

## Contents

---

- Project 1: Read and display an image(raw file) and Affine transformation
- Q1.Determine a process where you can acquire an image and use Matlab commands to get that image into Matlab. Make sure you can read in gray-scale images.
- Reading a raw image file
- Using imread for cr2 format:-
- Using fread function:
- Convert to grayscale":
- Using imread for nef format:
- Using fread function:
- Convert to grayscale:
- Using imread for DNG format:
- Using fread function:
- Convert to grayscale:
- Using imread for .3FR format:
- Using fread function:
- Convert to grayscale;
- Q2.Read in an image and display it.
- To grayscale using colormap
- To grayscale
- Q3.Read in an image and scale the intensity values from 0 – 255.
- Q4.Affine transformation. Read in an image, and rotate and scale it by 35 degrees and a scale of 0.7 in all directions.
- Q/A
- 1. A brief write-up of what you did.
- 2.An image displayed with 32 and 255 intensity levels.
- 3. An image displayed with two different colormaps.
- 4.MSE vs. N for noisy images
- 5.Result of affine transformation.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## Project 1: Read and display an image(raw file) and Affine transformation

---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

**Q1.Determine a process where you can acquire an image and use Matlab commands to get that image into Matlab. Make sure you can read in gray-scale images.**

---

```
% In the first part, we try to read an image in matlab and check if the
% image can be viewed successfully. Three different image file formats were
% used:
%1).3FR
%2).cr2
%3).nef
%4).dng
% A sample image of the format raw file can be viewed by using the following code:
```

## Reading a raw image file

---

```
row=576; col=768;
fin=fopen('Human.raw');
I=fread(fin, col*row*3,'uint8->uint8'); %// Read in as a single byte stream
I = reshape(I, [col row 3]); %// Reshape so that it's a 3D matrix - Note that this is column major
Ifinal = flipdim(imrotate(I, -90),2); % // The clever transpose
imshow(Ifinal);
fclose(fin);
```



### Using imread for cr2 format:-

Now we try to view the image using imread. On running the code, the image can be viewed successfully.

```
z = imread('Bike.cr2');  
info = imfinfo('Bike.cr2')  
figure  
image(z)
```

```
info =
```

```
4x1 struct array with fields:
```

```
Filename  
FileModDate  
FileSize  
Format  
FormatVersion  
Width  
Height  
BitDepth  
ColorType  
FormatSignature  
ByteOrder  
NewSubFileType  
BitsPerSample  
Compression  
PhotometricInterpretation  
StripOffsets  
SamplesPerPixel  
RowsPerStrip  
StripByteCounts  
XResolution  
YResolution  
ResolutionUnit  
Colormap  
PlanarConfiguration  
TileWidth  
TileLength  
TileOffsets  
TileByteCounts  
Orientation  
FillOrder  
GrayResponseUnit  
MaxSampleValue  
MinSampleValue  
Thresholding  
Offset  
Make  
Model  
DateTime  
Artist  
JPEGInterchangeFormat
```

JPEGInterchangeFormatLength  
 XMP  
 Copyright  
 DigitalCamera  
 GPSInfo  
 UnknownTags



#### Using fread function:

```
info = imfinfo('converted_images_1.dng')
fid=fopen('converted_images_1.dng');
A=fread(fid,[8896,5920],'uint16');
fclose(fid);
A=A';
imagesc(A);
```

% On running the code we see that the image had a lot of noise and hence  
 % the image was not clear.

info =

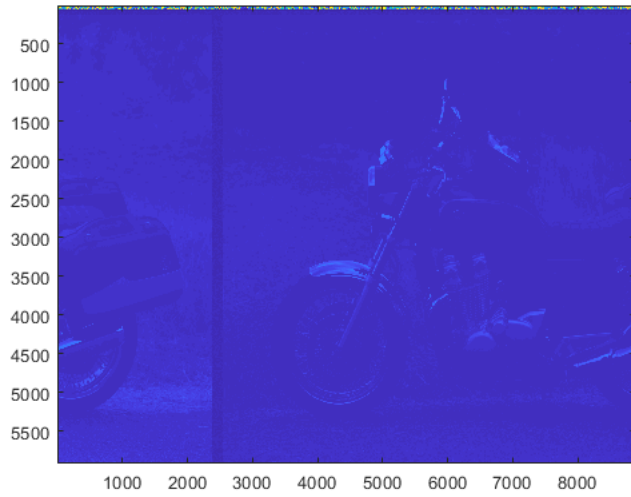
struct with fields:

Filename: 'C:\Users\Arun\Documents\FIT Sem 4\Digital Signal\converted\_images\_1.dng'

```

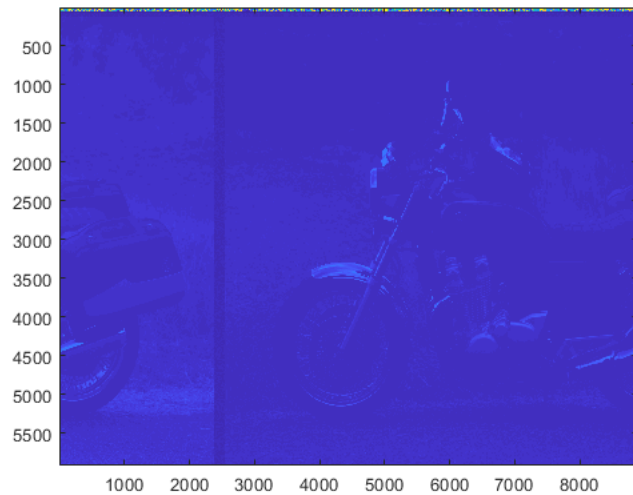
FileModDate: '27-Jan-2021 07:38:12'
FileSize: 106418722
Format: 'tif'
FormatVersion: []
Width: 256
Height: 171
BitDepth: 24
ColorType: 'truecolor'
FormatSignature: [73 73 42 0]
ByteOrder: 'little-endian'
NewSubFileType: 1
BitsPerSample: [8 8 8]
Compression: 'Uncompressed'
PhotometricInterpretation: 'RGB'
StripOffsets: 162020
SamplesPerPixel: 3
RowsPerStrip: 171
StripByteCounts: 131328
XResolution: []
YResolution: []
ResolutionUnit: 'Inch'
ColorMap: []
PlanarConfiguration: 'Chunky'
TileWidth: []
TileLength: []
TileOffsets: []
TileByteCounts: []
Orientation: 1
FillOrder: 1
GrayResponseUnit: 0.0100
MaxSampleValue: [255 255 255]
MinSampleValue: [0 0 0]
Thresholding: 1
Offset: 8
Make: 'Canon'
Model: 'Canon EOS 5DS'
Software: 'Adobe DNG Converter 13.1 (Windows)'
DateTime: '2021:01:27 02:38:12'
Artist: 'JAKOB ROHRBACH'
SubIFDs: {1x5 cell}
XMP: '<?xpacket begin="ï»¿" id="W5M0MpCehiHzreSzNTczkc9d">?<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="Adobe XMP Core 5.6-c140 79.160451
Copyright: 'www.jaro.ch'
DigitalCamera: [1x1 struct]
DNGVersion: [1 4 0 0]
DNGBackwardVersion: [1 1 0 0]
UniqueCameraModel: 'Canon EOS 5DS'
ColorMatrix1: [1x9 double]
ColorMatrix2: [1x9 double]
CameraCalibration1: [0.9539 0 0 0 1 0 0 0 0.9770]
CameraCalibration2: [0.9539 0 0 0 1 0 0 0 0.9770]
AnalogBalance: [1 1 1]
AsShotNeutral: [0.4638 1 0.5206]
BaselineExposure: 0.2500
BaselineNoise: 0.8000
BaselineSharpness: 1.2000
LinearResponseUnit: 1
CameraSerialNumber: '018021000087'
LensInfo: [24 105 NaN NaN]
ShadowScale: 1
DNGPrivateData: [1x41294 double]
CalibrationIlluminant1: 17
CalibrationIlluminant2: 21
AliasLayerMetadata: [1x16 double]
OriginalRawFileName: 'Bike.cr2'
UnknownTags: [20x1 struct]

```



Convert to grayscale":

```
I = rgb2gray(z);  
figure  
imshow(I)
```



#### Using imread for nef format:

Now, we try to open an image of the file format .NEF. On running the code, the image can be viewed.

```
warning('off')
o = imread('baby.nef');
info_1 = imfinfo('baby.nef')
figure
image(o)
```

```
info_1 =
```

```
struct with fields:
```

```
Filename: 'C:\Users\Arun\Documents\FIT Sem 4\Digital Signal\baby.nef'
FileModDate: '01-Jun-2008 00:10:12'
FileSize: 18672009
Format: 'tif'
```

```

FormatVersion: []
Width: 320
Height: 212
BitDepth: 24
ColorType: 'truecolor'
FormatSignature: [73 73 42 0]
ByteOrder: 'little-endian'
NewSubFileType: 1
BitsPerSample: [8 8 8]
Compression: 'Uncompressed'
PhotometricInterpretation: 'RGB'
StripOffsets: 30818
SamplesPerPixel: 3
RowsPerStrip: 212
StripByteCounts: 203520
XResolution: 300
YResolution: 300
ResolutionUnit: 'Inch'
Colormap: []
PlanarConfiguration: 'Chunky'
TileWidth: []
TileLength: []
TileOffsets: []
TileByteCounts: []
Orientation: 1
FillOrder: 1
GrayResponseUnit: 0.0100
MaxSampleValue: [255 255 255]
MinSampleValue: [0 0 0]
Thresholding: 1
Offset: 8
Make: 'NIKON CORPORATION'
Model: 'NIKON D3'
Software: 'Capture NX 2.0.0 M'
DateTime: '2008:06:01 02:07:1543503872'
Artist: 'Jason P. Odell'
SubIFDs: {[2x1 struct] [1x1 struct]}
ReferenceBlackWhite: [0 255 0 255 0 255]
XMP: '<?xpacket begin="ï¿" id="W5M0MpCehiHzreSzNTczkc9d"><x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="Public XMP Toolkit Core 3.5">
ITPC: [1x33 double]
DigitalCamera: [1x1 struct]
ICCProfileOffset: 6512
DateTimeOriginal: '2008:04:27 07:59:07'
UnknownTags: [1x1 struct]

```

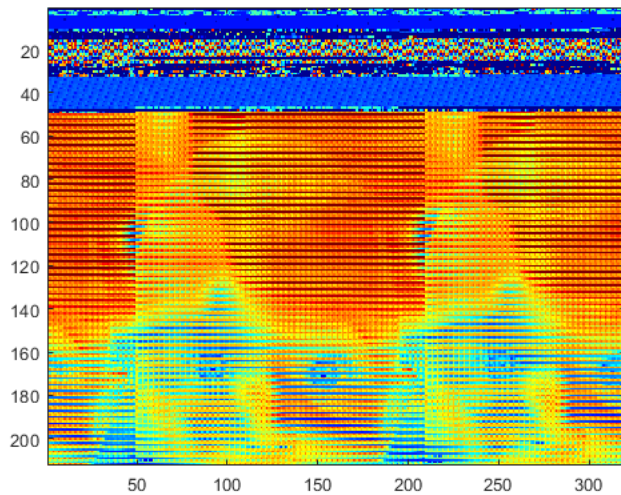


#### Using fread function:

```

fid=fopen('baby.nef');
B=fread(fid,[320,212],'uint16');
fclose(fid);
B=B';
colormap(jet);
imagesc(B);
% On running the code we see that the image had some noise even though it was almost visible.

```



### Convert to grayscale:

```
J = rgb2gray(o);
figure
imshow(J)
```



### Using imread for DNG format:

Now, we try to open an image of the file format .DNG. On running the code, the image can be viewed.

```
d = imread('Tree.DNG');
figure
info_2 = imfinfo('Tree.DNG')
image(d)
```

```
info_2 =
```

```
struct with fields:
```

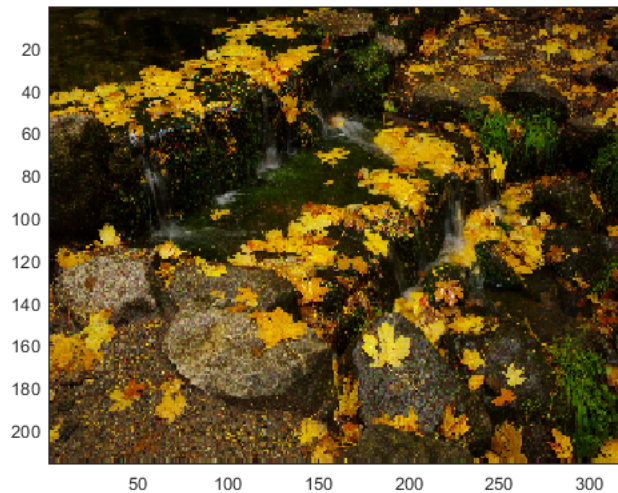
```
    Filename: 'C:\Users\Arun\Documents\FIT Sem 4\Digital Signal\Tree.DNG'
    FileModDate: '27-Jan-2021 01:18:46'
    FileSize: 18336108
    Format: 'tif'
    FormatVersion: []
    Width: 320
    Height: 216
    BitDepth: 24
    ColorType: 'truecolor'
    FormatSignature: [73 73 42 0]
    ByteOrder: 'little-endian'
    NewSubFileType: 1
    BitsPerSample: [8 8 8]
    Compression: 'Uncompressed'
    PhotometricInterpretation: 'RGB'
    StripOffsets: 7044
    SamplesPerPixel: 3
    RowsPerStrip: 216
    StripByteCounts: 207360
    XResolution: 72
    YResolution: 72
    ResolutionUnit: 'Inch'
    Colormap: []
    PlanarConfiguration: 'Chunky'
```



```

        TileWidth: []
        TileLength: []
        TileOffsets: []
        TileByteCounts: []
        Orientation: 1
        FillOrder: 1
        GrayResponseUnit: 0.0100
        MaxSampleValue: [255 255 255]
        MinSampleValue: [0 0 0]
        Thresholding: 1
            Offset: 18335586
            Make: 'Leica Camera AG'
            Model: 'M9 Digital Camera'
            Software: '1.002'
            SubIFDs: {[1x1 struct]}
                XMP: '<?xpacket begin="ï¿" id="W5M0MpCehiHzreSzNTczkc9d"?><x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="XMP toolkit 3.0-28, framework 1.6
        DigitalCamera: [1x1 struct]
        DateTimeOriginal: '2009:10:19 15:48:10'
        DNGVersion: [1 0 0 0]
        UniqueCameraModel: 'M9 Digital Camera'
        ColorMatrix1: [1x9 double]
        ColorMatrix2: [1x9 double]
        CameraCalibration1: [1 0 0 0 1 0 0 0 1]
        CameraCalibration2: [1 0 0 0 1 0 0 0 1]
        AsShotNeutral: [0.4190 1 0.8184]
        BaselineExposure: -0.5000
        BaselineNoise: 1
        BaselineSharpness: 1
        LinearResponseUnit: 1
        CameraSerialNumber: '*****'
        MakerNoteSafety: 1
        CalibrationIlluminant1: 17
        CalibrationIlluminant2: 21
        AliasLayerMetadata: [1x16 double]
        UnknownTags: [5x1 struct]

```



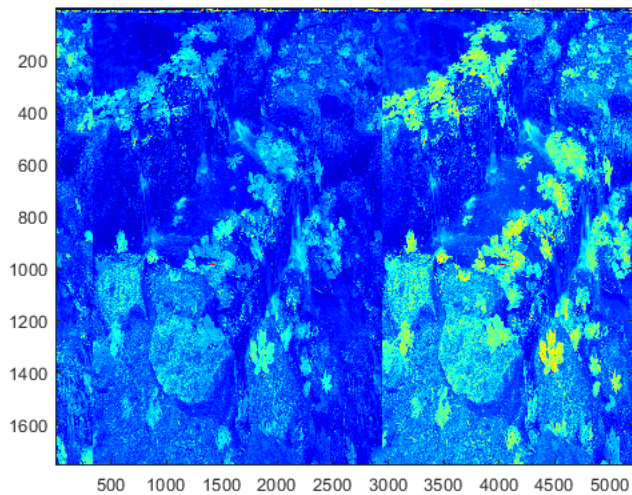
#### Using fread function:

```

fid=fopen('Tree.DNG');
C=fread(fid,[5216,3472],'uint16');
fclose(fid);
C=C';
colormap(jet);
imagesc(C);

```

% On running the code we see that the image is not very clear on using the  
% fread function



### Convert to grayscale:

```
K = rgb2gray(d);
figure
imshow(K)
```



### Using imread for .3FR format:

```
dd = imread('Raw2.3FR');
figure
info_3 = imfinfo('Raw2.3FR')
image(dd)
```

```
info_3 =
```

```
struct with fields:
```

```

    Filename: 'C:\Users\Arun\Documents\FIT Sem 4\Digital Signal\Raw2.3FR'
    FileModDate: '27-Jan-2021 05:47:17'
    FileSize: 108049920
    Format: 'tif'
    FormatVersion: []
    Width: 4136
    Height: 3100
    BitDepth: 24
    ColorType: 'YCbCr'
    FormatSignature: [73 73 42 0]
    ByteOrder: 'little-endian'
    NewSubFileType: 1
    BitsPerSample: [8 8 8]
    Compression: 'JPEG'
    PhotometricInterpretation: 'YCbCr'
    StripOffsets: 8192
    SamplesPerPixel: 3
    RowsPerStrip: 3100
    StripByteCounts: 1757206
    XResolution: 72
    YResolution: 72
    ResolutionUnit: 'Inch'
    Colormap: []
    PlanarConfiguration: 'Chunky'
    TileWidth: []
    TileLength: []

```

```

    TileOffsets: []
    TileByteCounts: []
    Orientation: 1
    FillOrder: 1
    GrayResponseUnit: 0.0100
    MaxSampleValue: [255 255 255]
    MinSampleValue: [0 0 0]
    Thresholding: 1
    Offset: 8
    Make: 'Hasselblad'
    Model: 'X1D II 50C'
    Software: '10.00.07.69'
    DateTime: '2019:06:11 08:16:46'
    SubIFDs: {[1x1 struct] [1x1 struct]}
    XMP: '<x:xmpmeta xmlns:x='adobe:ns:meta/'><rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'><rdf:Description rdf:about="" x
    Copyright: ''
    DigitalCamera: [1x1 struct]
    UniqueCameraModel: 'Hasselblad X1D II 50C'
    ColorMatrix1: [1x9 double]
    AsShotNeutral: [0.5702 1 0.4332]
    AliasLayerMetadata: [1x16 double]
    UnknownTags: [2x1 struct]

```

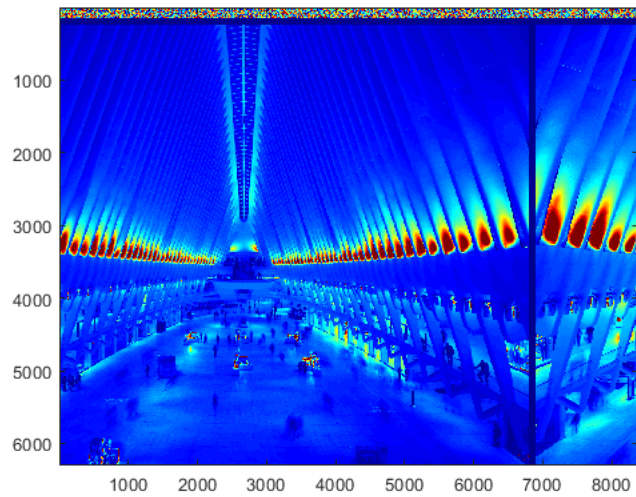


#### Using fread function:

```

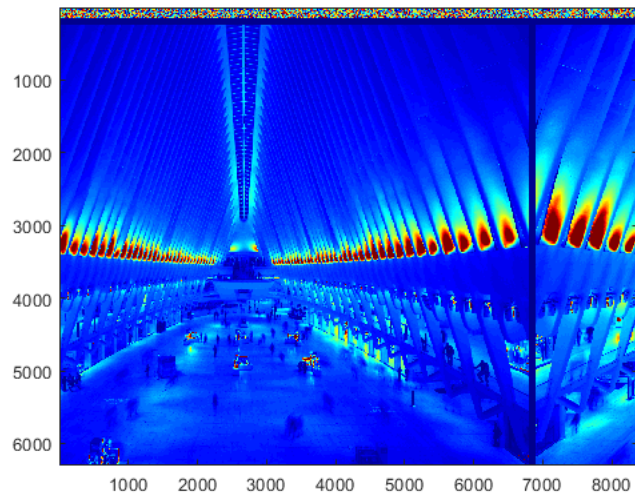
fid=fopen('Raw2.3FR');
CC=fread(fid,[8384,6304],'uint16');
fclose(fid);
MM=(CC)';
imagesc(MM);
colormap(jet);
% On running the code we see that the image is not very clear on using the
% fread function

```



Convert to grayscale;

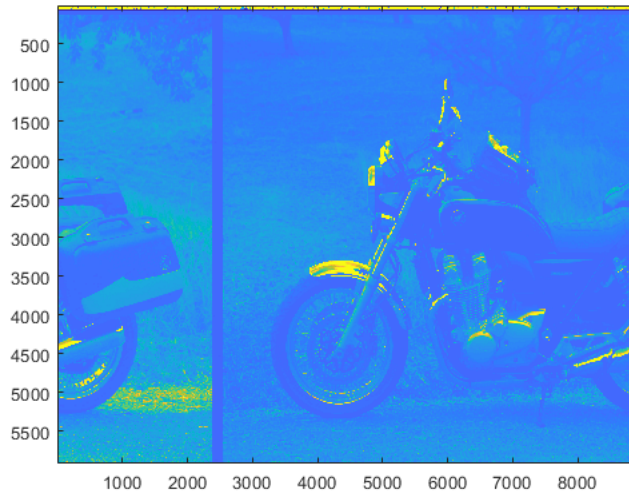
```
KK = rgb2gray(dd);  
figure  
imshow(KK)
```



**Q2.Read in an image and display it.**

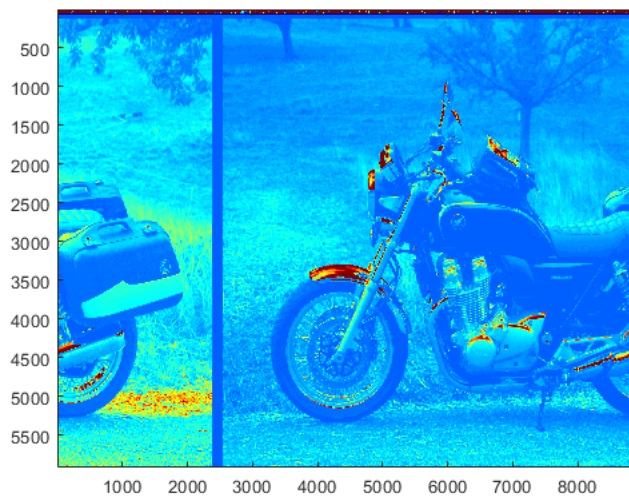
```
clims=[0,10000];  
figure  
imagesc(A, clims)
```





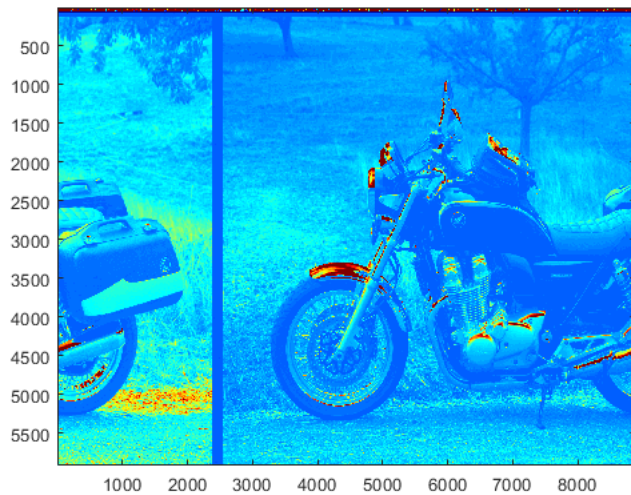
### To grayscale using colormap

```
colormap(jet)
%colormap(bone)
% Colormap function did not work as this is a 3D image.
```



### To grayscale

```
II = rgb2gray(Ifinal);
figure
imshow(II)
```



### Q3. Read in an image and scale the intensity values from 0 – 255.

```
%Read an image:
fid=fopen('Tree.DNG');
C=fread(fid,[5216,3472],'uint16');
fclose(fid);
C=C';
imagesc(C);

%Scale the intensity values to 0-255:
C = C./3; % some intensity values could be > 255
maxPix = max(max(C)); % find the maximum
minPix = min(min(C)); % find the minimum

% make it to the range between 0 to 1 and then multiply by 255.
C = ((C + minPix)/maxPix)*255;
image(C);

% Add noise to the image
% Take the average of N noisy versions of the original (using independent noise samples!)

N = [10,20,30,40,50];
SumOfImages = zeros(size(C));
mean = 20;
variance = 20;
l = length(N);
MSE=zeros(1,1);
```

```

for i=1:l
    for x=1:N(i)
        SumOfImages= SumOfImages + imnoise(C,'gaussian',mean,variance);
    end
    Avg = SumOfImages./i;
    MSE(i) = immse(C,Avg);
end
fprintf('\n The mean-squared error is %0.4f\n', MSE);
plot(N,MSE)
title('N vs MSE');
xlabel('N');
ylabel('MSE');

```

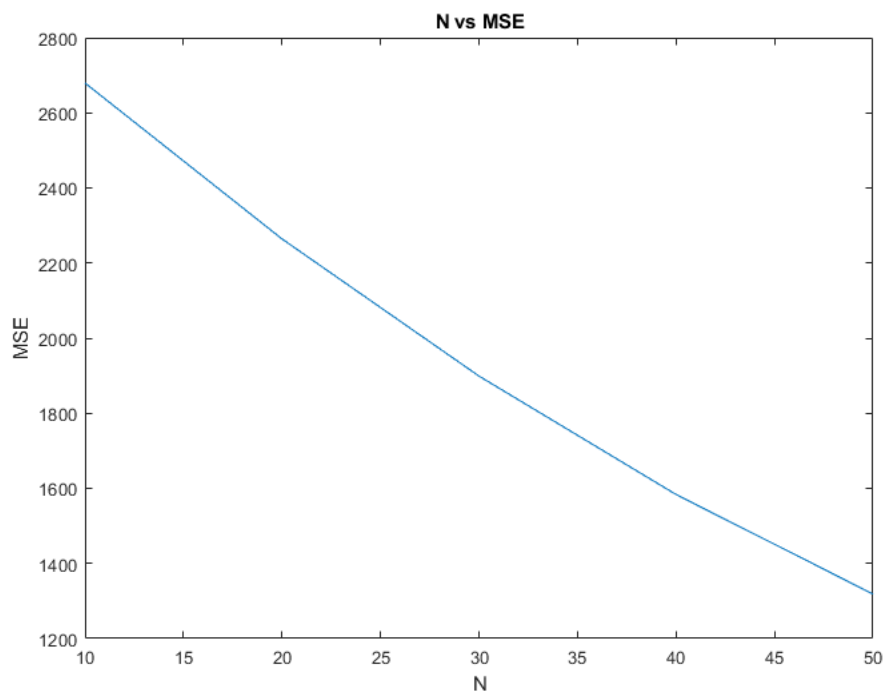
The mean-squared error is 2679.4197

The mean-squared error is 2264.1771

The mean-squared error is 1898.9343

The mean-squared error is 1583.6913

The mean-squared error is 1318.4479



**Q4.Affine transformation. Read in an image, and rotate and scale it by 35 degrees and a scale of 0.7 in all directions.**

```

%Read an Image
C=imread('converted_images_1.dng');
% Rotate and Scale the image
theta = 35;
tform= affine2d([cosd(theta) -sind(theta) 0; sind(theta) cosd(theta) 0; 0 0 1]);
J = imwarp(C,tform);
colormap(gray)
imagesc(J);

```





Scaling Image

```
scale = affine2d([0.7 0 0;0 0.7 0;0 0 1]);
L = imwarp(C,scale);
imagesc(L);
```



## Q/A

## 1. A brief write-up of what you did.

```
%A)
% In the first part, we try to read an image in matlab and check if the
% image can be viewed successfully. Three different image file formats were
% used:
%1).3FR
%2).cr2
%3).nef
%4).dng
```

```
% In the second part of the question, we try to use the colormap function
% to convert the image to grayscale. Although, when colourmap(gray)
% function was used, at some instances, the images were not being plotted.
% Hence I used the rgb2gray function and plotted the grayscale version of
% the image.

% In the third question, a value of 20 was provided for both the mean and
% variance and the image was scaled between 0-255 by using the immnoise
% function. Then MSE was computed using the immse function.

% In the fourth question, an image was read and rotated to an angle of 30
% degrees. Another image was scaled to 0.7 in all directions.
```

---

**2. An image displayed with 32 and 255 intensity levels.**

---

```
%A. A code has been used to check intensity levels and is shown above.
```

---

**3. An image displayed with two different colormaps.**

---

colormaps(gray) and colormaps(jet) was used on some of my images.

---

**4. MSE vs. N for noisy images**

---

A graph has been plotted between MSE and N values and the code has been implemented.

---

**5. Result of affine transformation.**

---

Bike.cr2 image file was used to test the rotation and scaling values. The image rotated successfully at an angle of 30 degrees and a scaling value of 0.7 was also used.

---

*Published with MATLAB® R2019a*