Matt Washburn

Kristine Nutter

Brandon Bench

Nick DeMarco

Delivery Report

When creating the hybrid image of two different images using the magnitude and phase components of the images, you must first take the fast Fourier transform of the two images. You then have to isolate the magnitude and phase of the images by using the **abs()** function (to isolate magnitude) and **angle()** function (to isolate the phase). Once you get these different components, you then have to recompute the frequency, by using the formula: **output1 = mag1 .* exp(1i*phase2)**, using the magnitude component of one image and the phase of the other. You do this for both images. You then take the inverse of the output from the above formula and that gives you the hybrid image of the two images.

When neutralizing the magnitude, we extract the phase after applying the fast Fourier transform on an image. To do this, we use **exp(1i*angle(im1_fft))** the '**angle()**' function that returns the phase angles, in radians for each element of the object. In order to restore the image with just the phase, apply the inverse fast Fourier transform after the phase angles have been calculated.

To remove the phase, first use fast Fourier transform, then you need to use the **fftshift**. Use '**abs()**' to get the complex magnitude of the image. You then need to take the inverse of this complex magnitude by using the formula: **log(abs(ifft2(im1_M*exp(1i*0)))+1)**. This will give you the restored image, which you then need to **fftshift** again. Once this is complete, you must then calculate the plotting limits to show the phase removed image. You do this by using these formulas: **I_Mag_min = min(min(abs(restoredP1))) and I_Mag_max = max(max(abs(restoredP1)))**. Once this is complete, you can use those max and mins to show the phase removed images.

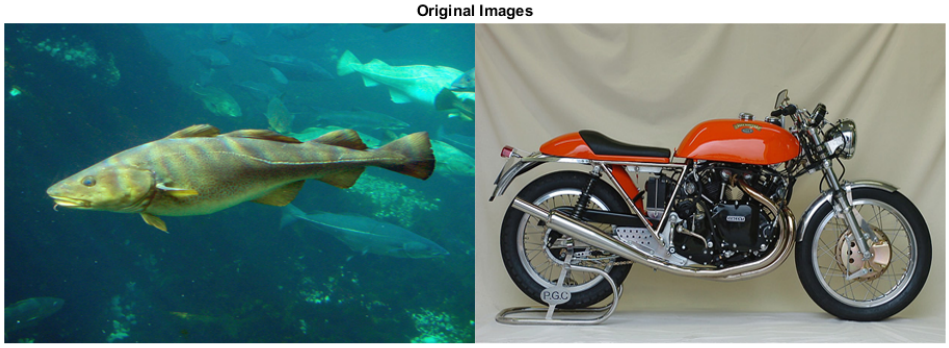
Table of Contents


```
% Author: Nick DeMarco
% Hybrid Image

clc;
clear;
close all; % closes all figures
```

Setup

```
image1 = imread('fish.bmp');
image2 = imread('motorcycle.bmp');

image1 = imresize(image1,[307 453]);
image2 = imresize(image2,[307 453]);

figure('Name', 'Original
 Images','NumberTitle','off');imshowpair(image1, image2, 'montage');
title("Original Images");

image1double = double(image1)/255;
image2double = double(image2)/255;

im1 = rgb2gray(image1double);
im2 = rgb2gray(image2double);

figure('Name', 'Grayscale Images','NumberTitle','off');imshowpair(im1,
 im2, 'montage');
title("Grayscale Images");

[im1h, im1w] = size(im1);
[im2h, im2w] = size(im2);

rows = max(im1h, im2h);
cols = max(im1w, im2w);
```

Original Images



Grayscale Images

Take the FFT of the two images

```
im1_FFT = fft2(im1, rows, cols);
im2_FFT = fft2(im2, rows, cols);
```

Find magnitude and phase of the two images

```
mag1 = abs(im1_FFT);
mag2 = abs(im2_FFT);

phase1 = angle(im1_FFT);
phase2 = angle(im2_FFT);
```

Recompute the frequency

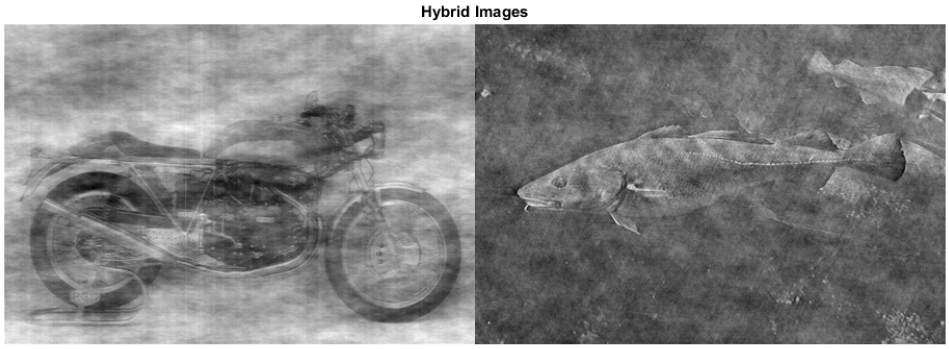
```
output1 = mag1 .* exp(1i*phase2);
output2 = mag2 .* exp(1i*phase1);
```

Find inverse images

```
inv1 = real(ifft2(output1));
inv2 = real(ifft2(output2));
```

Diplay New Hybrid Images

```
figure('Name', 'Hybrid Images','NumberTitle','off');imshowpair(inv1,
 inv2, 'montage');
title("Hybrid Images");
```



Published with MATLAB® R2017a

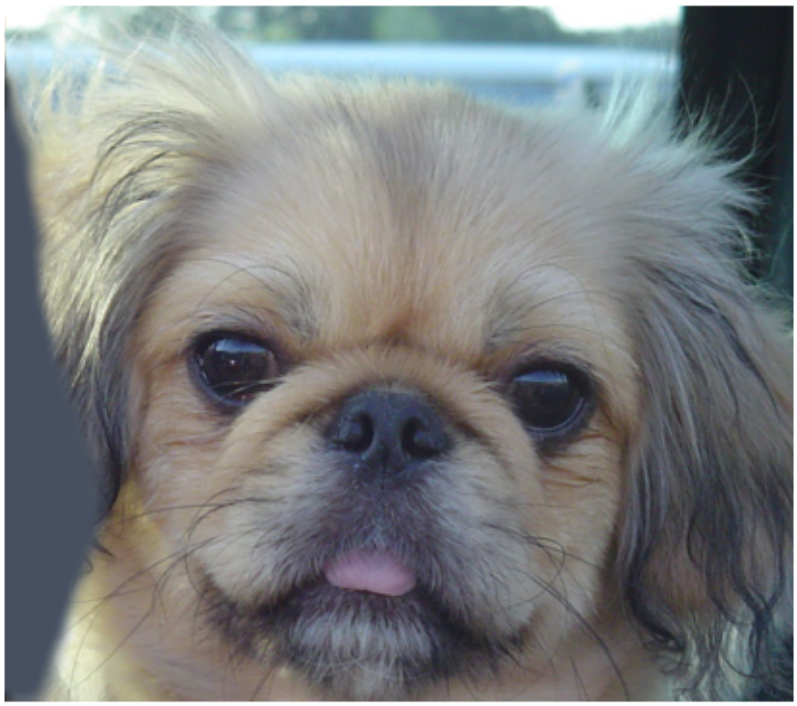
Table of Contents


```
% Author: Brandon Bench
% Hybrid Image

clc;
clear;
close all; % closes all figures
```

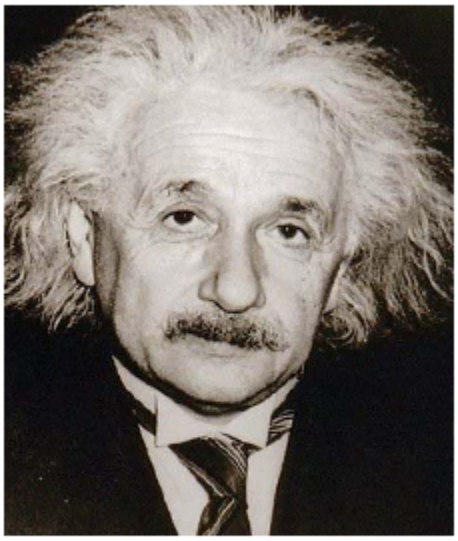
Setup

```
image1 = imread('dog.bmp');
image2 = imread('einstein.bmp');
image3 = imread('fish.bmp');

figure; imshow(image1);
title("Dog - Original Image");
figure; imshow(image2);
title("Einstein - Original Image");
figure; imshow(image3);
title("Fish - Original Image");

image1double = double(image1)/255;
image2double = double(image2)/255;
image3double = double(image3)/255;

im1 = rgb2gray(image1double);
im2 = rgb2gray(image2double);
im3 = rgb2gray(image3double);

figure; imshow(im1);
title("Dog - Grayscale Image");
figure; imshow(im2);
title("Einstein - Grayscale Image");
figure; imshow(im3);
title("Fish - Grayscale Image");
```

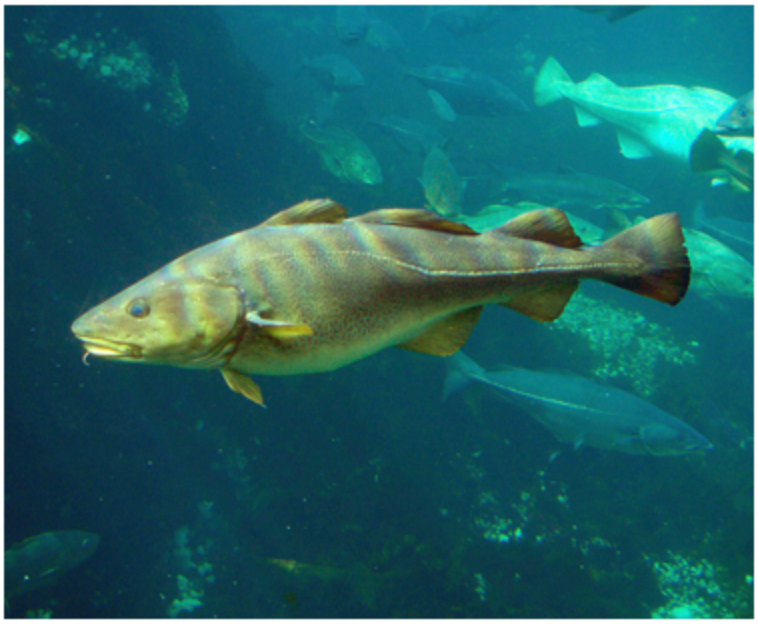
Dog - Original Image



Einstein - Original Image

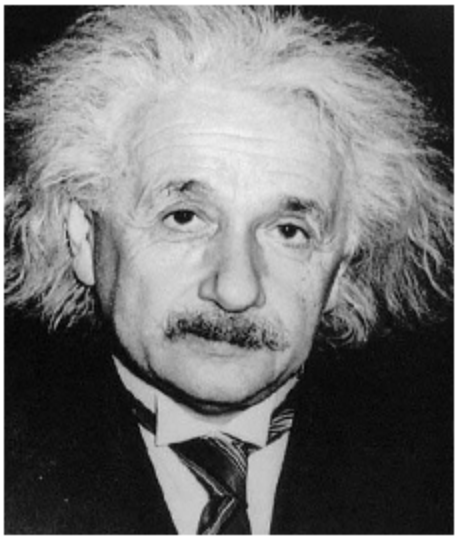


Fish - Original Image



Dog - Grayscale Image

Einstein - Grayscale Image



Fish - Grayscale Image

Applying the filters on input images

```
im1_fft  = fft2(im1);
im2_fft  = fft2(im2);
```

```
im3_fft  = fft2(im3);
```

Nuetralizing the Magnitude to display Phase only

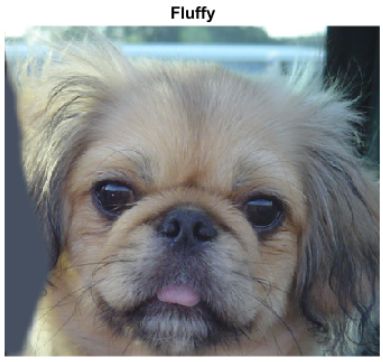
```
im1_P = exp(1i*angle(im1_fft));
im2_P = exp(1i*angle(im2_fft));
im3_P = exp(1i*angle(im3_fft));
```

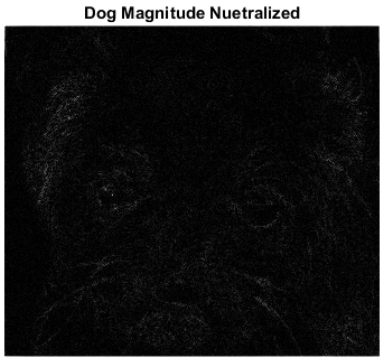
Inverse fft2

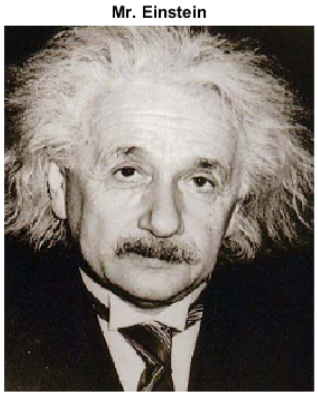
```
restoredP1 = ifft2(im1_P);
restoredP2 = ifft2(im2_P);
restoredP3 = ifft2(im3_P);
```

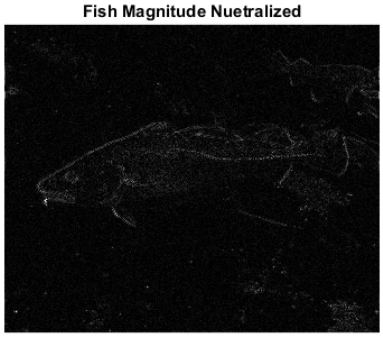
Calculating plotting limits

```
I_Phase_min = min(min(abs(restoredP1)));
I_Phase_max = max(max(abs(restoredP1)));


figure('position', [200, 200, 1000, 400]); subplot(1,2,1),
 imshow(image1), title("Fluffy")
subplot(1,2,2),
imshow(abs(restoredP1),[I_Phase_min I_Phase_max ]);
title("Dog Magnitude Nuetralized")

figure('position', [200, 200, 1000, 400]); subplot(1,2,1),
 imshow(image2), title("Mr. Einstein")
subplot(1,2,2),
imshow(abs(restoredP2),[I_Phase_min I_Phase_max ]);
title("Albert Magnitude Nuetralized")

figure('position', [200, 200, 1000, 400]); subplot(1,2,1),
 imshow(image3), title("Pescado")
subplot(1,2,2),
imshow(abs(restoredP3),[I_Phase_min I_Phase_max ]);
title("Fish Magnitude Nuetralized")
```









Published with MATLAB® R2017a

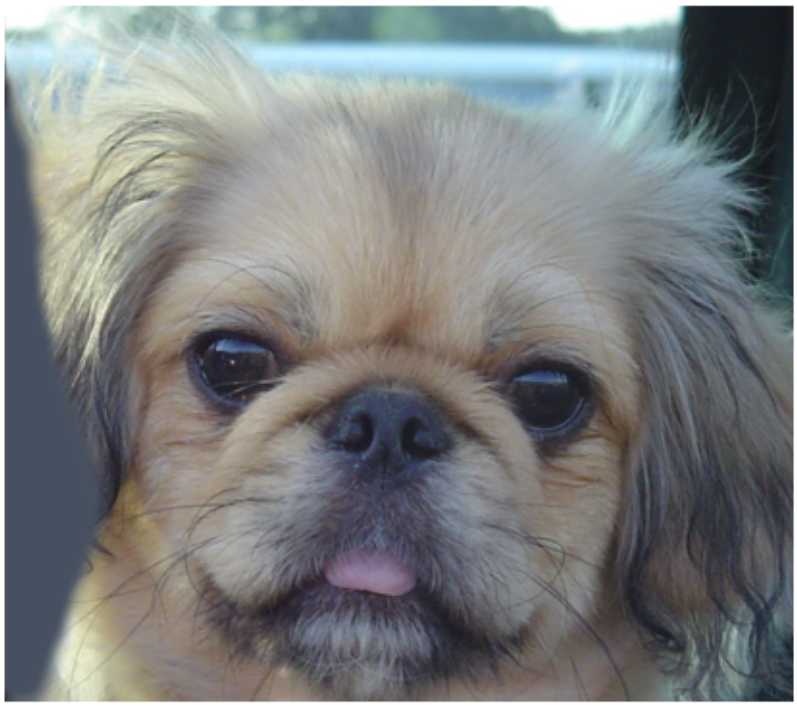
Table of Contents


```
% Author: Nick DeMarco
% Hybrid Image

clc;
clear;
close all; % closes all figures
```

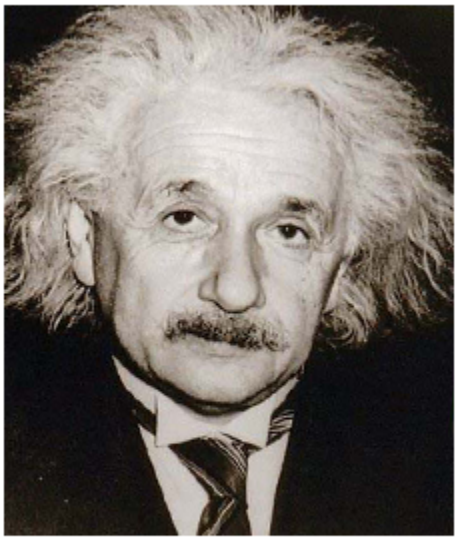
Setup

```
image1 = imread('dog.bmp');
image2 = imread('einstein.bmp');
image3 = imread('fish.bmp');

figure; imshow(image1);
title("Dog - Original Image");
figure; imshow(image2);
title("Einstein - Original Image");
figure; imshow(image3);
title("Fish - Original Image");

image1double = double(image1)/255;
image2double = double(image2)/255;
image3double = double(image3)/255;

im1 = rgb2gray(image1double);
im2 = rgb2gray(image2double);
im3 = rgb2gray(image3double);

figure; imshow(im1);
title("Dog - Grayscale Image");
figure; imshow(im2);
title("Einstein - Grayscale Image");
figure; imshow(im3);
title("Fish - Grayscale Image");
```

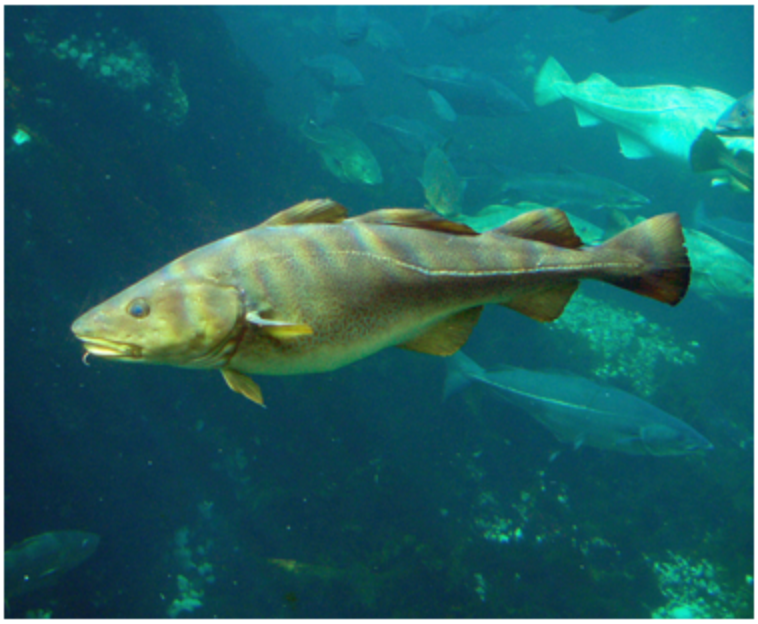
Dog - Original Image



Einstein - Original Image

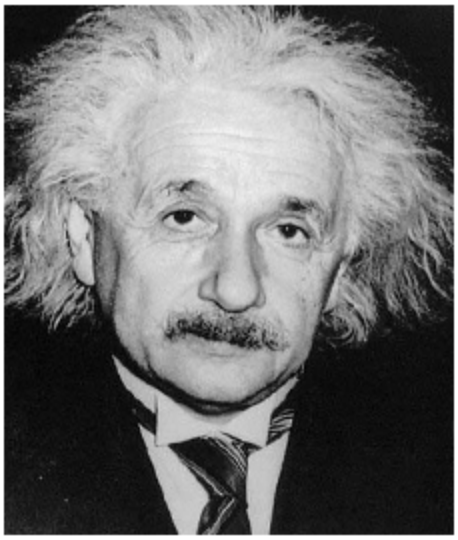


Fish - Original Image



Dog - Grayscale Image

Einstein - Grayscale Image



Fish - Grayscale Image

Applying the filters on input images

```
im1_fft  = fft2(im1);
im2_fft  = fft2(im2);
```

```
im3_fft  = fft2(im3);

gh = fftshift(im1_fft);
g2 = fftshift(im2_fft);
g3 = fftshift(im3_fft);
```

Nuetralizing the Phase to display Magnitude only

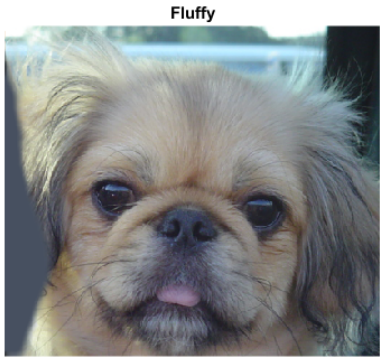
```
im1_M = abs(gh);
im2_M = abs(g2);
im3_M = abs(g3);
```

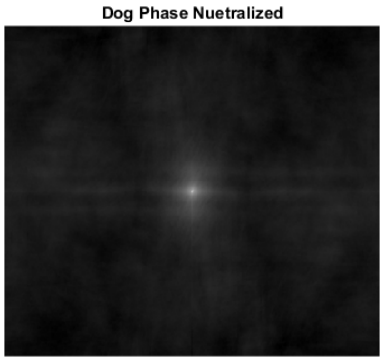
Inverse fft2

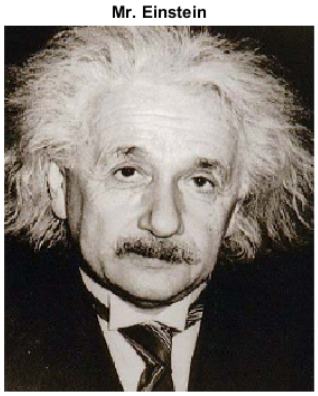
```
restoredP1 = log(abs(ifft2(im1_M*exp(1i*0)))+1);
restoredP2 = log(abs(ifft2(im2_M*exp(1i*0)))+1);
restoredP3 = log(abs(ifft2(im3_M*exp(1i*0)))+1);

re = fftshift(restoredP1);
r1 = fftshift(restoredP2);
r2 = fftshift(restoredP3);
```

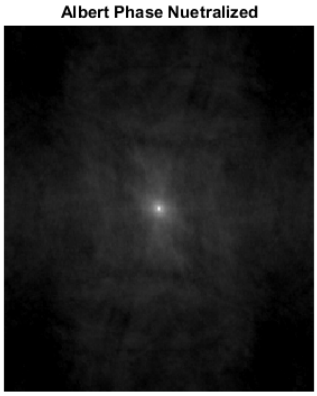
Calculating plotting limits

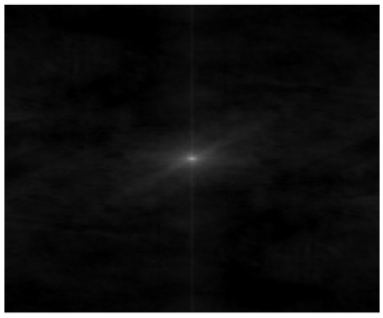
```
I_Mag_min = min(min(abs(restoredP1)));
I_Mag_max = max(max(abs(restoredP1)));


figure('position', [200, 200, 1000, 400]); subplot(1,2,1),
 imshow(image1), title("Fluffy")
subplot(1,2,2),
imshow(abs(re),[I_Mag_min I_Mag_max ]);
title("Dog Phase Nuetralized")

figure('position', [200, 200, 1000, 400]); subplot(1,2,1),
 imshow(image2), title("Mr. Einstein")
subplot(1,2,2),
imshow(abs(r1),[I_Mag_min I_Mag_max ]);
title("Albert Phase Nuetralized")

figure('position', [200, 200, 1000, 400]); subplot(1,2,1),
 imshow(image3), title("Pescado")
subplot(1,2,2),
imshow(abs(r2),[I_Mag_min I_Mag_max ]);
title("Fish Phase Nuetralized")
```











Published with MATLAB® R2017a

HW 1.C for 1.a.i (Hybrid Image)

Tic Toc Analysis

Critical Block	**Elapsed Time**
Reading in Images	0.159088 seconds
Image Resize	0.134052 seconds
Displaying Original Images	2.495655 seconds
Converting Images to Double	0.002885 seconds
Images to Grayscale	0.015694 seconds
Displaying Grayscale Images	0.263186 seconds
Finding size of images	0.000242 seconds
Initializing Rows and Cols	0.000188 seconds
Taking FFT	0.039679 seconds
Finding Magnitude	0.003576 seconds
Finding Phase	0.013051 seconds
Recomputing Frequency	0.008191 seconds
Taking Inverse	0.033244 seconds
Displaying Hybrid Images	0.254677 seconds

In the first part, FFT took around .003 seconds to run.

Profile Summary

Generated 02-Mar-2018 11:11:38 using performance time.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
HW1_Hybrid	1	0.134 s	0.016 s	
imread	2	0.051 s	0.006 s	
imread>call_format_specific_reader	2	0.039 s	0.001 s	
imagesci\private\readbmp	2	0.037 s	0.001 s	
imresize	2	0.035 s	0.007 s	
imagesci\private\imbmpinfo	2	0.029 s	0.002 s	
...\imbmpinfo>initializeBMPInfoStruct	2	0.019 s	0.001 s	
ima...i\private\initializeMetadataStruct	2	0.018 s	0.002 s	
datestr	2	0.016 s	0.002 s	
timefun\private\dateformverify	2	0.013 s	0.001 s	
timefun\private\formatdate	2	0.012 s	0.007 s	
imresize>parseInputs	2	0.011 s	0.003 s	
fft2	2	0.010 s	0.010 s	
ifft2	2	0.008 s	0.008 s	
imagesci\private\readbmpdata	2	0.008 s	0.001 s	
close	1	0.007 s	0.002 s	
ima...ivate\readbmpdata>bmpReadData24	2	0.007 s	0.004 s	
imresize>resizeAlongDim	4	0.007 s	0.003 s	
imagesci\private\imbmpinfo>readBMPInfo	2	0.006 s	0.001 s	
cnv2icudf	2	0.005 s	0.005 s	
contributions	4	0.005 s	0.004 s	
imagesci\private\imbmpinfo>readWin3xInfo	2	0.005 s	0.001 s	
close>safegetchildren	1	0.004 s	0.000 s	

angle	2	0.004 s	0.004 s	
allchild	1	0.004 s	0.002 s	
imresize>fixupSizeAndScale	2	0.003 s	0.002 s	
fileparts	2	0.003 s	0.002 s	
stringToChar	2	0.002 s	0.001 s	
imresize>parsePreMethodArgs	2	0.002 s	0.002 s	
rgb2gray	2	0.002 s	0.002 s	
...te\imbmpinfo>readWin3xBitmapHeader	2	0.002 s	0.002 s	
ima...rivate\readbmpdata>readFromFile	2	0.002 s	0.002 s	
imagesci\private\getFileFromURL	2	0.002 s	0.001 s	
resizeAlongDimUsingNearestNeighbor	2	0.002 s	0.002 s	
imresizemex (MEX-file)	2	0.002 s	0.002 s	
onCleanup>onCleanup.delete	5	0.002 s	0.001 s	
contains	2	0.001 s	0.001 s	
rot90	2	0.001 s	0.001 s	
stringToChar>@(x)convertToChar(x)	4	0.001 s	0.001 s	
cubic	4	0.001 s	0.001 s	
imread>get_full_filename	2	0.001 s	0.001 s	
cell.ismember	2	0.001 s	0.001 s	
isPureNearestNeighborComputation	8	0.001 s	0.001 s	
findFirstParamString	2	0.001 s	0.001 s	
deriveScaleFromSize	2	0.001 s	0.001 s	
imresize>findMethodArg	2	0.001 s	0.001 s	
rgb2gray>parse_inputs	2	0.001 s	0.001 s	
imresize>fixupSize	2	0.001 s	0.001 s	
imresize>scaleOrSize	2	0.001 s	0.001 s	
imagesci\private\imbmpinfo>getSignature	2	0.001 s	0.001 s	

ima...ate\imbmpinfo>readBMPFileHeader	2	0.001 s	0.001 s	
imresize>postprocessImage	2	0.001 s	0.001 s	
imagesci\private\imbmpinfo>postProcess	2	0.001 s	0.001 s	
imread>parse_inputs	2	0.001 s	0.001 s	
imresize>checkForMissingOutputArgument	2	0.001 s	0.000 s	
...te\imbmpinfo>readVersion3xColormap	2	0.001 s	0.001 s	
imresize>preprocessImage	2	0.000 s	0.000 s	
imresize>warnIfPostMethodArgs	2	0.000 s	0.000 s	
stringToChar>convertToChar	4	0.000 s	0.000 s	
close>getEmptyHandleList	1	0.000 s	0.000 s	
onCleanup>onCleanup.onCleanup	5	0.000 s	0.000 s	
strfun\private\isTextStrict	2	0.000 s	0.000 s	
...set(rootobj,'ShowHiddenHandles',Temp)	1	0.000 s	0.000 s	
allchild>getchildren	1	0.000 s	0.000 s	
close>request_close	1	0.000 s	0.000 s	
dimensionOrder	2	0.000 s	0.000 s	
imresize>parseParamValuePairs	2	0.000 s	0.000 s	
stringToLegacyText	2	0.000 s	0.000 s	
close>checkfigs	1	0.000 s	0.000 s	
datestr>getdateform	2	0.000 s	0.000 s	
ima...rivate\imbmpinfo>@()fclose(fid)	2	0.000 s	0.000 s	
uitools\private\allchildRootHelper	1	0.000 s	0.000 s	
ima...ate\imbmpinfo>decodeCompression	2	0.000 s	0.000 s	
imresize>isInputIndexed	6	0.000 s	0.000 s	
ima...vate\readbmpdata>@()fclose(fid)	2	0.000 s	0.000 s	
ispc	2	0.000 s	0.000 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self

time also includes overhead resulting from the process of profiling.

HW 1.C for 1.a.ii (Magnitude Neutralization)

1.c) We believe that the Fast Fourier Transform will be the most time consuming machine learning algorithm. It is the main algorithm we are using in this section of the homework.

1.c.i) This table is based on one function call per image. Not a combined time for all three images.
Also to note, these runtimes can vary widely each time the program is run.

Code Block	**Elapsed Time (averaged over 4 runs)**
Imread image	0.585700 E-02 seconds
Convert to Double type	0.326500 E-02 seconds
Convert to Grayscale	0.183425 E-02 seconds
2-D FFT	1.012050 E-02 seconds
Neutralizing the Magnitude	1.249530 E-02 seconds
2-D Inverse FFT	1.309500 E-02 seconds
Calculate Plot Limits to graph	1.752750 E-02 seconds

1.c.ii.2) fft2 and ifft2 each took 0.009 s and 0.006 s to run. Based off of one run.

Profile Summary

Generated 01-Mar-2018 17:11:12 using performance time.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
HW1_1_a_ii	1	0.104 s	0.016 s	
imread	3	0.054 s	0.007 s	
imread>call_format_specific_reader	3	0.042 s	0.001 s	
imagesci\private\readbmp	3	0.040 s	0.001 s	
imagesci\private\imbmpinfo	3	0.030 s	0.003 s	
...\imbmpinfo>initializeBMPInfoStruct	3	0.019 s	0.001 s	
ima...i\private\initializeMetadataStruct	3	0.018 s	0.002 s	
datestr	3	0.016 s	0.002 s	
timefun\private\dateformverify	3	0.014 s	0.001 s	
timefun\private\formatdate	3	0.012 s	0.007 s	
close	1	0.011 s	0.003 s	
ifft2	3	0.009 s	0.009 s	
imagesci\private\readbmpdata	3	0.009 s	0.001 s	
ima...ivate\readbmpdata>bmpReadData24	3	0.008 s	0.004 s	
close>safegetchildren	1	0.007 s	0.001 s	
imagesci\private\imbmpinfo>readBMPInfo	3	0.007 s	0.001 s	
fft2	3	0.006 s	0.006 s	
allchild	1	0.006 s	0.003 s	
imagesci\private\imbmpinfo>readWin3xInfo	3	0.006 s	0.002 s	
cnv2icudf	3	0.005 s	0.005 s	
angle	3	0.004 s	0.004 s	
rgb2gray	3	0.003 s	0.002 s	
...te\imbmpinfo>readWin3xBitmapHeader	3	0.003 s	0.002 s	
ima...rivate\readbmpdata>readFromFile	3	0.003 s	0.002 s	
fileparts	3	0.003 s	0.002 s	

onCleanup>onCleanup.delete	7	0.002 s	0.001 s	
rot90	3	0.002 s	0.002 s	
imread>get_full_filename	3	0.001 s	0.001 s	
imread>parse_inputs	3	0.001 s	0.001 s	
cell.ismember	3	0.001 s	0.001 s	
ima...ate\imbmpinfo>readBMPFileHeader	3	0.001 s	0.001 s	
imagesci\private\imbmpinfo>getSignature	3	0.001 s	0.001 s	
rgb2gray>parse_inputs	3	0.001 s	0.001 s	
imagesci\private\imbmpinfo>postProcess	3	0.001 s	0.001 s	
...set(rootobj,'ShowHiddenHandles',Temp)	1	0.001 s	0.001 s	
...te\imbmpinfo>readVersion3xColormap	3	0.001 s	0.001 s	
allchild>getchildren	1	0.001 s	0.001 s	
imagesci\private\getFileFromURL	3	0.001 s	0.001 s	
onCleanup>onCleanup.onCleanup	7	0.001 s	0.001 s	
close>request_close	1	0.000 s	0.000 s	
uitools\private\allchildRootHelper	1	0.000 s	0.000 s	
close>getEmptyHandleList	1	0.000 s	0.000 s	
close>checkfigs	1	0.000 s	0.000 s	
ima...vate\readbmpdata>@()fclose(fid)	3	0.000 s	0.000 s	
datestr>getdateform	3	0.000 s	0.000 s	
ispc	3	0.000 s	0.000 s	
ima...ate\imbmpinfo>decodeCompression	3	0.000 s	0.000 s	
ima...rivate\imbmpinfo>@()fclose(fid)	3	0.000 s	0.000 s	

HW 1.C for 1.a.iii (Phase Removal)

Tic Toc Analysis

Critical Block	**Elapsed Time**
Reading in Images	0.024449 seconds
Displaying Original Images	0.490118 seconds
Converting Images to Double	0.003109 seconds
Images to Grayscale	0.002685 seconds
Displaying Grayscale Images	0.433551 seconds
FFT2	0.010948 seconds
FFTShift	0.012342 seconds
Neutralizing Phase	0.003293 seconds
Inverse IFFT2	0.019009 seconds
FFTShift	0.001690 seconds
Calculating Plotting Limits	0.002852 seconds
Displaying Phase Neutralized Images	0.400440 seconds

In the third part, FFT took around .012 seconds to run while the inverse function IFFT2 took around .001 seconds to run.

Profile Summary

Generated 02-Mar-2018 11:21:00 using performance time.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
HW1_1_a_iii	1	0.357 s	0.016 s	
close	1	0.269 s	0.003 s	
close>request_close	1	0.239 s	0.010 s	
closereq	9	0.214 s	0.198 s	
imread	3	0.055 s	0.006 s	
imread>call_format_specific_reader	3	0.041 s	0.001 s	
imagesci\private\readbmp	3	0.040 s	0.001 s	
imagesci\private\imbmpinfo	3	0.030 s	0.003 s	
close>safegetchildren	1	0.026 s	0.004 s	
ismember	11	0.020 s	0.002 s	
...\imbmpinfo>initializeBMPInfoStruct	3	0.020 s	0.001 s	
ima...i\private\initializeMetadataStruct	3	0.019 s	0.002 s	
ismember>ismemberR2012a	11	0.019 s	0.004 s	
setdiff	2	0.018 s	0.001 s	
setdiff>setdiffR2012a	2	0.018 s	0.002 s	

datestr	3	0.017 s	0.002 s	
ismember>ismemberClassTypes	11	0.015 s	0.005 s	
close>request_close_helper	18	0.015 s	0.008 s	
timefun\private\dateformverify	3	0.014 s	0.001 s	
timefun\private\formatdate	3	0.013 s	0.007 s	
....graphics.internal.axesDestroyed(o,e)	6	0.012 s	0.011 s	
imagesci\private\readbmpdata	3	0.009 s	0.001 s	
unique	6	0.009 s	0.004 s	
ima...ivate\readbmpdata>bmpReadData24	3	0.009 s	0.004 s	
ifft2	3	0.006 s	0.006 s	
imagesci\private\imbmpinfo>readBMPInfo	3	0.006 s	0.001 s	
fft2	3	0.006 s	0.006 s	
cnv2icudf	3	0.006 s	0.006 s	
unique>uniqueR2012a	6	0.005 s	0.005 s	
imagesci\private\imbmpinfo>readWin3xInfo	3	0.005 s	0.001 s	
allchild	1	0.004 s	0.002 s	
fileparts	3	0.003 s	0.002 s	

ima...rivate\readbmpdata>readFromFile	3	0.003 s	0.002 s	
fftshift	6	0.002 s	0.002 s	
...te\imbmpinfo>readWin3xBitmapHeader	3	0.002 s	0.002 s	
rgb2gray	3	0.002 s	0.002 s	
gcbf	18	0.002 s	0.002 s	
...ddenHandles',oldUDDShowHiddenHandles)	9	0.002 s	0.002 s	
...ger>SubplotListenersManager.delete	3	0.002 s	0.002 s	
imagesci\private\getFileFromURL	3	0.002 s	0.001 s	
rot90	3	0.002 s	0.002 s	
axesDestroyed	6	0.002 s	0.002 s	
imread>get_full_filename	3	0.001 s	0.001 s	
onCleanup>onCleanup.delete	7	0.001 s	0.000 s	
contains	3	0.001 s	0.001 s	
close>checkfigs	10	0.001 s	0.001 s	
cell.ismember	3	0.001 s	0.001 s	
imagesci\private\imbmpinfo>postProcess	3	0.001 s	0.001 s	
ismember>ismemberBuiltinTypes	2	0.001 s	0.001 s	

close>getEmptyHandleList	3	0.001 s	0.001 s	
imagesci\private\imbmpinfo>getSignature	3	0.001 s	0.001 s	
imread>parse_inputs	3	0.001 s	0.001 s	
rgb2gray>parse_inputs	3	0.001 s	0.001 s	
ima...ate\imbmpinfo>readBMPFileHeader	3	0.001 s	0.001 s	
...te\imbmpinfo>readVersion3xColormap	3	0.000 s	0.000 s	
allchild>getchildren	1	0.000 s	0.000 s	
...set(rootobj,'ShowHiddenHandles',Temp)	1	0.000 s	0.000 s	
onCleanup>onCleanup.onCleanup	7	0.000 s	0.000 s	
strfun\private\isTextStrict	3	0.000 s	0.000 s	
stringToLegacyText	3	0.000 s	0.000 s	
datestr>getdateform	3	0.000 s	0.000 s	
uitools\private\allchildRootHelper	1	0.000 s	0.000 s	
ima...vate\readbmpdata>@()fclose(fid)	3	0.000 s	0.000 s	
ima...rivate\imbmpinfo>@()fclose(fid)	3	0.000 s	0.000 s	
ima...ate\imbmpinfo>decodeCompression	3	0.000 s	0.000 s	
ispc	3	0.000 s	0.000 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self

time also includes overhead resulting from the process of profiling.

When thinking about what could make our critical portions of our code run faster, we were thinking that it might be best to already have the images loaded ahead of time. We noticed that a bulk of our runtime involves loading the images. If we could eliminate this step, then we could drastically cut down on our critical portion runtime, thus improving the program's performance.