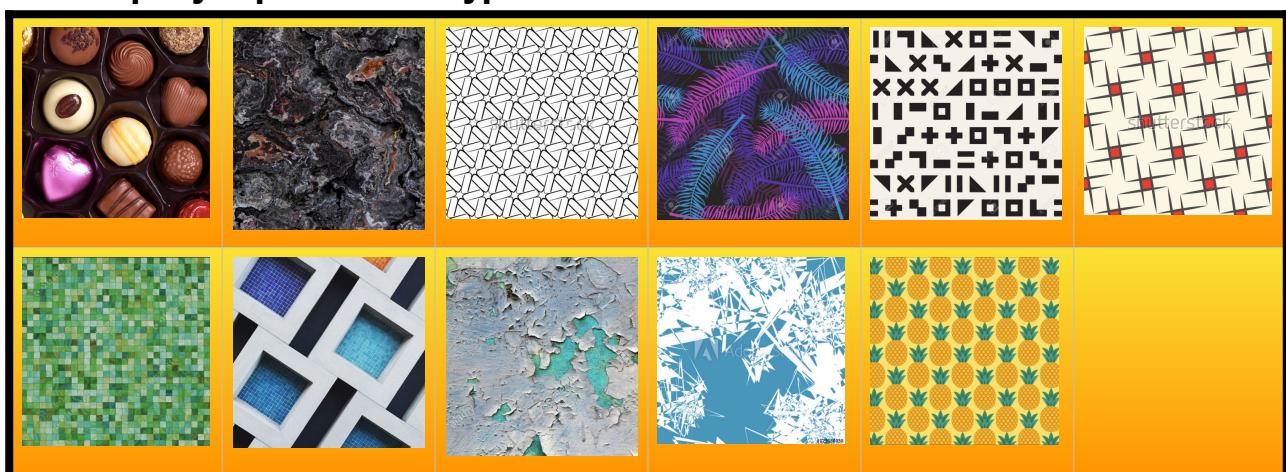
В качестве предполагаемых результатов ожидаем кластеры, сгруппированные по следующим, наблюдаемым визуально «признакам»

- Регулярность, гранулярность
- Интенсивность, плотность, однородность
- Яркость
- Конtrаст

K-means

K = 2

Более регулярные текстуры

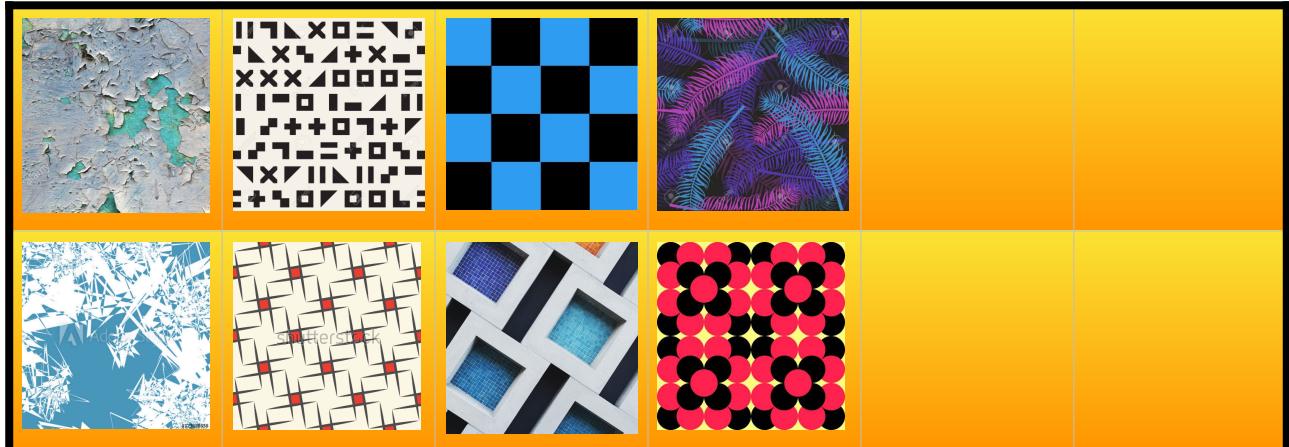


Менее регулярные текстуры

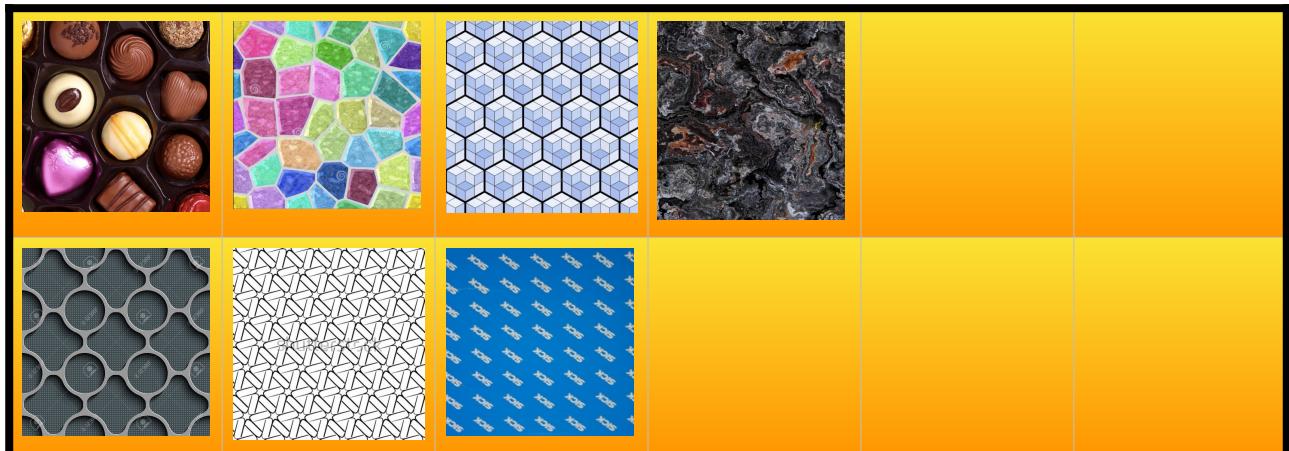


K = 3

Наименее плотные текстуры



Средние по плотности текстуры



Наиболее плотные текстуры

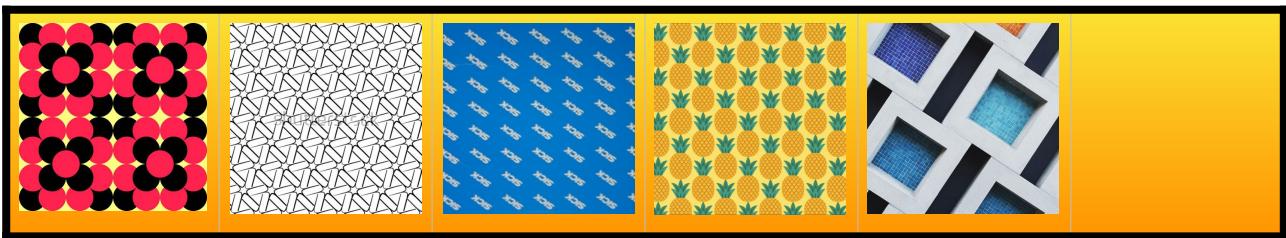


K = 4

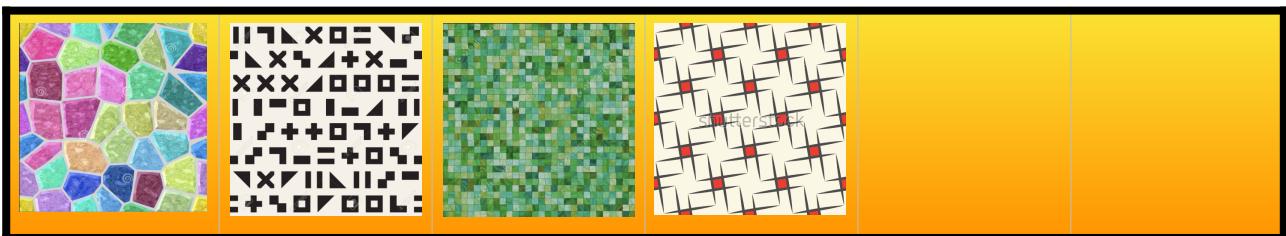
Наиболее темные



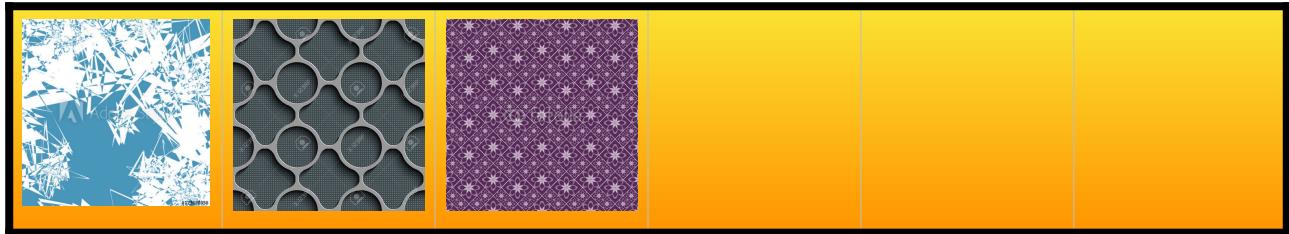
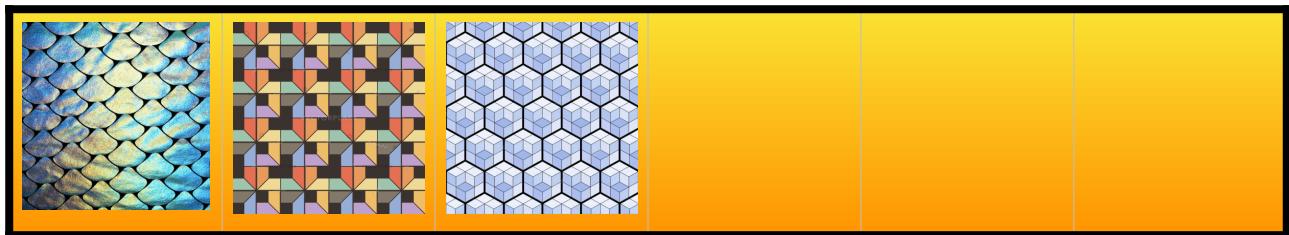
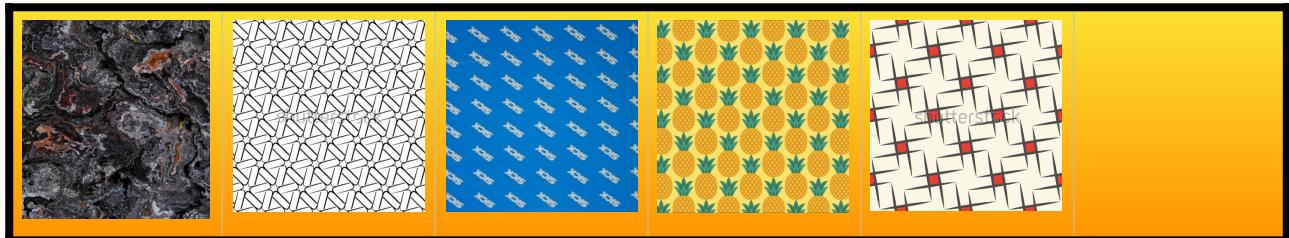
Средние



Наиболее светлые



K = 5

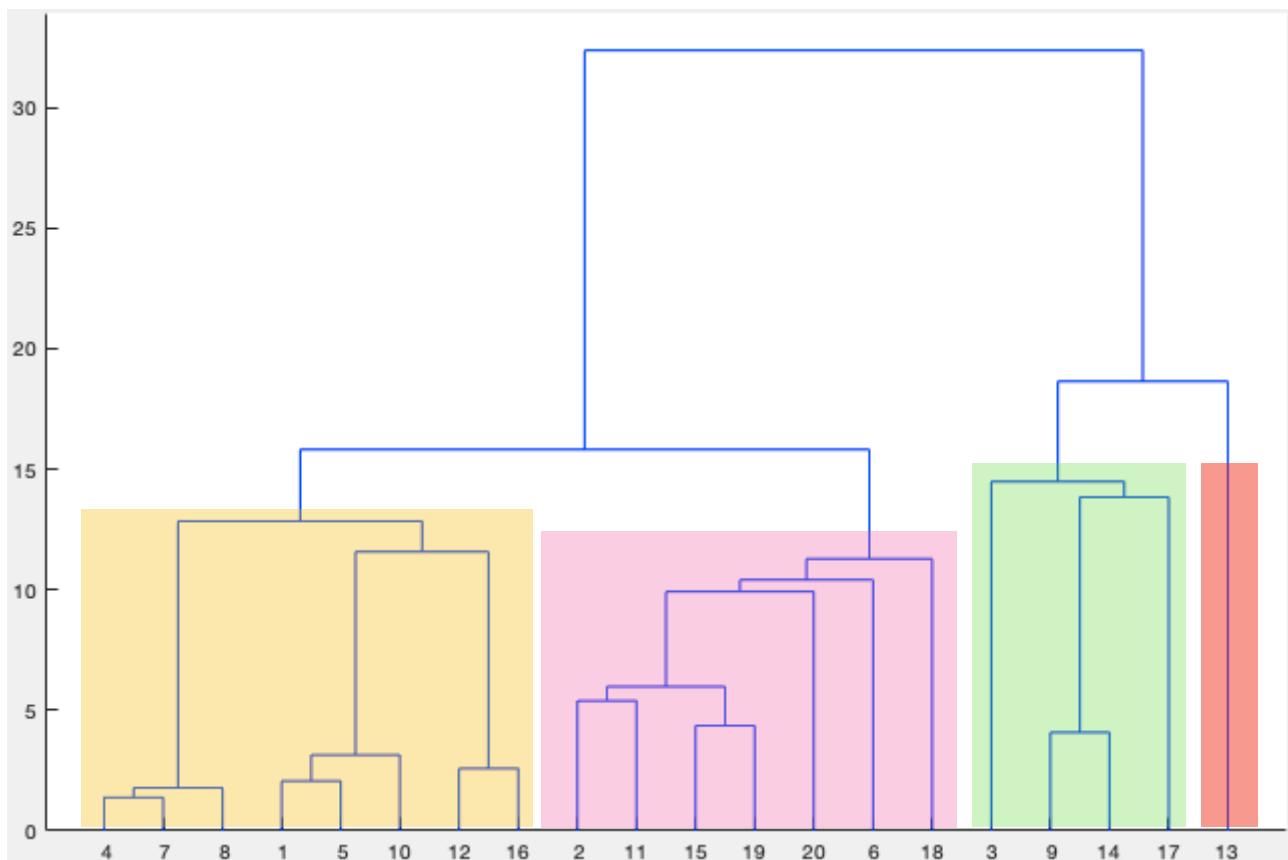


При разделении на 5 кластеров сложнее выделить закономерность

Иерархическая кластеризация

```
NUM_CLUSTERS = 3;  
  
%% Hierachial  
Y = pdist(haralicks);  
dendrogram(Z);  
  
Z = linkage(Y);  
T = cluster(Z, 'maxclust', NUM_CLUSTERS);  
  
disp(T);
```

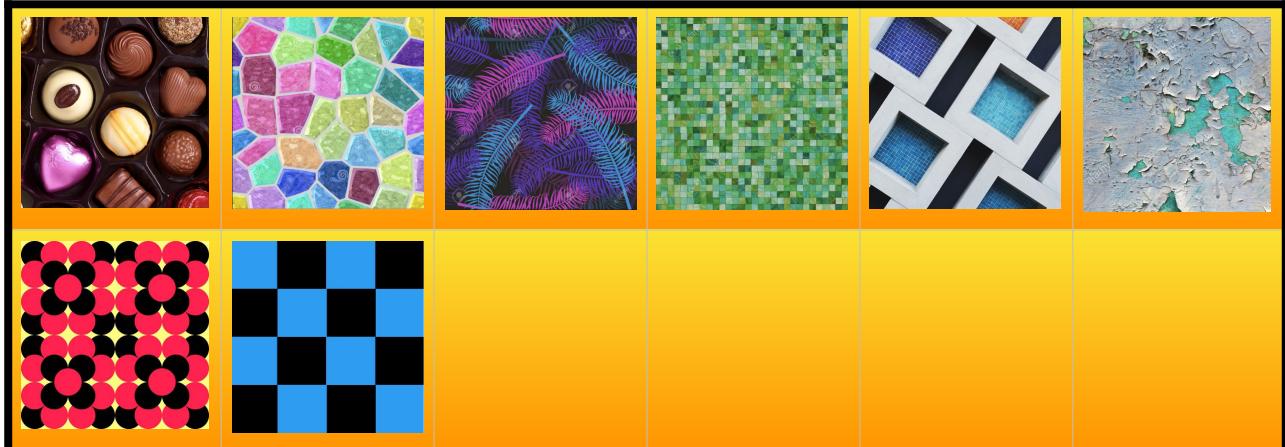
Дендрограмма:



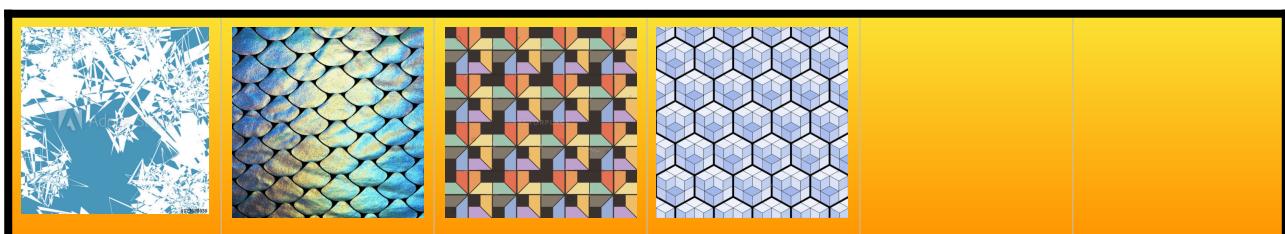
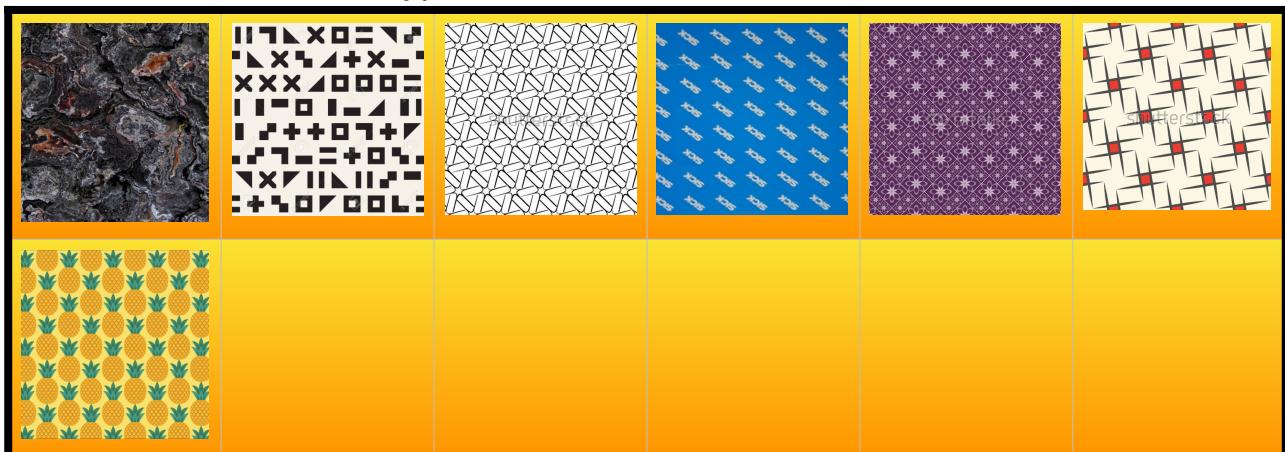
Далее попробуем разбить текстуры на 4 кластера

K = 4

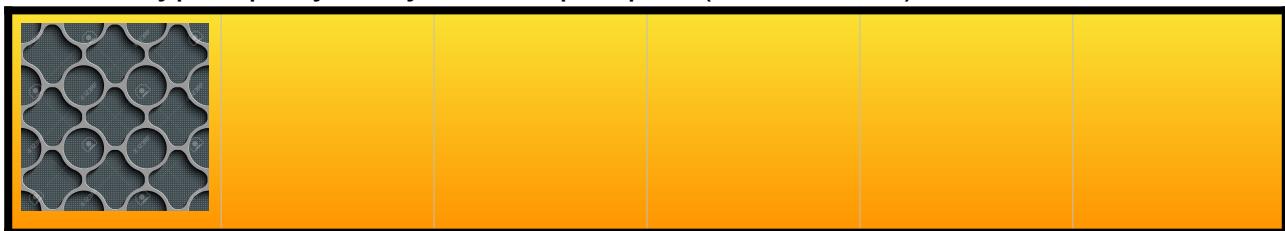
Средние по гранулярности текстуры



Мелкие плотные текстуры



На текстуре присутствуют водяные знаки (watermarks)



На дендрограмме выше прозрачными прямоугольниками обозначены получившиеся кластеры в масштабах всей дендрограммы.

Заключение

С изменением количества кластеров на приведенном наборе картинок интуитивность восприятия меняется.

Так,

- разбиение на 3 кластера может соответствовать интуитивному
- Разбиение на 4 может быть континтуитивно
- Разбиение на 5 вновь интуитивно

Лабораторная работа подтвердила то, что человеческое восприятие текстур может не совпадать с «машинным», т.к. Алгоритм может «подмечать» невидимые человеком статистические признаки изображения, которые не видит человек. И наоборот.