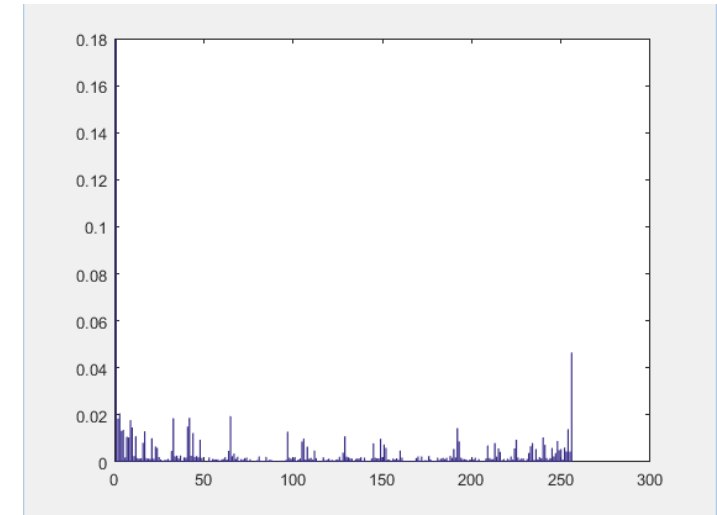


Lab 9

CPS592 – Visual Computing and Mixed Reality

Objective

- Compute local binary patterns (LBP) feature



Preparation

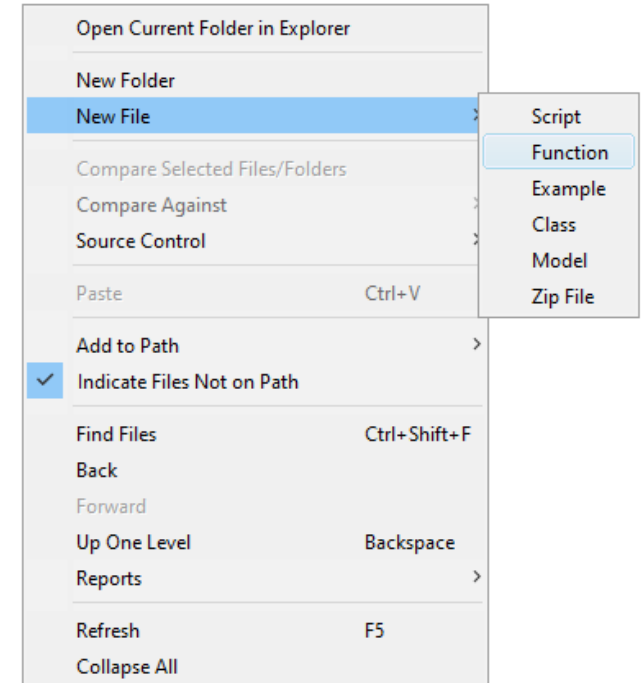
- Open MATLAB
- Create Lab9 folder
- Copy *ud1.jpg*, *ud2.jpg*, *tower.jpg* to Lab9 folder

Create lbp function

```
function feat = lbp(img)
```

```
%LBP This function is used to extract local binary  
% patterns from the input image
```

```
end
```



Convert the input image to grayscale

```
function feat = lbp(img)
```

```
img_gray = rgb2gray(img);
```

```
end
```

Get the image's size and initiate the returned feature

```
function feat = lbp(img)
```

```
img_gray = rgb2gray(img);
```

```
[height, width] = size(img_gray);
```

```
feat = zeros(1,256);
```

```
end
```

Check every single pixel

```
function feat = lbp(img)
img_gray = rgb2gray(img);
[height, width] = size(img_gray);
feat = zeros(1,256);
```

```
    for i = 2:height - 1
        for j = 2:width - 1

            end

        end
    end
```

```
end
```

Get the neighboring pixels

```
function feat = lbp(img)
img_gray = rgb2gray(img);
[height, width] = size(img_gray);
feat = zeros(1,256);
for i = 2:height - 1
    for j = 2:width - 1
        neighbors = img_gray(i-1:i+1,j-1:j+1);
        bits = double(neighbors(:));
    end
end
end
```


Get the threshold values

```
function feat = lbp(img)
img_gray = rgb2gray(img);
[height, width] = size(img_gray);
feat = zeros(1,256);
for i = 2:height - 1
    for j = 2:width - 1
        neighbors = img_gray(i-1:i+1,j-1:j+1);
        bits = double(neighbors(:));
        threshold = bits(5);
        bits(5) = [];
    end
end
end
```

Thresholding

```
function feat = lbp(img)
img_gray = rgb2gray(img);
[height, width] = size(img_gray);
feat = zeros(1,256);
for i = 2:height - 1
    for j = 2:width - 1
        neighbors = img_gray(i-1:i+1,j-1:j+1);
        bits = double(neighbors(:));
        threshold = bits(5);
        bits(5) = [];
        bits = bits - threshold;
        bits = sign(bits);
        bits(bits < 0) = 0;
    end
end
end
```

Converting LBP bits to byte

```
function feat = lbp(img)
img_gray = rgb2gray(img);
[height, width] = size(img_gray);
feat = zeros(1,256);
for i = 2:height - 1
    for j = 2:width - 1
        neighbors = img_gray(i-1:i+1,j-1:j+1);
        bits = double(neighbors(:));
        threshold = bits(5);
        bits(5) = [];
        bits = bits - threshold;
        bits = sign(bits);
        bits(bits < 0) = 0;
        byte = sum(bits.*2.^(length(bits)-1 : -1 : 0)');
    end
end
end
```

Updating the returned feature

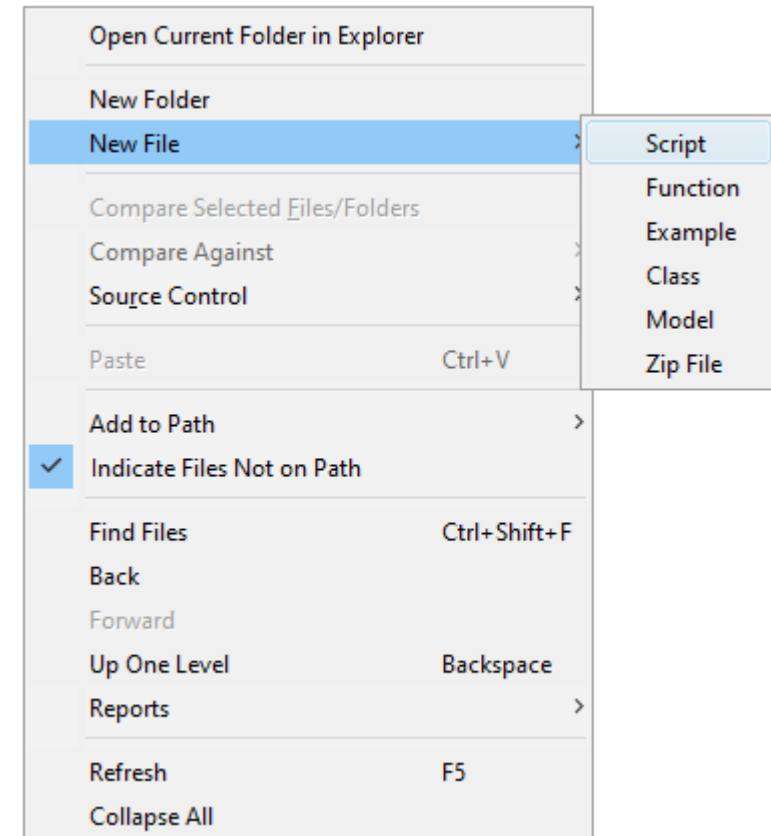
```
function feat = lbp(img)
img_gray = rgb2gray(img);
[height, width] = size(img_gray);
feat = zeros(1,256);
for i = 2:height - 1
    for j = 2:width - 1
        neighbors = img_gray(i-1:i+1,j-1:j+1);
        bits = double(neighbors(:));
        threshold = bits(5);
        bits(5) = [];
        bits = bits - threshold;
        bits = sign(bits);
        bits(bits < 0) = 0;
        byte = sum(bits.*2.^(length(bits)-1 : -1 : 0)');
        feat(byte + 1) = feat(byte + 1) + 1;
    end
end
end
```

Histogram normalization

```
function feat = lbp(img)
img_gray = rgb2gray(img);
[height, width] = size(img_gray);
feat = zeros(1,256);
for i = 2:height - 1
    for j = 2:width - 1
        neighbors = img_gray(i-1:i+1,j-1:j+1);
        bits = double(neighbors(:));
        threshold = bits(5);
        bits(5) = [];
        bits = bits - threshold;
        bits = sign(bits);
        bits(bits < 0) = 0;
        byte = sum(bits.*2.^(length(bits)-1 : -1 : 0)');
        feat(byte + 1) = feat(byte + 1) + 1;
    end
end
feat = feat./sum(feat);
end
```

Create new script “Lab9.m”

```
close all;  
clear all;  
clc;
```



Read the input image and compute its lbp feature

close all;

clear all;

clc;

```
img = imread('ud1.jpg');
```

```
feat = lbp(img);
```

Display the LBP feature

```
close all;
```

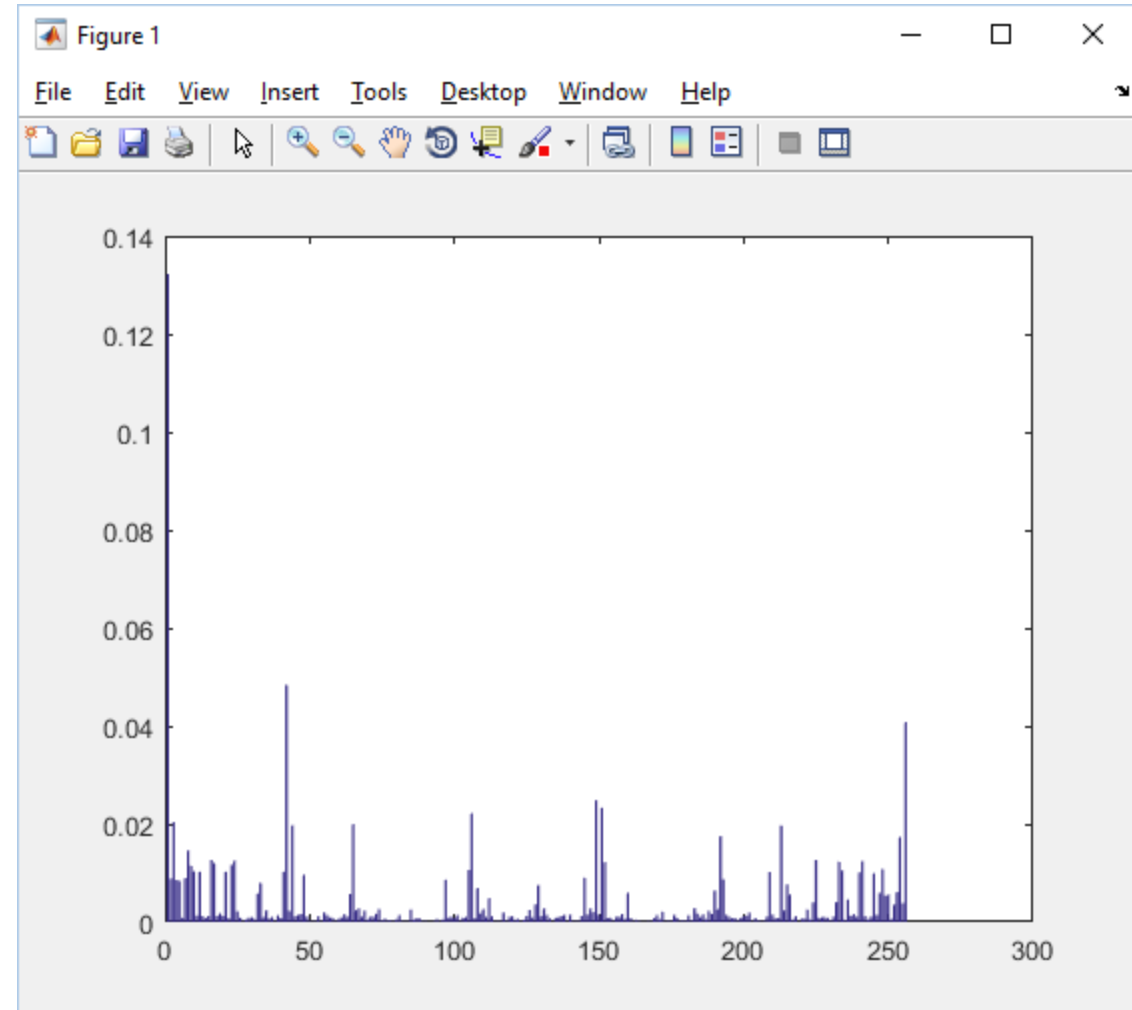
```
clear all;
```

```
clc;
```

```
img = imread('ud1.jpg');
```

```
feat = lbp(img);
```

```
figure, bar(feat);
```

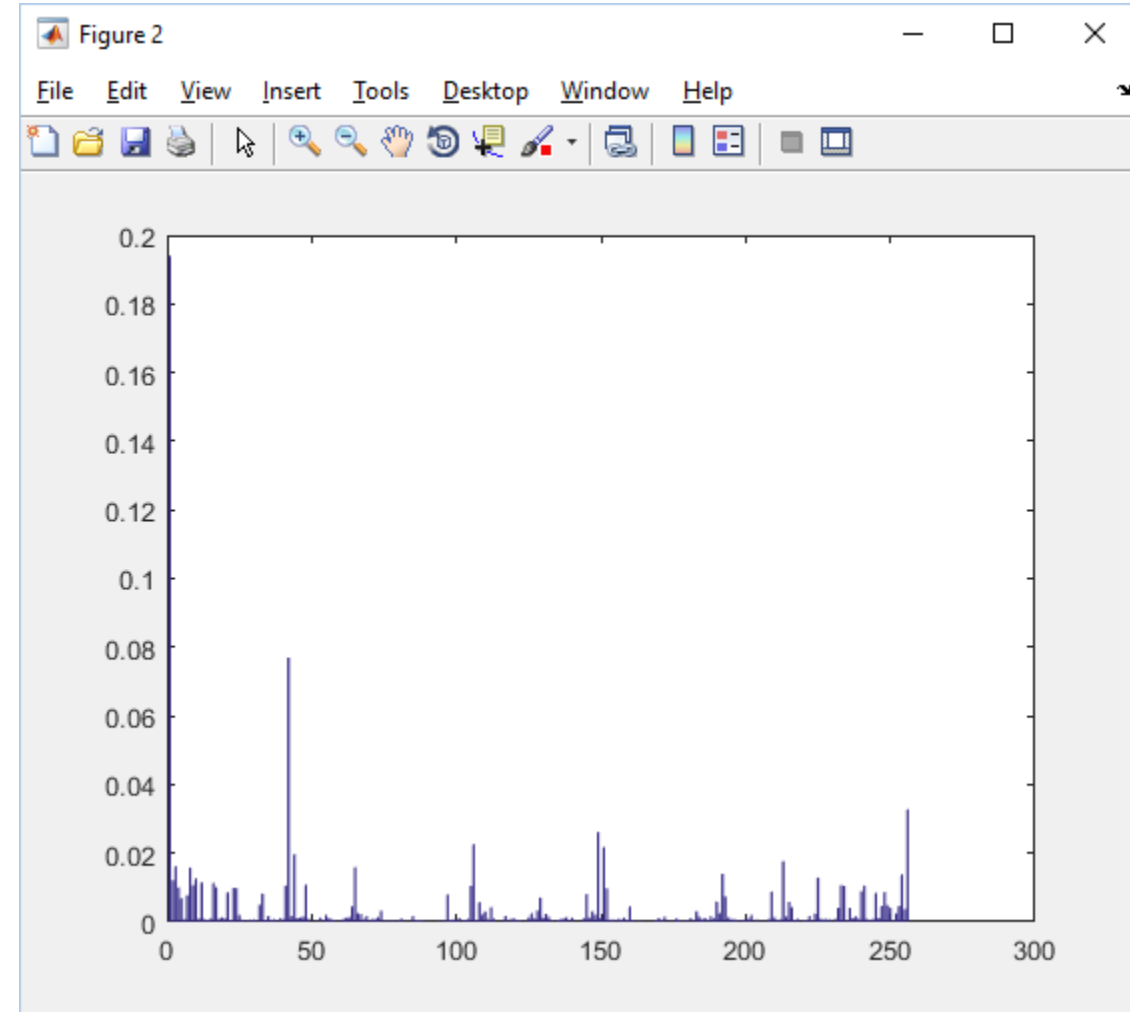


Try with another image

```
close all;  
clear all;  
clc;
```

```
img = imread('ud1.jpg');  
feat = lbp(img);  
figure, bar(feat);
```

```
img = imread('ud2.jpg');  
feat = lbp(img);  
figure, bar(feat);
```



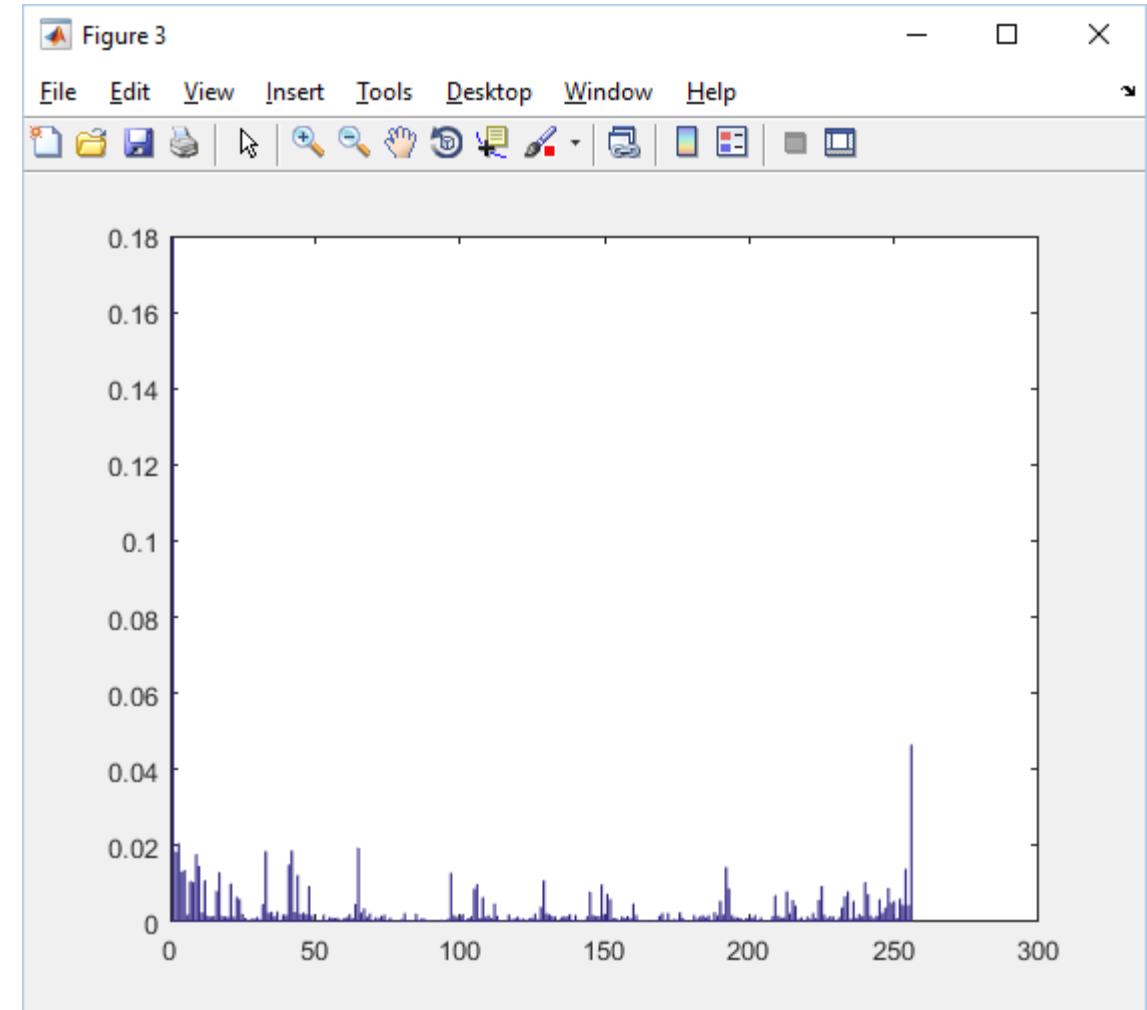
Try with another image

```
close all;  
clear all;  
clc;
```

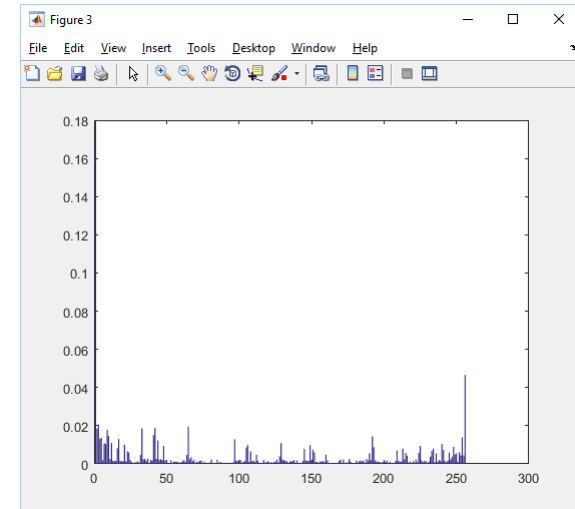
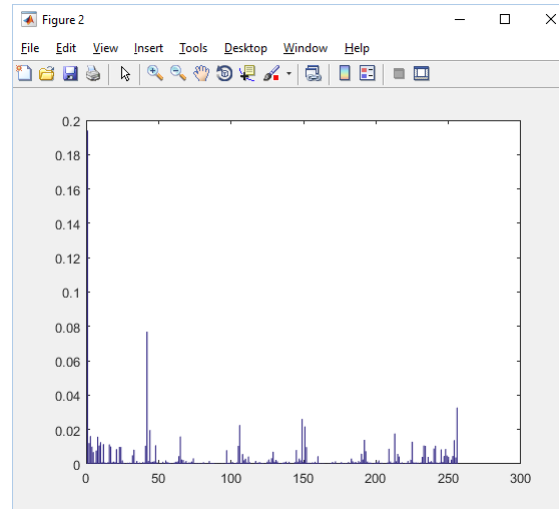
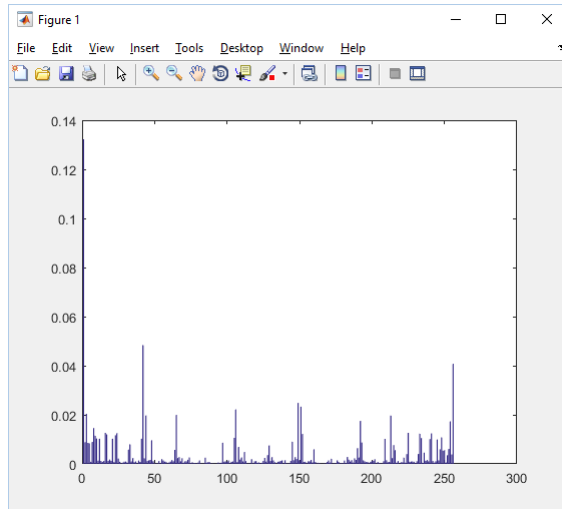
```
img = imread('ud1.jpg');  
feat = lbp(img);  
figure, bar(feat);
```

```
img = imread('ud2.jpg');  
feat = lbp(img);  
figure, bar(feat);
```

```
img = imread('tower.jpg');  
feat = lbp(img);  
figure, bar(feat);
```



Compare the LBP histograms



Q&A