**ELEC 221: Signals and Systems**

**Climate Change and The Antarctic Sea Ice**

**Extent - Visual to Statistics**

**Submitted By: Atif Mahmud**

**TASK 1**

**Q0.** Total Number of Frames = 3081

Frame Rate = 29.97 frames per second

%% MATLAB Code to Complete Q0

% Creating a VideoReader Object

vidObj = VideoReader('NASA.mp4');

get(vidObj);

% % Source: [http://stackoverflow.com/questions/31932380/why-is-matlab-unable-%](http://stackoverflow.com/questions/31932380/why-is-matlab-unable-%25) to-determine-the-number-of-frames-in-a-video-file

% We need to read the last frame

data = read(vidObj, Inf);

numFrames = vidObj.NumberOfFrames;

MATLAB Code for Q0

**Q1.** Size of Data Array: <720 x 1280 x 3unit8>

Height of each frame: 720 pixels

Width of each frame: 1280 pixels

Bits per pixel: 24

%% Extracting all the frames onto my drive

% I will then use the ones I need

% Source: https://www.youtube.com/watch?v=AI-1ch6CHkI&t=295s

for img = 1:vidObj.NumberOfFrames;

filename=strcat('frame\_number', num2str(img), '.png');

frame=read(vidObj, img);

imwrite(frame, filename);

end

MATLAB Code for Writing the Frames Onto my Drive

**TASK 2**

**Q2.** 8 bits per pixel

**Q3.**  To convert to grayscale, I used MATLAB’s rgb2gray()function. The function works by removing hue (actual colour), and saturation (amount of grey) in the colour, but retaining the luminance. The luminance shows up as intensity in the grayscale image.

%% Converting from RGB to Grayscale and writing on drive

% Source: https://www.mathworks.com/help/matlab/ref/rgb2gray.html

for num= 1363:1703 %the frames we are interested in

file=strcat('frame\_number', num2str(num), '.png');

RGB=imread(file);

gray\_file=strcat('Gray\_Scale ','frame\_number', num2str(num),'.png');

gray\_image=rgb2gray(RGB);

imwrite(gray\_image, gray\_file);

end

MATLAB Code for Converting the Images to Grayscale



Figure 1: Displaying a frame using imshow()

My student number is 48660147, hence my date was April 18.

% Using imshow to display April 18 Photograph

imshow('Gray\_Scaleframe\_number1416.png');

MATLAB Code for Fig 1.

**Q4.** Once we convert to grayscale, each pixel is represented as an *intensity,* in 8-bits, and not a combination of red, green, and blue. Thus, it is easier to convert to black-and-white using the algortihm explained in **Q5** once each pixel is represented as an intensity.

**TASK 3**

**Q5**. To convert from Grayscale to black-and-white, I used MATLAB’s im2bw()function. What that function does is, upon analysis of an image using a certain value for intensity, converts everything below that level to black, and everything above that to white. The intensity is specified in the level input of the function. If no level is specified by the user, MATLAB uses a default of 0.5.

%% Changing from grayscale to black and white

% Source: https://www.mathworks.com/help/images/ref/im2bw.html

for num= 1363:1703 %the frames we are interested in

gray\_file=strcat('Gray\_Scale ','frame\_number', num2str(num),'.png');

I=imread(gray\_file);

blackwhite=im2bw(I); % No level specified; MATLAB uses 0.5

blackwhite\_filename=strcat('Black\_White','frame\_number', num2str(num),'.png');

imwrite(blackwhite, blackwhite\_filename);

end

MATLAB code for Coverting the Images to Black and White



Fig 2. Displaying the same frame as Fig1 in black-and-white (using imshow())

% Using imshow to display black and white April 18 photograph

imshow('Black\_Whiteframe\_number1416.png');

MATLAB Code for Fig 2.

**TASK 4**

**Q6.** For every black-and-white .png image, I will specify a certain section (section occupied by the date stamp) of the image using a range of pixels in both horizontal (pixels 73 to 291) and vertical (pixels 28 to 94) direction. Then I will change thos pixels to black, thus masking out the date stamp.   
Source: <https://www.mathworks.com/matlabcentral/answers/86410-changing-values-of-pixels-in-an-image-pixel-by-pixel-thresholding>

%% Masking out the date stamp

% Source: https://www.mathworks.com/matlabcentral/answers/86410-changing-%values-of-pixels-in-an-image-pixel-by-pixel-thresholdin

for number = 1363:1703; % the images we are working with

blackwhite\_filename=strcat('Black\_White','frame\_number', num2str(number),'.png');

my\_image =imread(blackwhite\_filename);

for R=28:94

for C=73:291

my\_image(R,C)=0;

end

end

masked\_BW\_filename=strcat(num2str(number), '.png');

imwrite(my\_image, masked\_BW\_filename);

end

MATLAB code for Q6



Fig 3. Displaying the same frame as Fig1 and Fig2 in black-and-white (using imshow())

%% Using imshow to display MASKED black and white April 18 photograph

imshow('1416.png');

MATLAB Code for Fig 3.

**TASK 5**

**Q7.** I used N= special('gaussian', [12, 12], 15);

I used it once per frame.

%% Gaussian Low-pass Filter

% Source: ` %https://www.mathworks.com/matlabcentral/newsreader/view\_thread/156894

for number = 1363:1703; % the images we are working with

masked\_BW\_filename=strcat(num2str(number), '.png');

masked\_image=imread(masked\_BW\_filename);

N = fspecial('gaussian', [12, 12], 15);

image=imfilter(masked\_image,N);

filtered\_name=strcat(num2str(number),'filter', '.png');

imwrite(image,filtered\_name);

end

MATLAB Code for Filtering



Fig 4. Displaying the same frame as Fig1 – Fig3 in black-and-white (after filtering) (using imshow())

%% Using imshow to display FILTERED black and white April 18 photograph

imshow('1416filter.png');

MATLAB Code for Fig 4.

**Q8.** The value of the 2-D DFT is <12 x 12>



**Shown Above:** Plot of the fft

%% Finding 2-D DFT of averaging filter

N = fspecial('gaussian', [12, 12], 15);

fft=fft2(N);

MATLAB code for Q8.

**TASK 6**

*Please see the next page…*



Figure 5: Plot of change in ice-area

%% Counting white pixels in Day 1

% Source: https://www.mathworks.com/matlabcentral/answers/12011-count-black-and-white-pixels-on-a-image

image\_1=imread('1363filter.png');

whitepix\_1=0;

blackpix\_1=0;

for R=1:720

for C=1:1280

if image\_1(R,C)==1;

whitepix\_1=whitepix\_1+1;

else

blackpix\_1=blackpix\_1+1

end

end

end

area\_1=whitepix\_1;

% Caclulating the percentage change in ice by counting white pixels Source:

% https://www.mathworks.com/matlabcentral/answers/12011-count-black-and-white-pixels-on-a-image

for number = 1363:1703; % the images we are working with

filtered\_name=strcat(num2str(number),'filter', '.png');

image\_x=imread(filtered\_name);

whitepix=0;

blackpix=0;

day=number-1363+1;

index=0;

for R=1:720

for C=1:1280

if image\_x(R,C)==1;

whitepix=whitepix+1;

else

blackpix=blackpix+1;

end

end

end

area=whitepix

ice\_change\_tag= ((area-area\_1)/area\_1)\*100;

% Source:

% https://www.mathworks.com/matlabcentral/answers/113171-how-to-store-values-from-a-loop

y\_val(day)=ice\_change\_tag;

end

x=linspace(1,183,341);

figure

plot(x,y\_val);

% Source:

% https://www.mathworks.com/help/matlab/creating\_plots/add-title-axis-labels-and-legend-to-graph.html

xlabel('Days');

ylabel('Percentage change in ice-area');

MATLAB Code For Figure 5

**Q9.** The ice area has quite steadily increased. By the end of this time period, the ice area has increased by about 60%

**TASK 7**



Fig 6: The plot and polynomial regression (I intentionally left it small) (polynomial is in red)

**Q10.** I used a third order polynomial. The polynomial was:

-0.000003x3  - 0.000871566168956669x2 + 0.603560779036551x 1.09784583678310

%% Using polyfit to model ice change as polynomial regression

poly=polyfit(x,y\_val,3); % n=3

val=polyval(poly,x);

% Source:

% https://www.mathworks.com/help/matlab/creating\_plots/add-title-axis-labels-and-legend-to-graph.html

figure

xlabel('Days');

ylabel('Percentage change in ice-area');

plot(x,y\_val);

hold

% Source: https://www.mathworks.com/matlabcentral/newsreader/view\_thread/2131

plot(x,val, 'Color', 'red');

exp=-3.29472397891272e-06;

MATLAB Code For TASK 7

**Q11** September 30th is 11 days after September 19th. So in our case, it is Day(183+11)= Day 194

Thus, plugging into the polynomial regression we have in Q10, the ice coverage is 63.422% more than the start date.

**TASK 8**

Fig: Plot showing change in Ice-extent over the year

**Q12.** Minimum Ice area: 3.548 million square kilometers, February 21

Maximum Icea area: 20.178 million square kilometers, September 18.

The ice area is low at the start and end of the year (summer in southern hemispere)

The ice area is high at the middle of year (winter in southern hemispere)

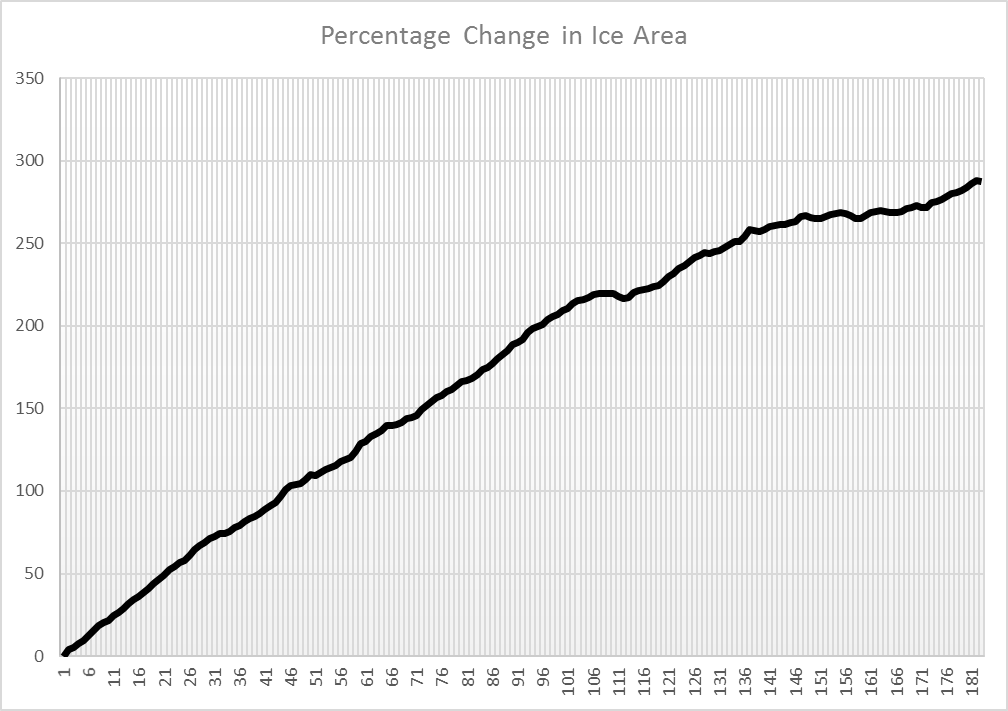


Fig 8. Percentage change in ice extent from Dataset

**Q13.**  The results are not very compatible. The graphs look the same, but the final percentage change in ice area are vastly different. From counting pixels, we have an approximate 60% change, but from the dataset, we about 287.5%. The possible causes are

1. **Image size:** Considering that we are dealing with area the order of million square kilometres, it is very difficult to get an acurate estimate using a 720x1280 image, because of the scale.
2. **Imperfect Filter:** My low-pass filter may have been imperfect, and may have left some black pixels uncovered, and destroyed some white pixels at the edges .

**Q14.** Other techniques include using functions like bwarea, regionprops. Source: <https://www.mathworks.com/help/images/pixel-values-and-image-statistics.html>

We can also use imshowpair. Source: <https://www.mathworks.com/help/images/ref/imshowpair.html>

**APPENDIX: The MATLAB Script**

%% Creating a VideoReader Object

vidObj = VideoReader('NASA.mp4');

get(vidObj);

% Source: <http://stackoverflow.com/questions/31932380/why-is-matlab-unable-to-determine-the-number-of-frames-in-a-video-file>

% We need to read the last frame

data = read(vidObj, Inf);

numFrames = vidObj.NumberOfFrames;

%% Determining Height and Width of Frames

% Source: https://www.mathworks.com/help/matlab/import\_export/read-video-files.html

vidHeight = vidObj.Height;

vidWidth = vidObj.Width;

%% Extracting all the frames onto my drive

% I will then use the ones I need

% Source: https://www.youtube.com/watch?v=AI-1ch6CHkI&t=295s

for img = 1:vidObj.NumberOfFrames;

filename=strcat('frame\_number', num2str(img), '.png');

frame=read(vidObj, img);

imwrite(frame, filename);

end

%% Converting from RGB to Grayscale and writing on drive

% Source: https://www.mathworks.com/help/matlab/ref/rgb2gray.html

for num= 1363:1703 %the frames we are interested in

file=strcat('frame\_number', num2str(num), '.png');

RGB=imread(file);

gray\_file=strcat('Gray\_Scale ','frame\_number', num2str(num),'.png');

gray\_image=rgb2gray(RGB);

imwrite(gray\_image, gray\_file);

end

% Using imshow to display April 18 Photograph

imshow('Gray\_Scaleframe\_number1416.png');

%% To mask out date stamp, use pixels X= 81:292

%%Y = 41:100

%% Changing from grayscale to black and white

% Source: https://www.mathworks.com/help/images/ref/im2bw.html

for num= 1363:1703 %the frames we are interested in

gray\_file=strcat('Gray\_Scale ','frame\_number', num2str(num),'.png');

I=imread(gray\_file);

blackwhite=im2bw(I); % No level specified; MATLAB uses 0.5

blackwhite\_filename=strcat('Black\_White','frame\_number', num2str(num),'.png');

imwrite(blackwhite, blackwhite\_filename);

end

%% Using imshow to display black and white April 18 photograph

imshow('Black\_Whiteframe\_number1416.png');

%% Masking out the date stamp

% Source: https://www.mathworks.com/matlabcentral/answers/86410-changing-values-of-pixels-in-an-image-pixel-by-pixel-thresholdin

for number = 1363:1703; % the images we are working with

blackwhite\_filename=strcat('Black\_White','frame\_number', num2str(number),'.png');

my\_image =imread(blackwhite\_filename);

for R=28:94

for C=73:291

my\_image(R,C)=0;

end

end

masked\_BW\_filename=strcat(num2str(number), '.png');

imwrite(my\_image, masked\_BW\_filename);

end

%% Using imshow to display MASKED black and white April 18 photograph

imshow('1416.png');

%% Gaussian Low-pass Filter

% Source: https://www.mathworks.com/matlabcentral/newsreader/view\_thread/15689

for number = 1363:1703; % the images we are working with

masked\_BW\_filename=strcat(num2str(number), '.png');

masked\_image=imread(masked\_BW\_filename);

N = fspecial('gaussian', [12, 12], 15);%

image=imfilter(masked\_image,N);

filtered\_name=strcat(num2str(number),'filter', '.png');

imwrite(image,filtered\_name);

end

Using imshow to display FILTERED black and white April 18 photograph

imshow('1416filter.png');

Finding 2-D DFT of averaging filter

fft=fft2(N);

%% Counting white pixels in Day 1

%Source: https://www.mathworks.com/matlabcentral/answers/12011-count-black-and-white-pixels-on-a-image

image\_1=imread('1363filter.png');

whitepix\_1=0;

blackpix\_1=0;

for R=1:720

for C=1:1280

if image\_1(R,C)==1;

whitepix\_1=whitepix\_1+1;

else

blackpix\_1=blackpix\_1+1

end

end

end

area\_1=whitepix\_1;

% Caclulating the percentage change in ice by counting white pixels Source:

% https://www.mathworks.com/matlabcentral/answers/12011-count-black-and-white-pixels-on-a-image

for number = 1363:1703; % the images we are working with

filtered\_name=strcat(num2str(number),'filter', '.png');

image\_x=imread(filtered\_name);

whitepix=0;

blackpix=0;

day=number-1363+1;

index=0;

for R=1:720

for C=1:1280

if image\_x(R,C)==1;

whitepix=whitepix+1;

else

blackpix=blackpix+1;

end

end

end

area=whitepix

ice\_change\_tag= ((area-area\_1)/area\_1)\*100;

% Source:

% https://www.mathworks.com/matlabcentral/answers/113171-how-to-store-values-from-a-loop

y\_val(day)=ice\_change\_tag;

end

x=linspace(1,183,341);

figure

plot(x,y\_val);

%% Using polyfit to model ice change as polynomial regression

poly=polyfit(x,y\_val,3); % n=3

val=polyval(poly,x);

% Source: https://www.mathworks.com/help/matlab/creating\_plots/add-title-axis-labels-and-legend-to-graph.html

figure

xlabel('Days');

ylabel('Percentage change in ice-area');

plot(x,y\_val);

hold % Source: https://www.mathworks.com/matlabcentral/newsreader/view\_thread/2131

plot(x,val, 'Color', 'red');