
halftone.m

```
function out = halftone(in)
% halftone The function converts a greyscale image into a binary
image
%           using dot patterns.
%
% Syntax:
%   out = halftone(in)
%
% Input:
%   in = the grayscale image to be converted, and should be of type
uint8
%   with values in the range 0-255.
%
% Output:
%   out = the binary image that will be outputted. It is of type uint8
and will consist of two values: 0 and
%       255.
% History:
%   J.Luciano - Created function halftone 2/21/2022
%   J.Luciano - Attempted to add edge cases 2/23/2022
%

%create out with size of in
[r c] = size(in);
out = uint8(ones(r, c));
%init dot patterns, 1 to 10
dots = uint8(zeros(3, 3, 10));

dots(:, :, 1) = [0 0 0; 0 0 0; 0 0 0];
dots(:, :, 2) = [0 255 0; 0 0 0; 0 0 0];
dots(:, :, 3) = [0 255 0; 0 0 0; 0 0 255];
dots(:, :, 4) = [255 255 0; 0 0 0; 0 0 255];
dots(:, :, 5) = [255 255 0; 0 0 0; 255 0 255];
dots(:, :, 6) = [255 255 255; 0 0 0; 255 0 255];
dots(:, :, 7) = [255 255 255; 0 0 255; 255 0 255];
dots(:, :, 8) = [255 255 255; 0 0 255; 255 255 255];
dots(:, :, 9) = [255 255 255; 255 0 255; 255 255 255];
dots(:, :, 10) = [255 255 255; 255 255 255; 255 255 255];

in_float = double(in);

%check edges with mod
c_edge = mod(r, 3);
r_edge = mod(c, 3);

%loop whole input image 3x3 pixels -> whats the dot pattern we want in
%the out-> change the 3x3 pixels of output with the dot pattern
for r1 = 1:3:r-r_edge
    for c1 = 1:3:c-c_edge
        average = mean(mean(in_float(r1:r1+2,c1:c1+2)));
```

```

        %find the dot pattern(:, :, i)
        i = floor((average/255)*10);
        %set i = 9 because we do i+1 to avoid cases where i = 0
        if i == 10
            i = 9;
        end
        out(r1:r1+2, c1:c1+2) = dots(:, :, i+1);
    end
    if c_edge > 0
        %use the remainder c to calculate the average of a 3x1 or 3x2
    block
        average = mean(mean(in_float(r1:r1+2, c-c_edge+1:c)));
        i = floor((average/255)*10);
        if i == 10
            i = 9;
        end
        out(r1:r1+2, c-c_edge+1:c) = dots(:, 1:c_edge, i+1);
    end
    if r_edge > 0
        %use the remainder r to calculate the average of a 1x3 or 2x3
    block
        average = mean(mean(in_float(r-r_edge+1:r, c1:c1+2)));
        %function floor() rounds the value to the nearest integer
        i = floor((average/255)*10);
        if i == 10
            i = 9;
        end
        out(r-r_edge+1:r, c1:c1+2) = dots(1:r_edge, :, i+1);
    end
end
end

```

Published with MATLAB® R2020b