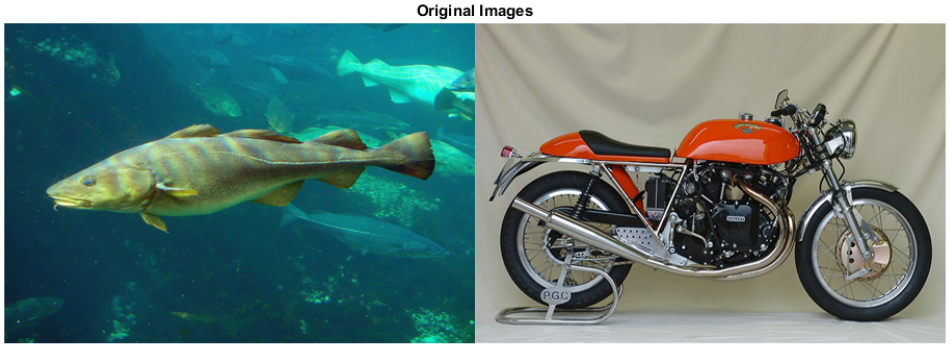
Table of Contents


```
% 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

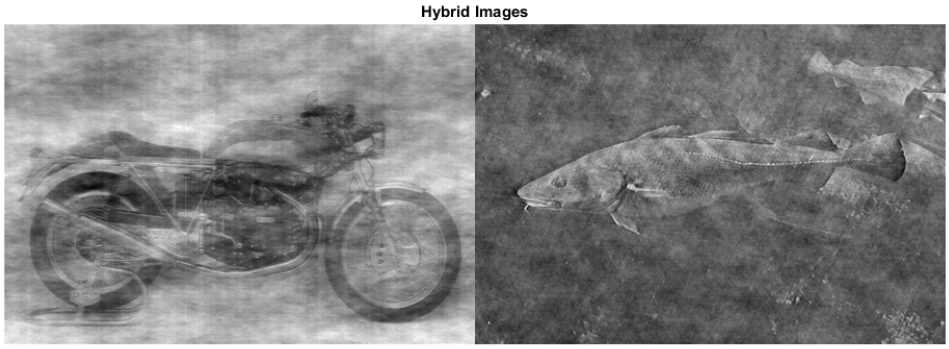
```
output1 = mag1 .* exp(1i*phase2);
output2 = mag2 .* exp(1i*phase1);
```

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

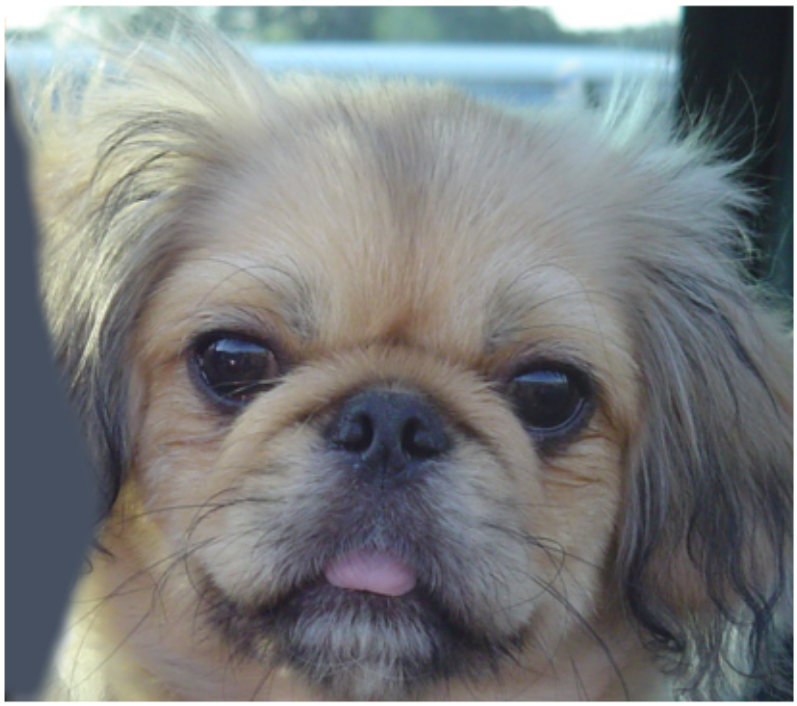
Table of Contents


```
% Author: Brandon Bench
% Hybrid Image

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

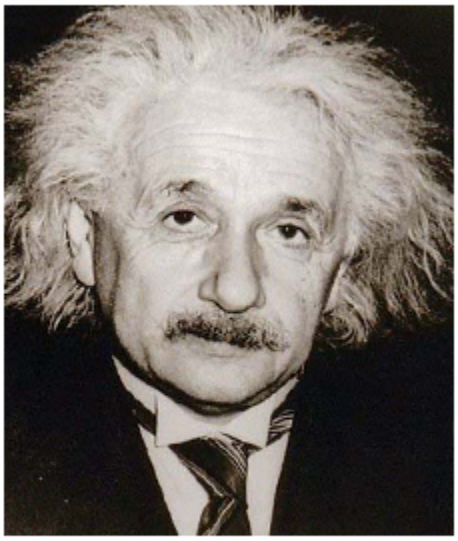
Setup

```
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");
```

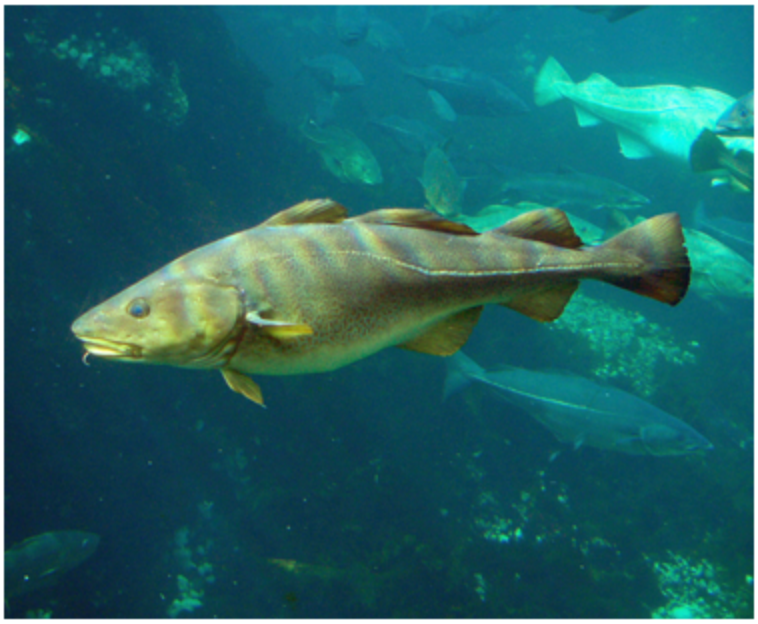
Dog - Original Image



Einstein - Original Image

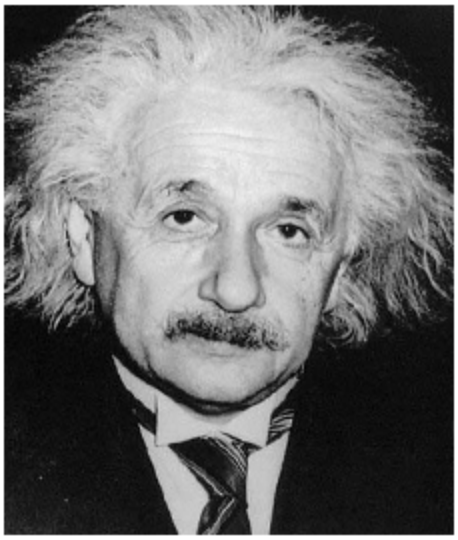


Fish - Original Image



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

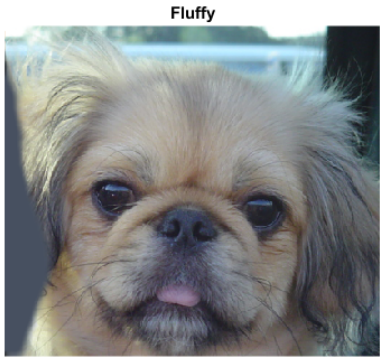
```
im1_P = exp(1i*angle(im1_fft));
im2_P = exp(1i*angle(im2_fft));
im3_P = exp(1i*angle(im3_fft));
```

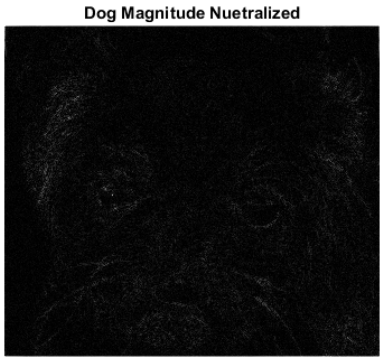
Inverse fft2

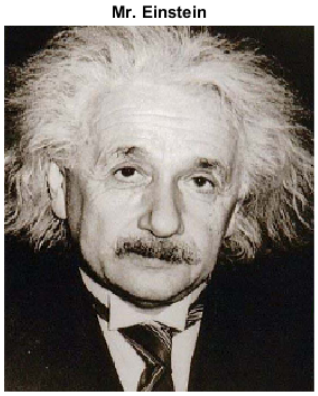
```
restoredP1 = ifft2(im1_P);
restoredP2 = ifft2(im2_P);
restoredP3 = ifft2(im3_P);
```

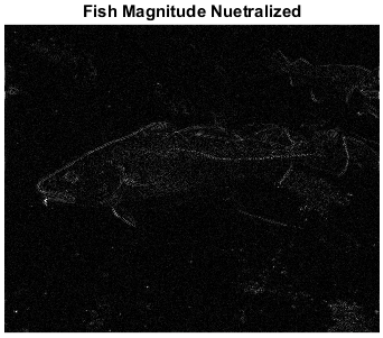
Calculating plotting limits

```
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

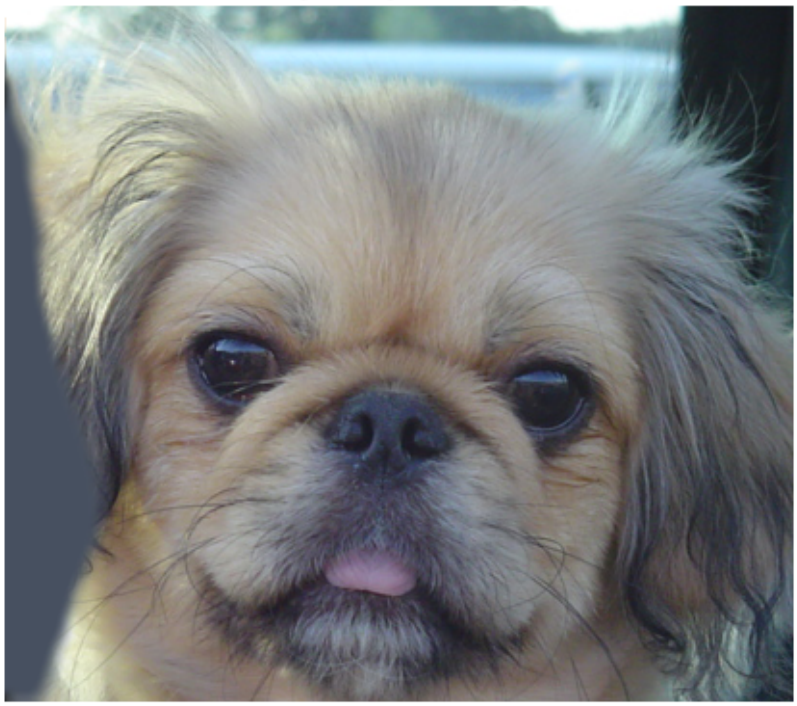
Table of Contents


```
% Author: Nick DeMarco
% Hybrid Image

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

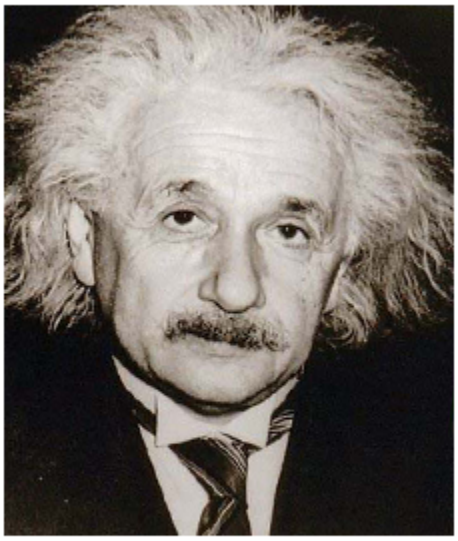
Setup

```
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");
```

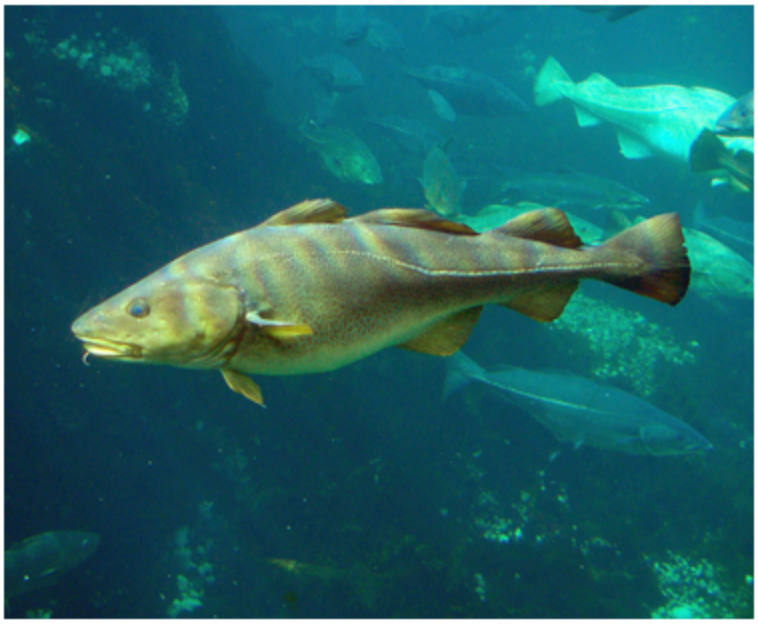
Dog - Original Image



Einstein - Original Image

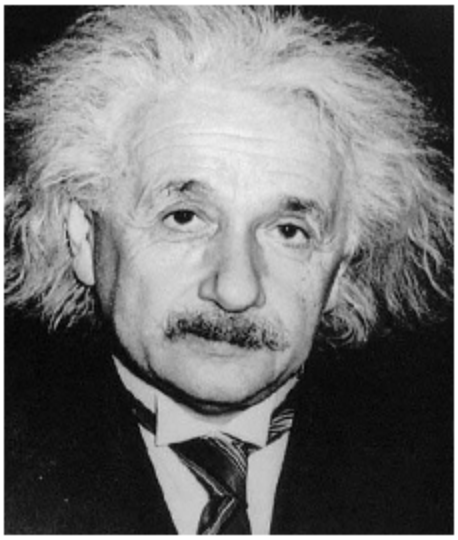


Fish - Original Image



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

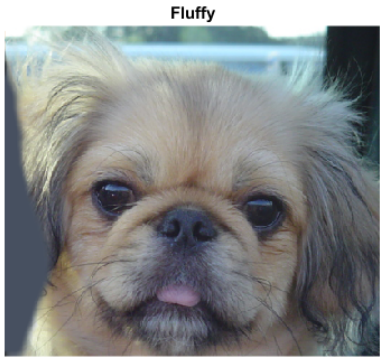
```
im1_M = abs(gh);
im2_M = abs(g2);
im3_M = abs(g3);
```

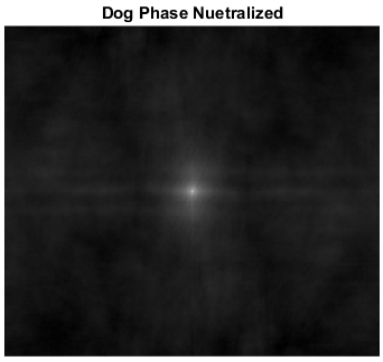
Inverse fft2

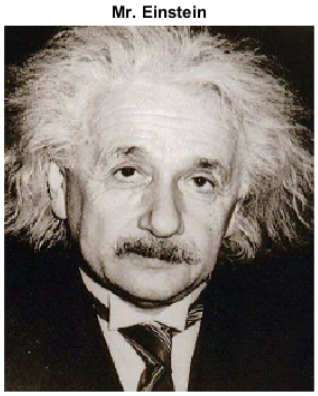
```
restoredP1 = log(abs(ifft2(im1_M*exp(1i*0)))+1);
restoredP2 = log(abs(ifft2(im2_M*exp(1i*0)))+1);
restoredP3 = log(abs(ifft2(im3_M*exp(1i*0)))+1);

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

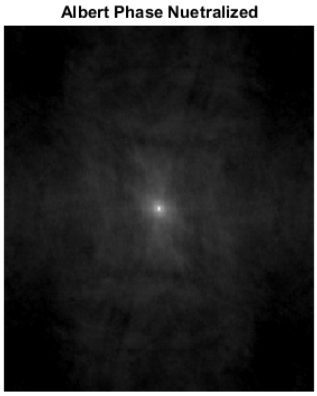
Calculating plotting limits

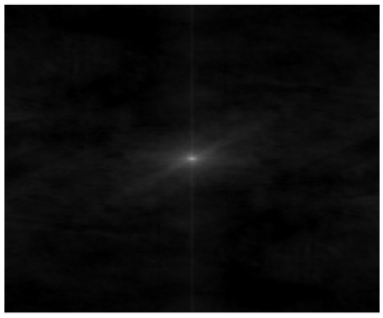
```
I_Mag_min = min(min(abs(restoredP1)));
I_Mag_max = max(max(abs(restoredP1)));


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