

## **PROJECT CODE**

% Demo to watermark an image by hiding another image in a certain bit  
% plane. Sometimes called "LSB Watermarking" or something similar.  
% User is asked which bit plane they want to hide the image in.

```
clc; % Clear the command window.  
close all; % Close all figures (except those of imtool.)  
imtool close all; % Close all imtool figures.  
clear; % Erase all existing variables.  
workspace; % Make sure the workspace panel is showing.  
fontSize = 12;
```

% Read in the image what will have another image hidden into it.

```
baseFileName='navy.jpg';  
% baseFileName='cameraman.tif';  
folder = fullfile(matlabroot, '\toolbox\images\imdemos');  
% Get the full filename, with path prepended.  
fullFileName = fullfile(folder, baseFileName);  
if ~exist(fullFileName, 'file')  
% Didn't find it there. Check the search path for it.  
fullFileName = baseFileName; % No path this time.  
if ~exist(fullFileName, 'file')  
% Still didn't find it. Alert user.  
errorMessage = sprintf('Error: %s does not exist.', fullFileName);  
uiwait(warndlg(errorMessage));  
return;  
end  
end  
originalImage = imread(fullFileName);
```

% Get the number of rows and columns in the original image.

```
[visibleRows, visibleColumns, numberOfColorChannels] = size(originalImage);  
if numberOfColorChannels > 1  
% If it's color, extract the red channel.  
originalImage = originalImage(:, :, 1);  
end
```

```

% Display the original gray scale image.
subplot(3, 3, 4);
imshow(originalImage, []);
title('Original Grayscale Starting Image', 'FontSize', fontSize);
% Enlarge figure to full screen.
set(gcf, 'units','normalized','outerposition',[0 0 1 1]);
set(gcf,'name','Demo by ImageAnalyst','numbertitle','off')

% read the message image you want to hide in the cover image
baseFileName='airforcesymbol.jpeg';
% baseFileName='moon.tif';
% Get the full filename, with path prepended.
fullFileName = fullfile(folder, baseFileName);
if ~exist(fullFileName, 'file')
% Didn't find it there. Check the search path for it.
fullFileName = baseFileName; % No path this time.
if ~exist(fullFileName, 'file')
% Still didn't find it. Alert user.
errorMessage = sprintf('Error: %s does not exist.', fullFileName);
uiwait(warndlg(errorMessage));
return;
end
end
hiddenImage = imread(fullFileName);
% Get the number of rows and columns in the hidden image.
[hiddenRows, hiddenColumns, numberOfColorChannels] = size(hiddenImage);
if numberOfColorChannels > 1
% If it's color, extract the red channel.
hiddenImage = hiddenImage(:,:,1);
end
% Display the image.
subplot(3, 3, 1);
imshow(hiddenImage, []);
title('Image to be Hidden', 'FontSize', fontSize);

% Let's compute and display the histogram.
[pixelCount, grayLevels] = imhist(hiddenImage);
subplot(3, 3, 2);
bar(pixelCount);
title('Histogram of image to be hidden', 'FontSize', fontSize);

```

```

xlim([0 grayLevels(end)]); % Scale x axis manually.
grid on;

thresholdValue =200;
binaryImage = hiddenImage < thresholdValue;
% Display the image.
subplot(3, 3, 3);
imshow(binaryImage, []);
caption = sprintf('Hidden Image Thresholded at %d', thresholdValue);
title(caption, 'FontSize', fontSize);

% Get the bit plane to hide the image in.
prompt = 'Enter the bit plane you want to hide the image in (1 - 8) ';
dialogTitle = 'Enter Bit Plane to Replace';
numberOfLines = 1;
defaultResponse = {'6'};
bitToSet = str2double(cell2mat(inputdlg(prompt, dialogTitle, numberOfLines,
defaultResponse)));

% If image to be hidden is bigger than the original image, scale it down.
if hiddenRows > visibleRows || hiddenColumns > visibleColumns
amountToShrink = min([visibleRows / hiddenRows, visibleColumns / hiddenColumns]);
binaryImage = imresize(binaryImage, amountToShrink);
% Need to update the number of rows and columns.
[hiddenRows, hiddenColumns] = size(binaryImage);
end
% Tile the hiddenImage, if it's smaller, so that it will cover the original image.
if hiddenRows < visibleRows || hiddenColumns < visibleColumns
watermark = zeros(size(originalImage), 'uint8');
for column = 1:visibleColumns
for row = 1:visibleRows
watermark(row, column) = binaryImage(mod(row,hiddenRows)+1,
mod(column,hiddenColumns)+1);
end
end
% Crop it to the same size as the original image.
watermark = watermark(1:visibleRows, 1:visibleColumns);
else
% Watermark is the same size as the original image.
watermark = binaryImage;

```

```

end
% Display the thresholded binary image - the watermark alone.
subplot(3, 3, 5);
imshow(watermark, []);
caption = sprintf('Hidden Image\nto be Inserted into Bit Plane %d', bitToSet);
title(caption, 'FontSize', fontSize);

% Set the bit of originalImage(a copy, actually) to the value of the watermark.
watermarkedImage = originalImage; % Initialize
for column = 1 : visibleColumns
    for row = 1 : visibleRows
        watermarkedImage(row, column) = bitset(originalImage(row, column), bitToSet,
        watermark(row, column));
    end
end

% Display the image.
subplot(3, 3, 6);
imshow(watermarkedImage, []);
caption = sprintf('Final Watermarked Image\nwithout added Noise');
title(caption, 'FontSize', fontSize);

% add noise to watermarked image
noisyWatermarkedImage = imnoise(watermarkedImage, 'gaussian', 0, 0.0005);

% Display the image.
subplot(3, 3, 7);
imshow(noisyWatermarkedImage, []);
caption = sprintf('Watermarked Image\nwith added Noise');
title(caption, 'FontSize', fontSize);
% Now let's pretend we are starting with the watermarked noisy corrupted image.
% We want to recover the watermark.

% Use the known bitplane of watermarked image to recover the watermark.
recoveredWatermark = zeros(size(noisyWatermarkedImage));
recoveredNoisyWatermark = zeros(size(noisyWatermarkedImage));
for column = 1:visibleColumns
    for row = 1:visibleRows
        recoveredWatermark(row, column) = bitget(watermarkedImage(row, column), bitToSet);
        recoveredNoisyWatermark(row, column) = bitget(noisyWatermarkedImage(row, column),

```

```

bitToSet);
end
end

% Scale the recovered watermark to 0=255
recoveredWatermark = uint8(255 * recoveredWatermark);
recoveredNoisyWatermark = uint8(255 * recoveredNoisyWatermark);

% Display the images.
subplot(3, 3, 8);
imshow(recoveredWatermark, []);
caption = sprintf('Watermark Recovered\nfrom Bit Plane %d of\nNoise-Free Watermarked
Image', bitToSet);
title(caption, 'FontSize', fontSize);
% Display the images.
subplot(3, 3, 9);
imshow(recoveredNoisyWatermark, []);
caption = sprintf('Watermark Recovered\nfrom Bit Plane %d of\nNoisy Watermarked Image',
bitToSet);
title(caption, 'FontSize', fontSize);
msgbox('Done with demo!');

```