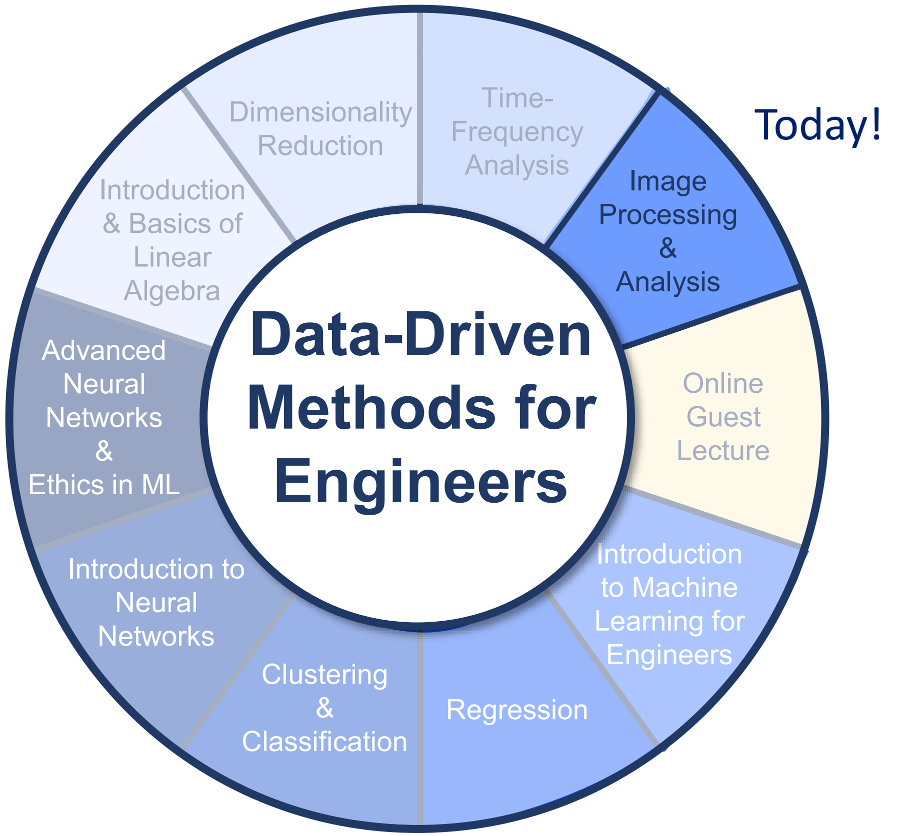
Data-Driven Methods for Engineers (MECH0107) 2025 - 2026

Dr Lama Hamadeh
Module Lead
Office 429 | Roberts Building
Mechanical Engineering Department
l.hamadeh@ucl.ac.uk

Dr Llewellyn Morse
Module Tutor
Office 503D| Roberts Building
Mechanical Engineering Department
l.morse@ucl.ac.uk

Module Lectures



Introduction

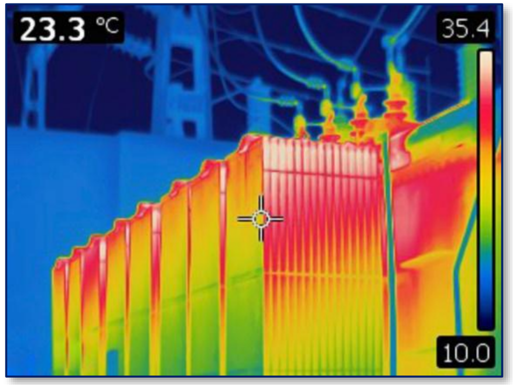
- Image data today is massive because of high-resolution sensors, cameras, satellites, and the growth of AI applications. This data is in the petabyte to exabyte scale worldwide — one of the biggest drivers of Big Data!

~1 million gigabytes

~1 billion gigabytes

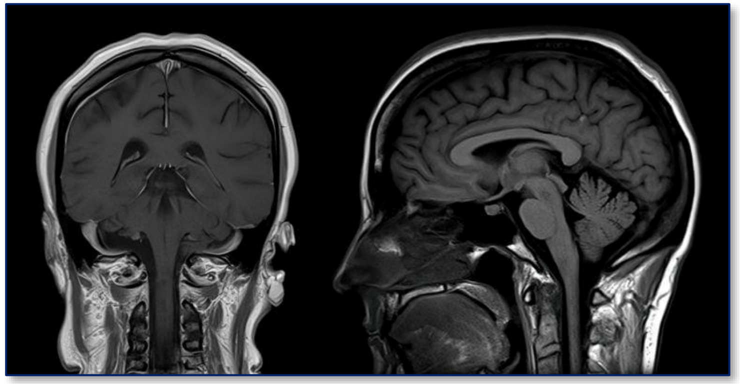
Ultrasound

Infrared/thermal imaging

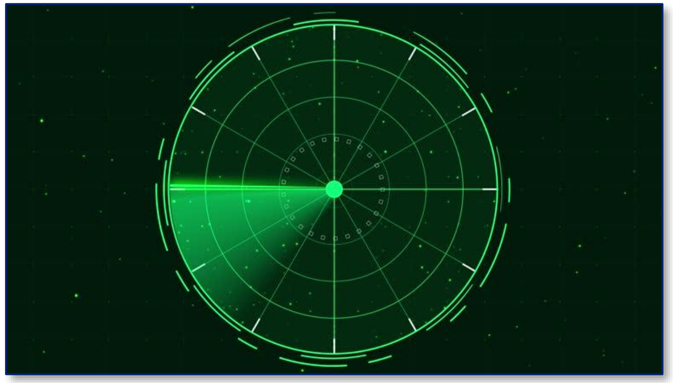


Tomography (CAT scan)

Magnetic resonance imaging (MRI)



Radar and sonar imaging



Digital photos.

Introduction

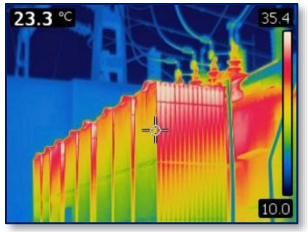
- **Image Processing** & **Analysis** is the field of study that deals with techniques to enhance, manipulate, and extract useful information from images.

- **Image Processing** focuses on improving image quality or transforming images (e.g., noise reduction, sharpening, contrast adjustment, edge detection) - *make the image better.*

- **Image Analysis** goes one step further: it interprets and extracts meaningful information from the image (e.g., object detection, segmentation, pattern recognition, measurements) - *understand what's in the image.*

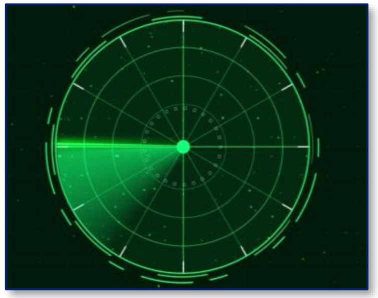
Ultrasound

Infrared/thermal imaging

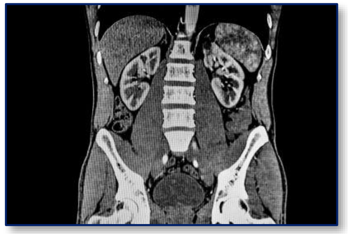


Magnetic resonance imaging (MRI)

Radar and sonar imaging



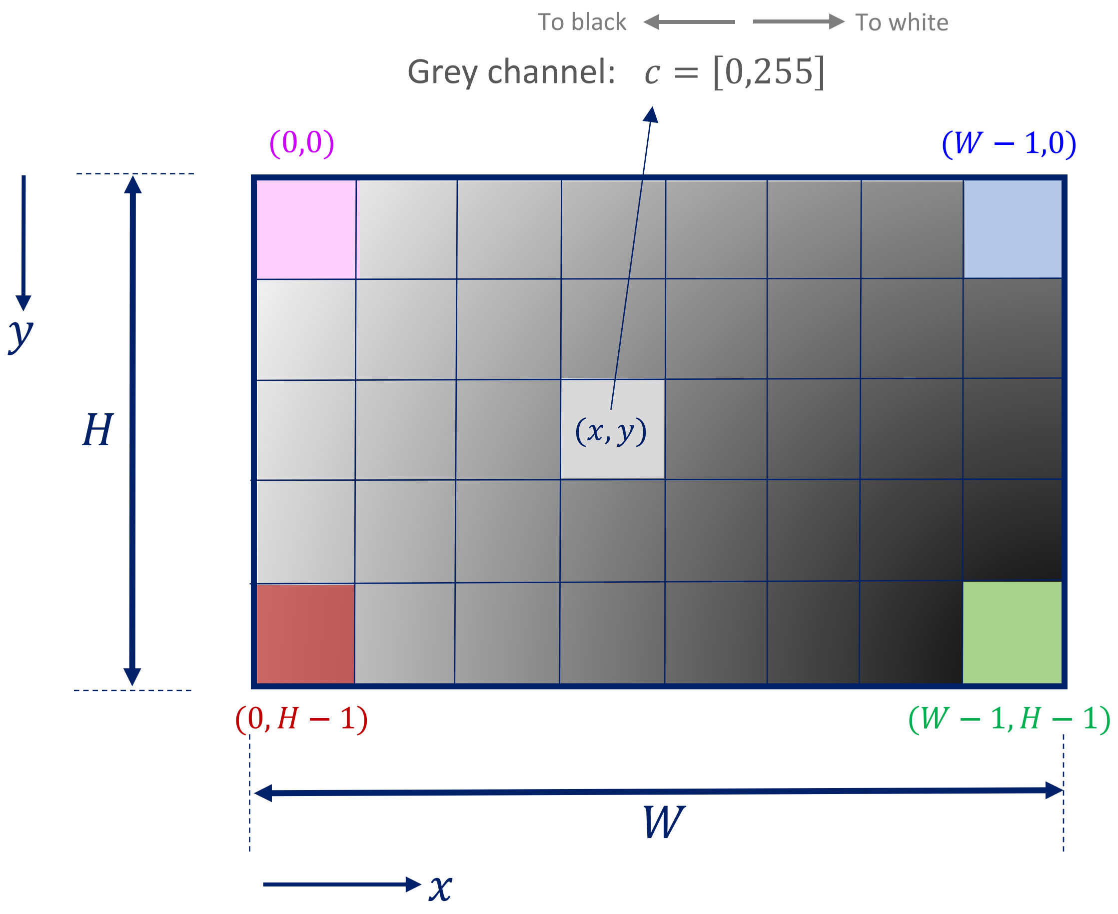
Tomography (CAT scan)



Digital photos.

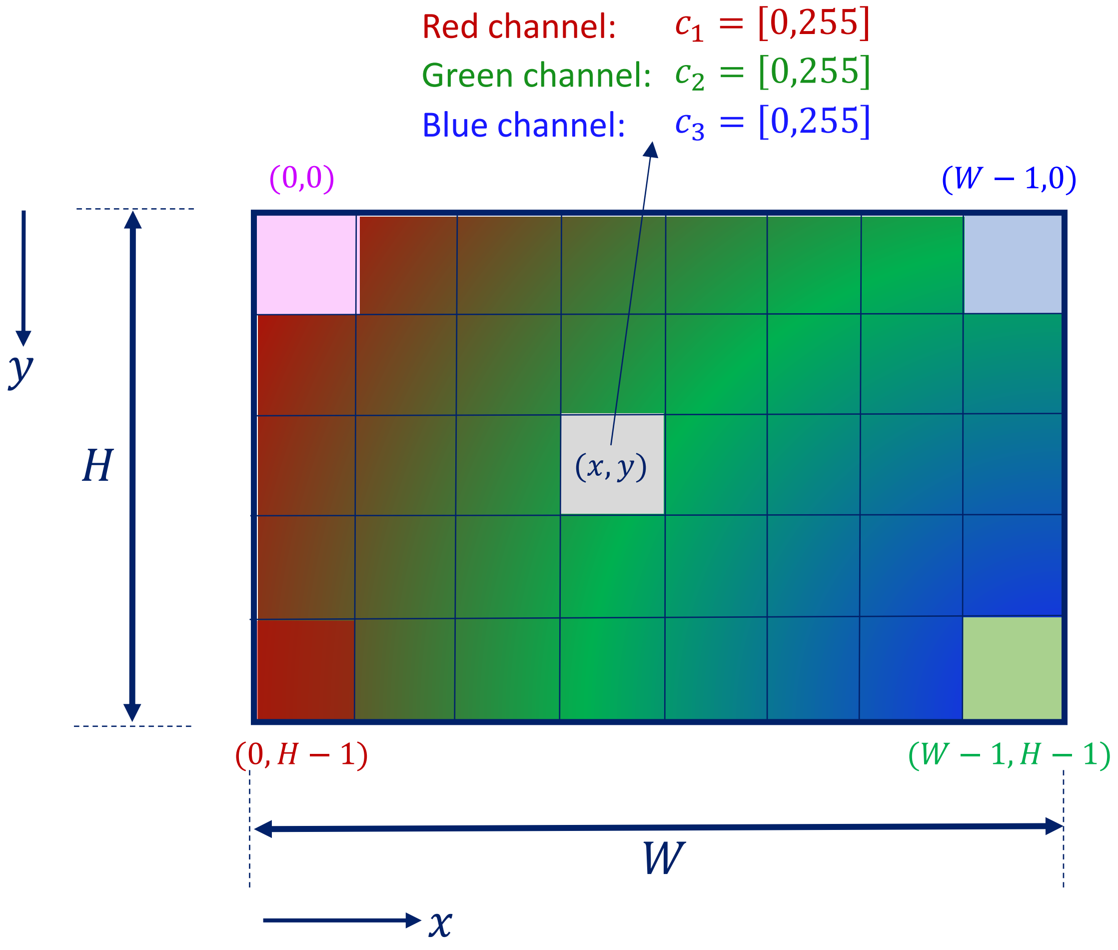
Fundamentals of Imaging
Image Dimensions: Pixels & Colours

- Each image is a 2D array (generally a rectangular array) of pixels (short for picture elements), and each pixel contains information.
- The address of each pixel within the image is usually specified as an (x, y) pair, with x indicating the distance from the left edge and y indicating the distance down from the top.
- The information associated with each pixel could be a single value (or a single channel) that ranges from 0 to 255, representing the **gray-scale (monochrome) brightness of that point in a scene**.



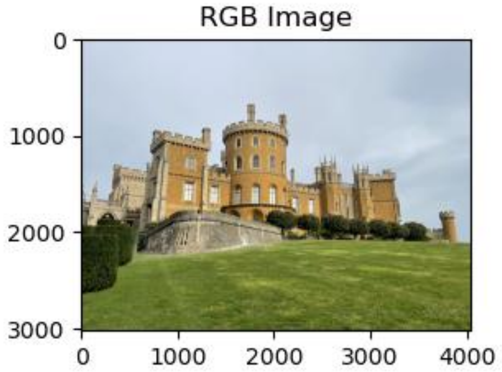
Fundamentals of Imaging
Image Dimensions: Pixels & Colours

- Each image is a 2D array (generally a rectangular array) of pixels (short for picture elements), and each pixel contains information.
- The address of each pixel within the image is usually specified as an (x, y) pair, with x indicating the distance from the left edge and y indicating the distance down from the top.
- The colour information could also be a **combination of red, green, and blue (RGB) values (or three channels)** where each channel has values that range between 0 and 255.



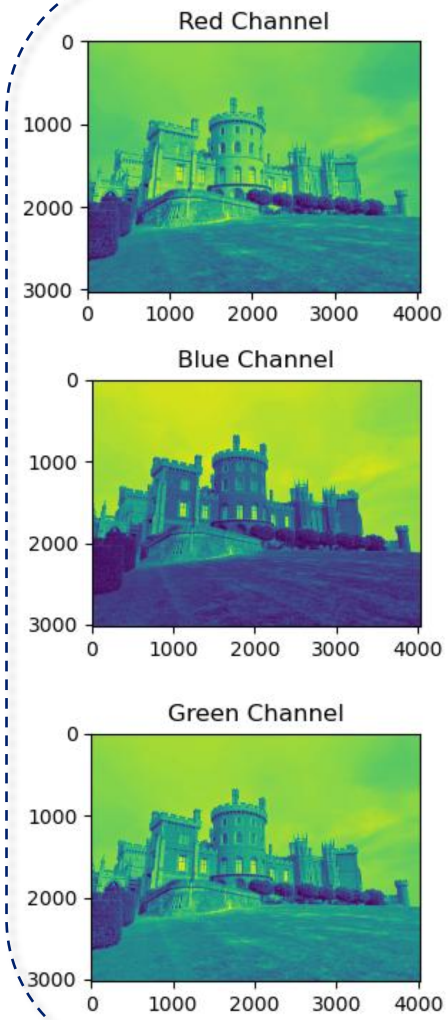
Fundamentals of Imaging
Image Dimensions: Pixels & Colours

Belvoir Castle




```
%% Read RGB Image

%import image
A = imread('Belvoir Castle','jpeg');

%show the coloured image
figure;
imshow(A)

%convert the image from uint8 to double (matrix)
A2 = double(A);

%check the size: height, width, and colour channels
size = size(A2);
%--------------------
```

```
size =
     height     width     Colours
      3024       4032        3
```

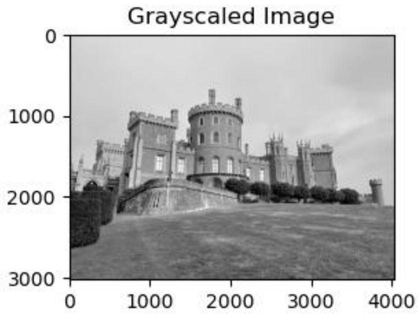



```
%show the colour channels
figure;
subplot(3,1,1)
imshow(A(:,:,1)) %Red

subplot(3,1,2)
imshow(A(:,:,2)) %Green

subplot(3,1,3)
imshow(A(:,:,3)) %Blue
%--------------------
```

RGB image is a good idea when colour itself encodes information (e.g. medical images, material identification, traffic lights, heat maps).

more channels,
higher computation

It's a trade-off, there is not a universal rule.




```
%% Convert the image from RGB to gray scale

%convert image to black and white
Abw = rgb2gray(A);

%show the grayscaled image
figure;
imshow(Abw)
%--------------------
```

Grayscale is a good idea:
- When colour does not carry meaningful information, i.e., colour is redundant for the task.
- When you want simpler and faster analysis.
- When lighting dominates more than colour differences.

fewer channels,
less computation

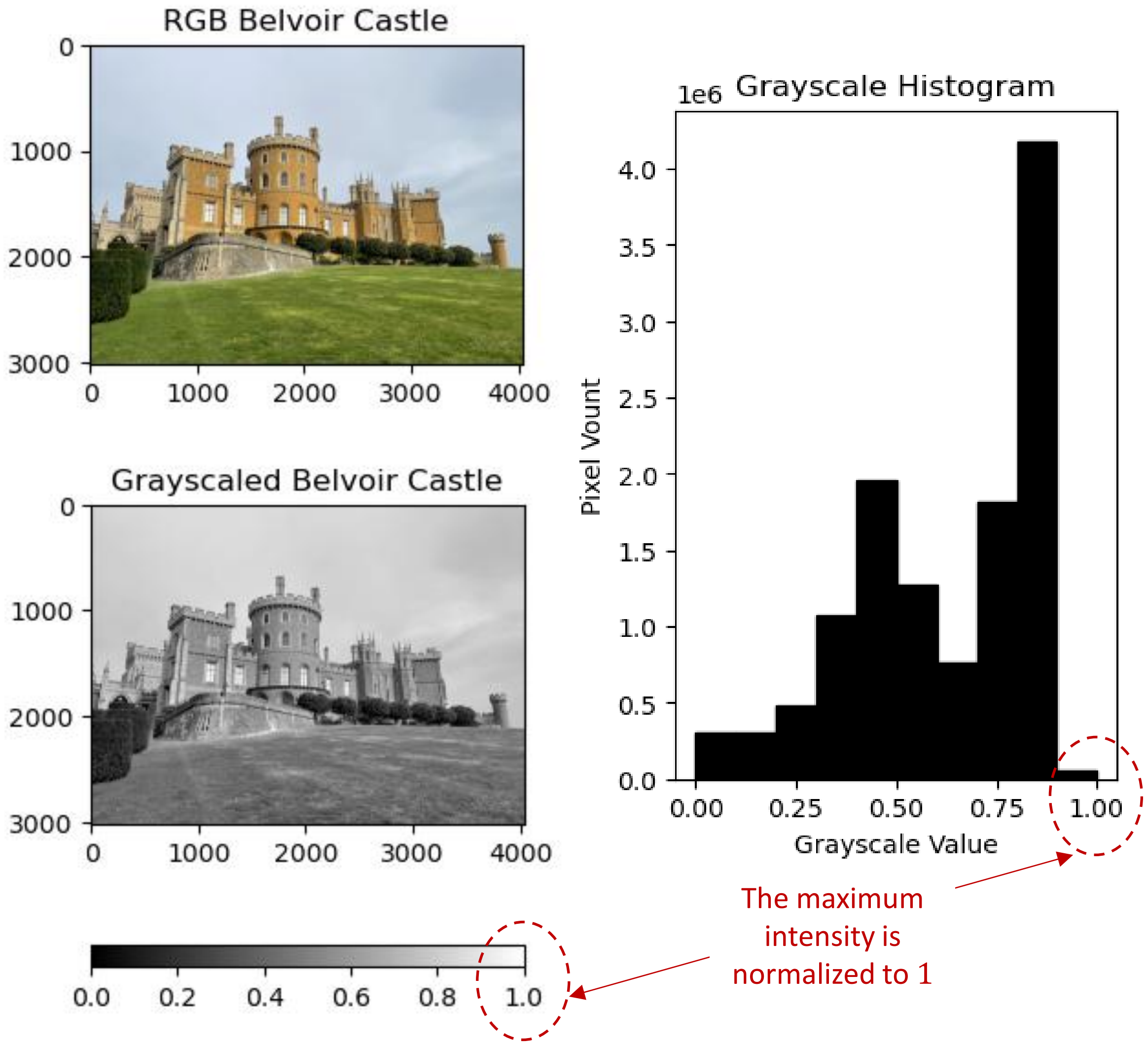
MECHANICAL ENGINEERING

Fundamentals of Imaging
Measurements & Histograms

- An **image histogram** is a graph that shows how pixel intensities (brightness values) are distributed in an image. It is a summary of **how dark or bright an image is** and how contrast is spread across it.

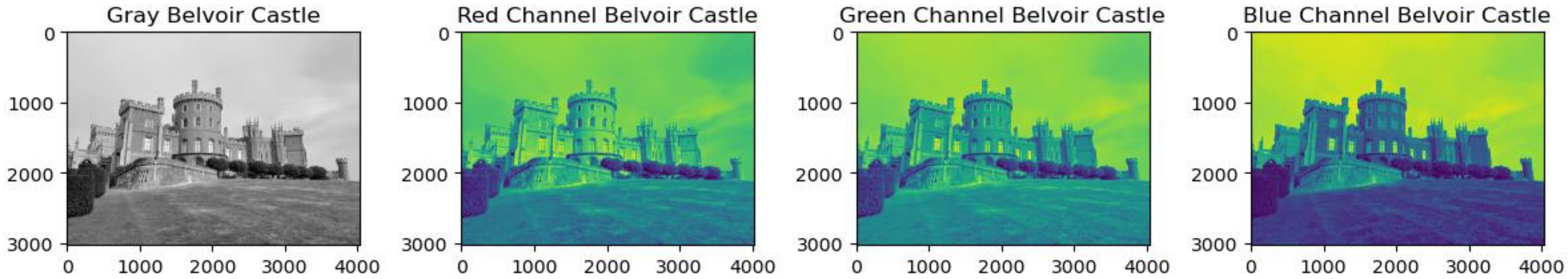
- ***x*-axis**: pixel values (e.g., 0 = black, 255 = white for grayscale images).
  ***y*-axis**: number of pixels having that intensity.

- Image histograms are useful for quickly tells whether an image is dark, bright, low-contrast, or well-balanced.



Fundamentals of Imaging
Measurements & Histograms

For the same image, can we get different histograms?



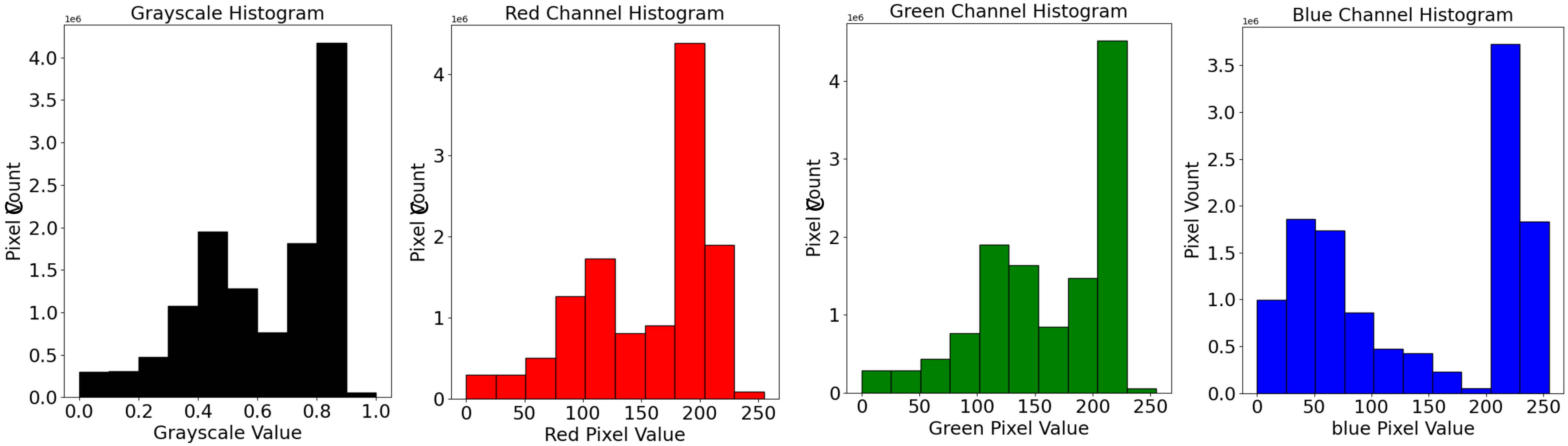


Image Denoising

- Image filtering (or denoising) is the process of removing **unwanted noise** or enhancing certain features in an image **to improve its quality** or make it more useful for analysis. Noise can come from many sources such as poor lighting, sensor imperfections, or transmission errors.

- Main Types of Filtering:

  Linear Filtering – each pixel is replaced by a weighted average of its neighbors. Good for smoothing and reducing random noise. Example: Mean filter, Gaussian filter, Shannon filter.

  Non-Linear Filtering – works on pixel intensity relationships without simple averaging. Better at preserving edges and details while reducing noise. Example: Median filter, Bilateral filter.

  Frequency-Domain Filtering – filtering is done after transforming the image into frequency space. Useful for removing periodic noise or enhancing certain frequency components. Example: Low-pass filter (for smoothing), High-pass filter (for sharpening).

Original Image

Denoised Image

Image Denoising
Gaussian Filter

- The **Gaussian filter** is a linear and frequency-domain filter widely used for smoothing and noise reduction.
- In the frequency domain, it acts as a **low-pass filter**, attenuating high-frequency components often associated with noise while preserving low-frequency components that carry the main image structures.
- Unlike ideal filters with sharp cutoffs, the Gaussian filter has a smooth frequency response, which reduces artifacts such as ringing.

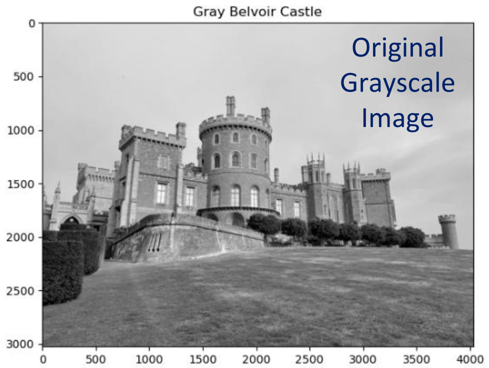
frequency coordinates

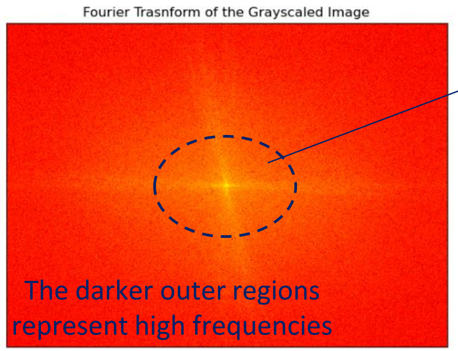
Center of frequency domain

$$G(k_x, k_y) = \exp\left(-\sigma_x (k_x - a)^2 - \sigma_y (k_y - b)^2\right)$$

filter widths (Large widths lead to more blurring)

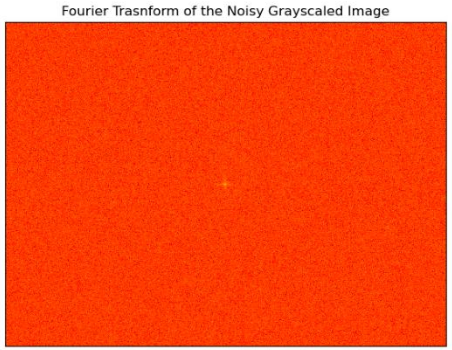
Image Denoising
Gaussian Filter





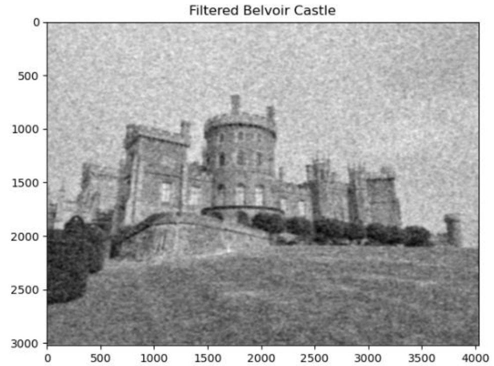
The bright centre represents low frequencies

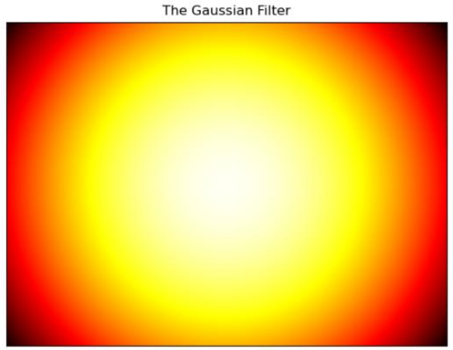
Random noise has been added. The image looks grainy, and fine details are harder to see.



The noise introduces many high-frequency components, making the frequency plot brighter and more spread out. This shows that noise mainly affects high frequencies.

After applying the **Gaussian filter**:
- Noise is reduced
- The image looks smoother
- Some sharp edges are softened (this is the trade-off)





This is the Gaussian filter itself.
- Bright in the centre -> **keeps low frequencies**
- Dark at the edges -> **suppresses high frequencies**

Image Denoising
Shannon Filter

- The **Shannon filter** is a linear and frequency-domain filter widely used for smoothing and noise reduction.

- The **Shannon (square) filter** is a type of ideal low-pass filter that attenuates high-frequency components, which often correspond to noise, while retaining low-frequency components that represent the main image structure

- In practice, the filter's frequency response has a **square (rectangular) shape**, allowing frequencies within a specified cutoff to pass unchanged and completely blocking higher frequencies. This **makes it effective for denoising images with predominantly high-frequency noise**.

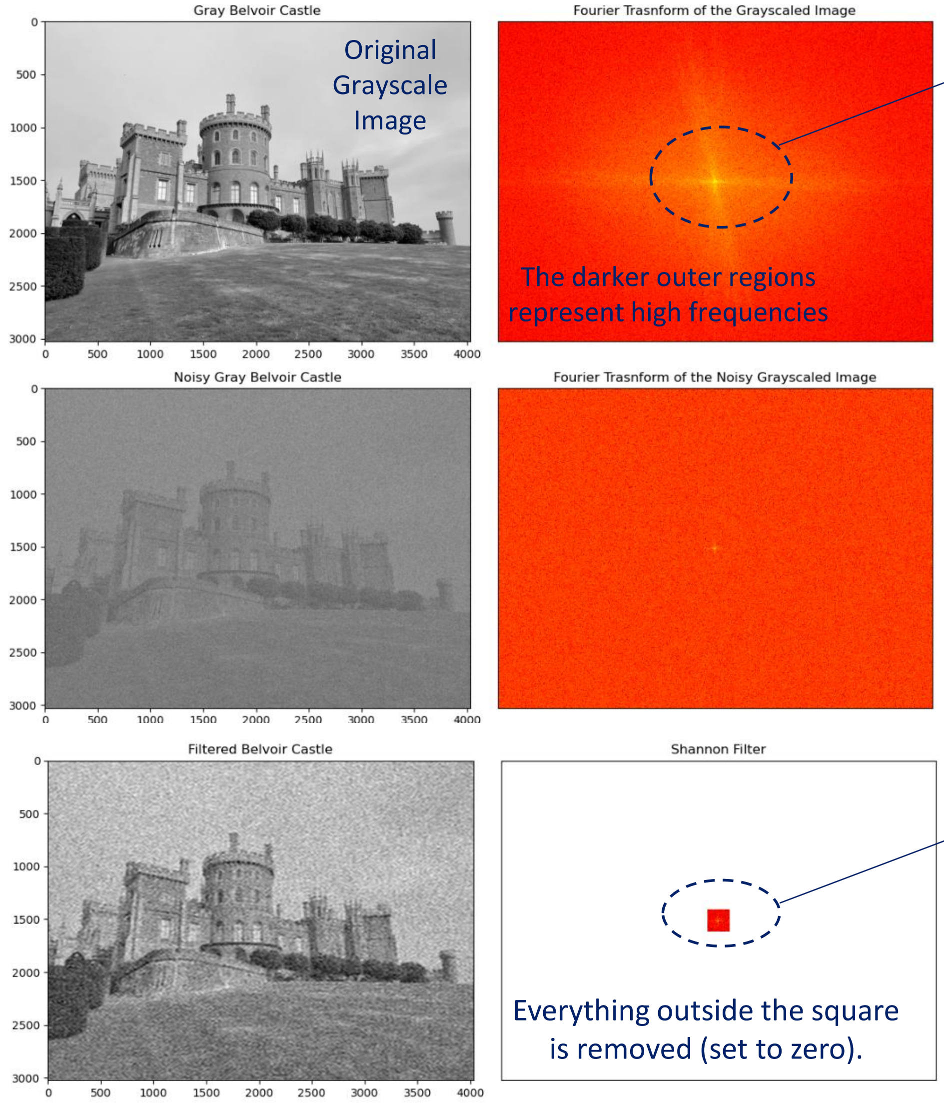
frequency coordinates

Center of frequency domain

$$S(k_x, k_y) = \begin{cases} 1 & \text{if} \quad |k_x - a| \leq D_0 \quad \text{and} \quad |k_y - b| \leq D_0 \\ 0 & \text{otherwise} \end{cases}$$

cutoff frequency along each axis, defining the square passband

Image Denoising Shannon Filter



The bright centre represents low frequencies

Random noise has been added. The image looks grainy, and fine details are harder to see.

The noise introduces many high-frequency components, making the frequency plot brighter and more spread out. This shows that noise mainly affects high frequencies.

After applying the **Shannon filter**:
- Noise is reduced
- The image looks grainy
- You may notice ringing or artefacts around edges (Gibbs phenomenon).

This happens because the sharp cutoff in frequency space introduces oscillations in the image domain.

The bright square in the centre represents the frequencies that are kept.

Unlike the Gaussian filter, the transition here is abrupt: frequencies are either fully kept or fully discarded.

Image Compression

- Image compression is the process of **reducing the amount of data required to represent an image** while preserving its visual quality as much as possible.
- The goal is to save storage space, reduce transmission time, and improve efficiency in applications such as multimedia, medical imaging, and industrial systems.
- Image compression changes **how information is stored**, **not how many pixels exist**. That's why a compressed image can look the same on screen but take much less storage space.
- Among the vast number of methods that are used in image compression applications, two methods will be presented: Singular Value Decomposition (SVD) and Fast Fourier Transform (FFT).

Original Gray

~MB in data

Image Compression

This is done by removing redundancy and discarding information the human eye is less sensitive to

Compressed Image

~KB in data

Image Compression
Singular Value Decomposition

- Singular Value Decomposition (SVD) is a mathematical technique from linear algebra that can be applied to image compression by exploiting redundancy in pixel data. An image can be represented as a matrix $X(m \times n)$ (grayscale) or multiple matrices $X(m \times n \times 3)$ (for RGB channels). Using SVD, the matrix is factorized as:

$$X = U\Sigma V^*$$

diagonal matrix with **singular values** arranged in decreasing order (contain the image's energy/importance)

orthogonal matrices (contain spatial information)

- Most of the important image information is concentrated in the **first few singular values**. By keeping only the top r singular values and discarding the rest, we can approximate the original image:

$$X_{\text{Approx}} = U_r \Sigma_r V_r^*$$

- This reduces storage because instead of saving the full matrix, **we only store the first r singular values** and corresponding vectors.

Image Compression
Singular Value Decomposition

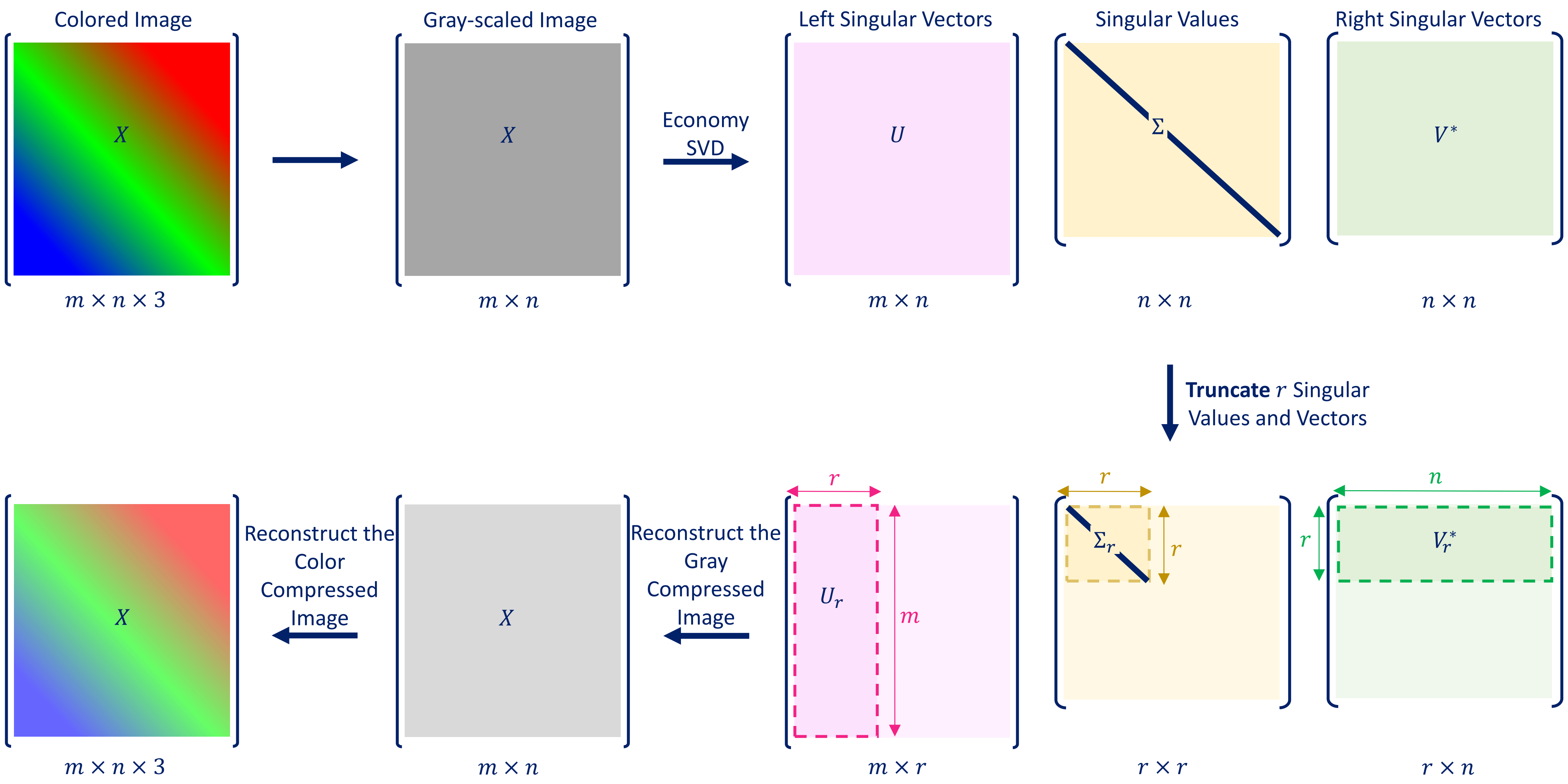


Image Compression
Singular Value Decomposition

Image Compression
Singular Value Decomposition

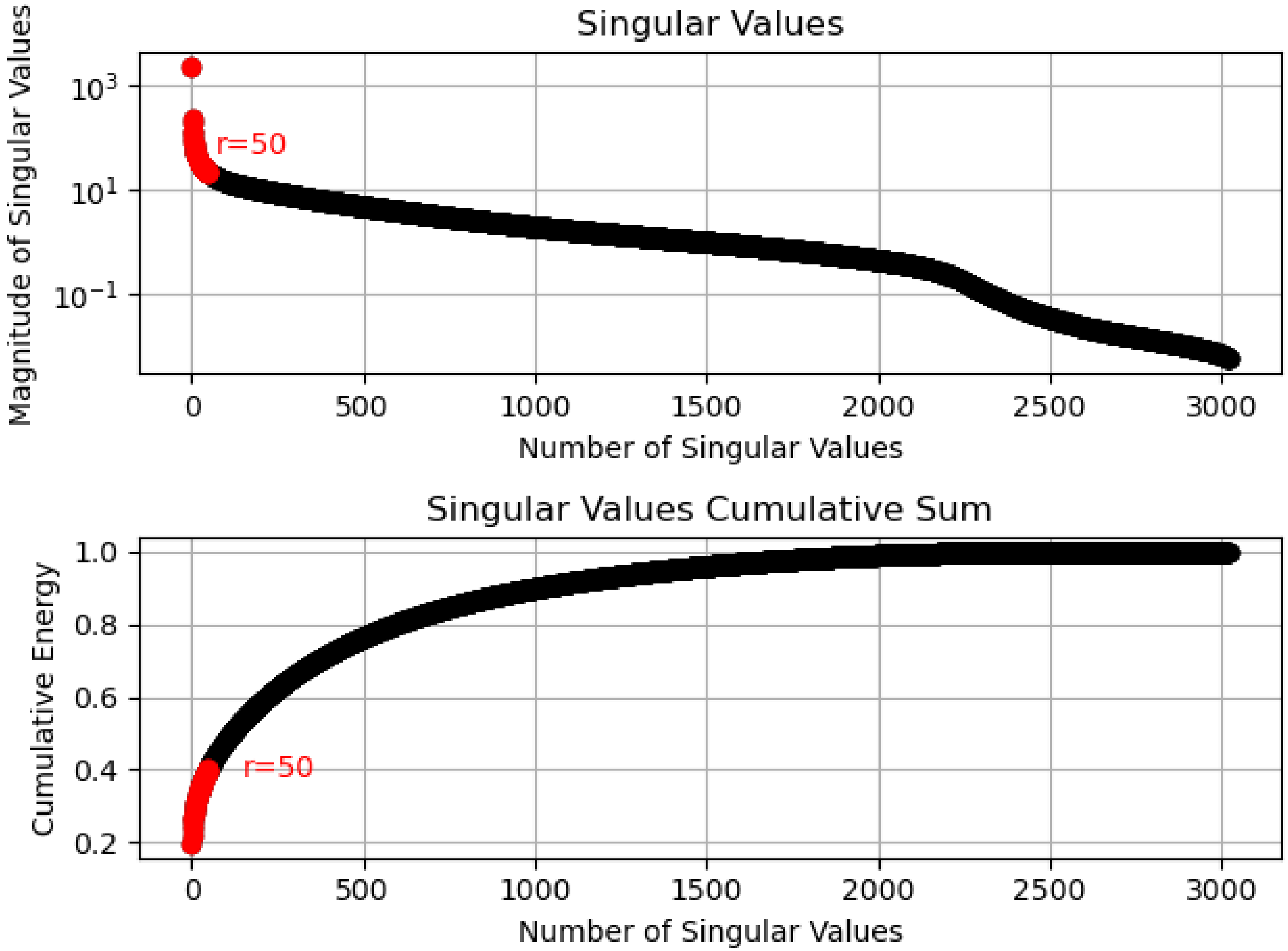


Image Compression
Fast Fourier Transform

- In image compression, the FFT is used to exploit the fact that **most of the important visual information in an image lies in the low-frequency components** (near the center of the frequency domain), while high-frequency components often correspond to fine details or noise.

- The idea in using FFT in image compression is to retain only the largest/low-frequency coefficients and discard the small/high-frequency coefficients (less perceptible to the human eye). The compressed image can be reconstructed by using the **Inverse FFT** from the retained coefficients

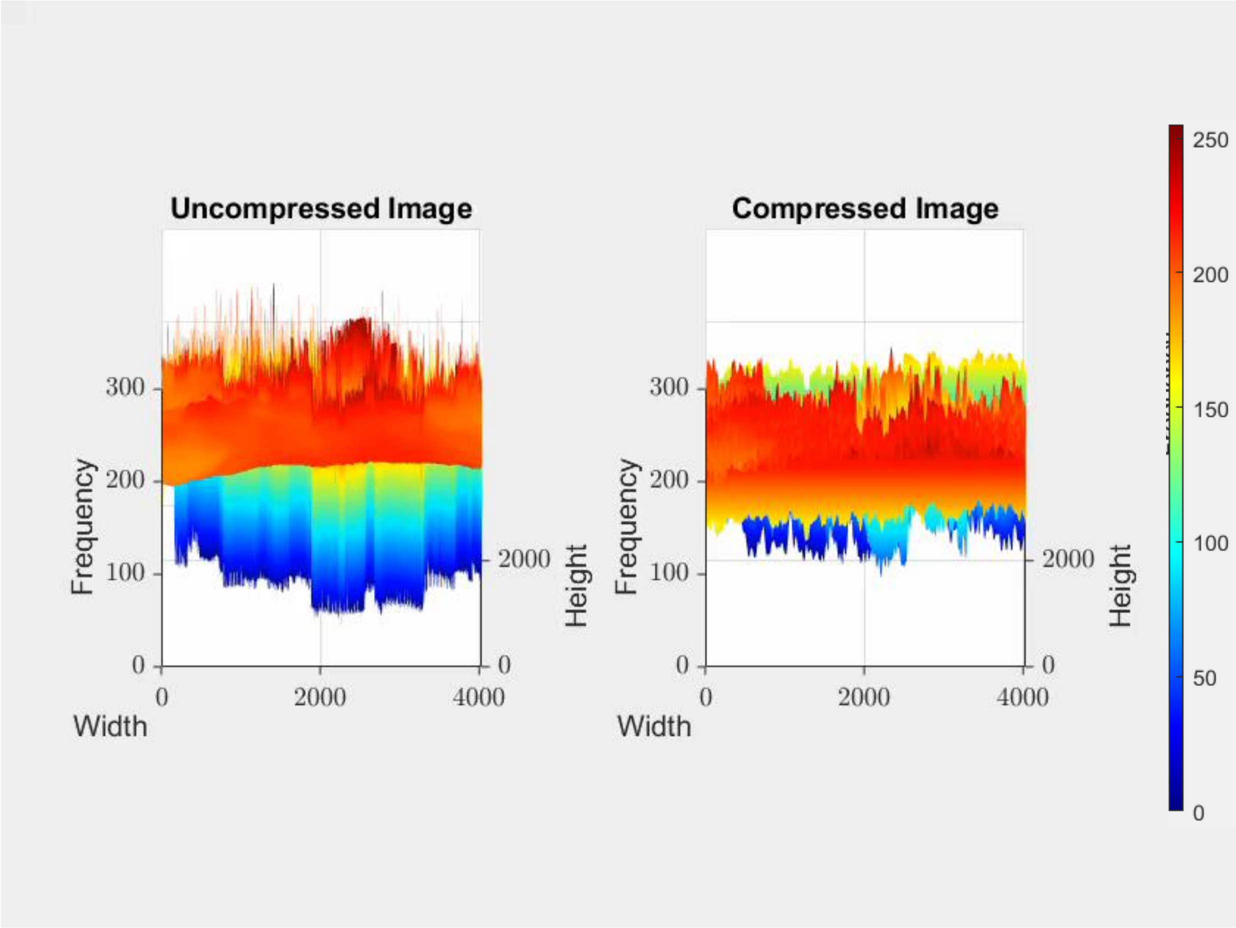


Image Compression
Fast Fourier Transform

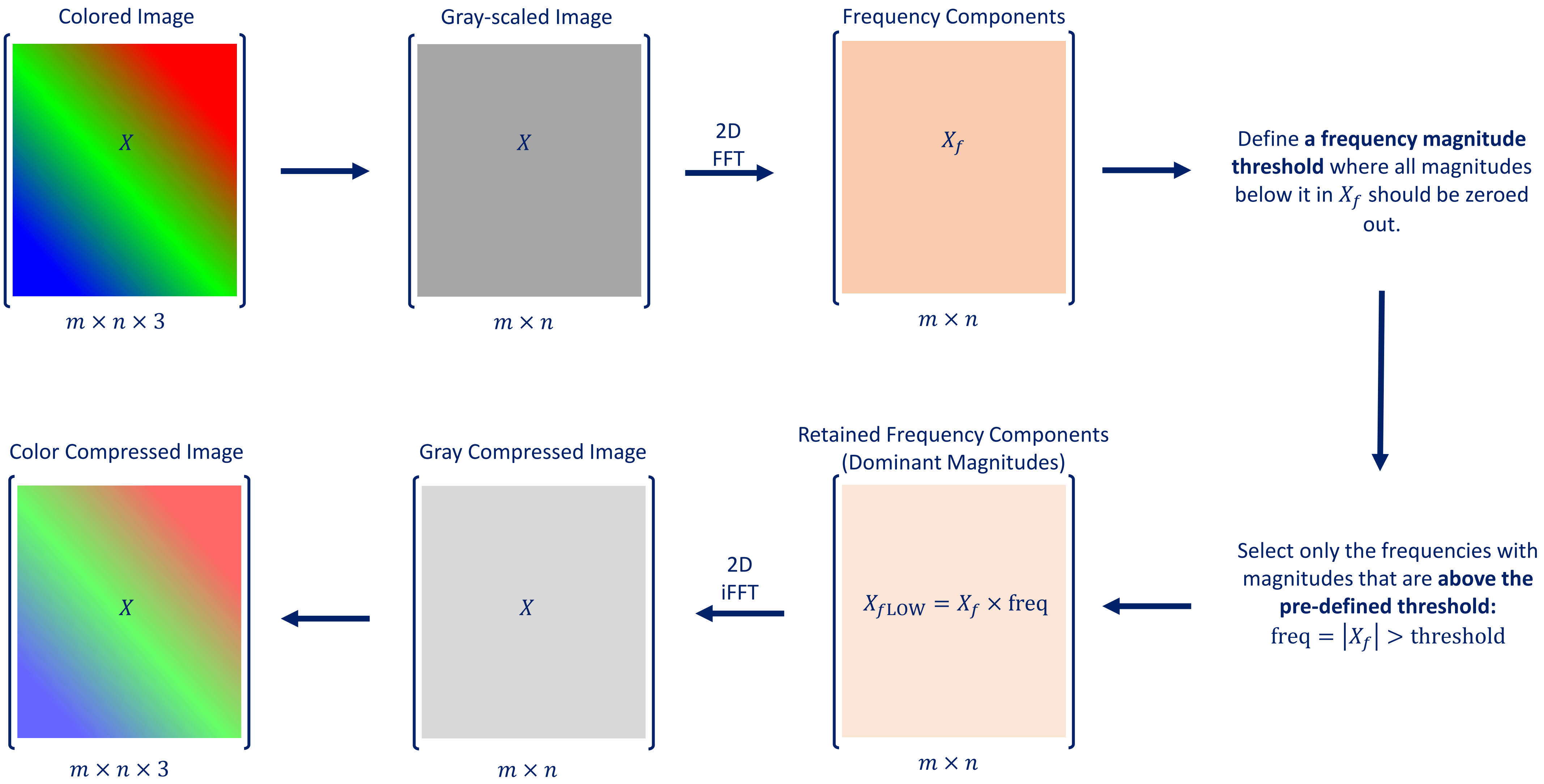
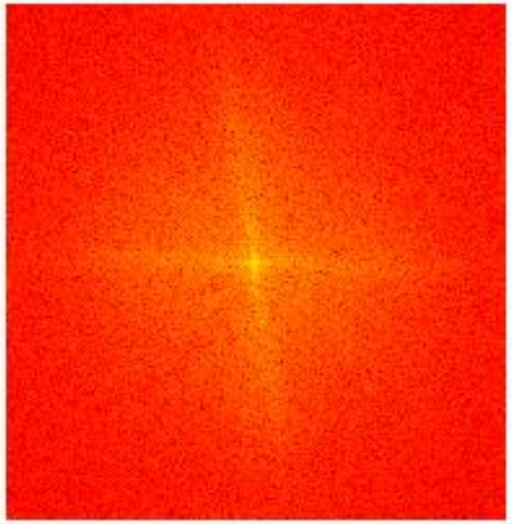


Image Compression
Fast Fourier Transform

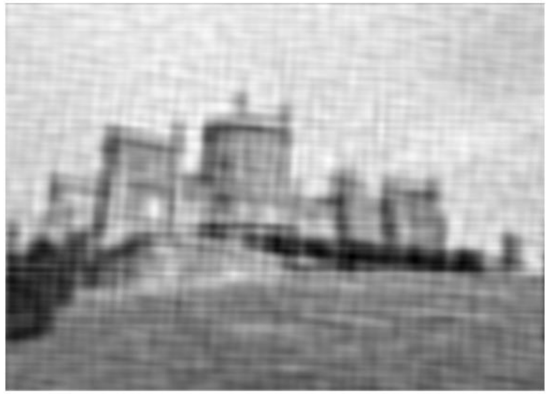
Original RGB

Original Gray

Fourier Trasnform



0.01% of FFT Basis



0.05% of FFT Basis

0.10% of FFT Basis

1% of FFT Basis

Edge Detection

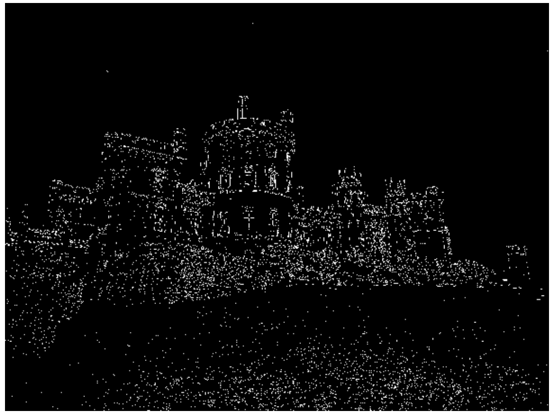
- **Edge detection** is a fundamental technique in image processing and computer vision used to identify points in an image where the intensity changes sharply. These points, called **edges**, correspond to important structural features such as **object boundaries, textures, and surface discontinuities**.

- Detecting edges simplifies image analysis by reducing the amount of data while preserving essential shape and structural information.

- The location of edges in a 2D image is normally determined either by:
  - finding the image **gradient** extrema (maximum or minimum), i.e., taking the 1st derivative of the image
  - finding the zero-crossings of the **Laplacian** of the image, i.e., taking the 2nd derivative of the image.

Grayscaled Image

1st Order Derivative Image: Prewitt



Edge Detection

First Derivative Operators

- The **gradient** is a measure of change in a function, and an image can be considered to be an array of samples of some continuous function of image intensity, say $f(x,y)$. Therefore, the gradient of the image function is given by:

Vector quantity

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- The **magnitude of the gradient** (or the strength of the edge) is given by:

Scalar quantity

$$\underbrace{\|\nabla f(x,y)\|}_{} = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

Edges correspond to the local maxima of the gradient magnitude.

Second Derivative Operators

- **Second-order derivatives** have a stronger response than first derivatives to fine detail, such as thin lines, weak edges, and isolated points. The Laplacian is the two-dimensional equivalent of the second derivative $f(x,y)$:

Scalar quantity

$$\underbrace{\nabla^2 f(x,y)}_{} = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

Edges are detected at zero-crossings, i.e., when $\nabla^2 f(x,y) = 0$.

Edge Detection

Original RGB

Original Gray

Prewitt Operator 1st Derivative

Sobel Operator 1st Derivative

Laplacian Operator 2nd Derivative

UCL MECHANICAL ENGINEERING

Tutorial on Tuesday 10th Feb

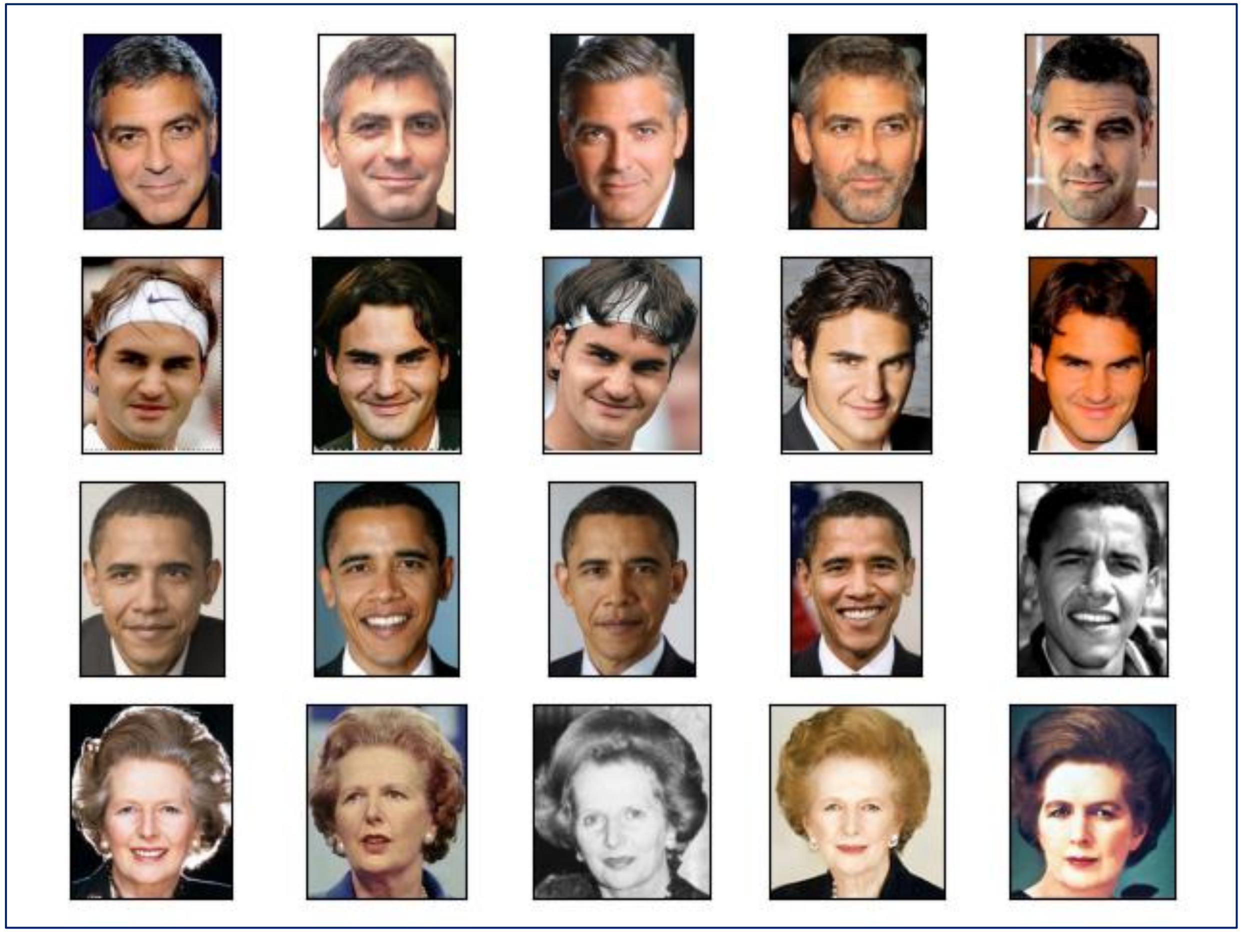
Eigenfaces for Recognition

George Clooney

Roger Federer

Barack Obama

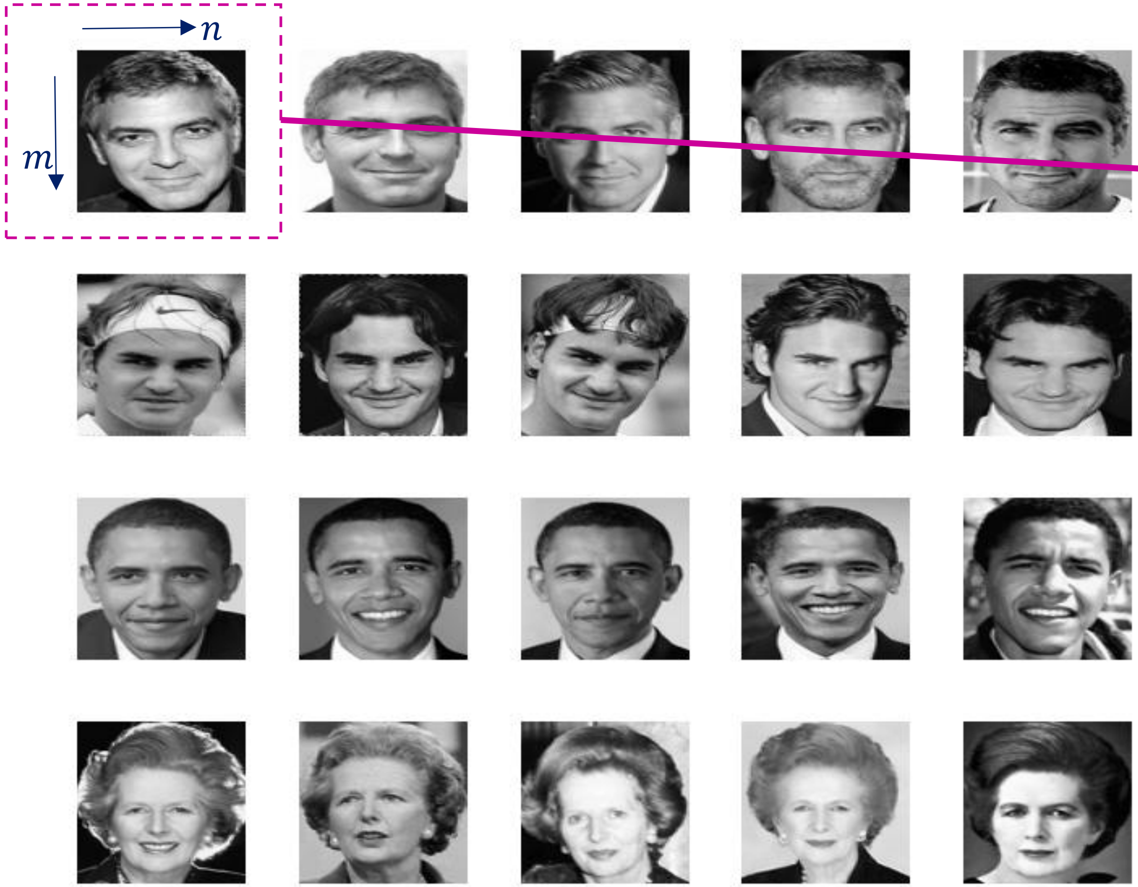
Margaret Thatcher



Training Images

Tutorial on Tuesday 10th Feb

Eigenfaces for Recognition




```
%% contrucst the PCA data matrix
% The rows correspond to the images so we have 20 rows as we havw 20 imags
D = [reshape(C1,1,m*n)
     reshape(C2,1,m*n)
     reshape(C3,1,m*n)
     reshape(C4,1,m*n)
     reshape(C5,1,m*n)
     reshape(F1,1,m*n)
     reshape(F2,1,m*n)
     reshape(F3,1,m*n)
     reshape(F4,1,m*n)
     reshape(F5,1,m*n)
     reshape(O1,1,m*n)
     reshape(O2,1,m*n)
     reshape(O3,1,m*n)
     reshape(O4,1,m*n)
     reshape(O5,1,m*n)
     reshape(T1,1,m*n)
     reshape(T2,1,m*n)
     reshape(T3,1,m*n)
     reshape(T4,1,m*n)
     reshape(T5,1,m*n)];
```

$$D = \begin{pmatrix} \text{Image 1} \\ \text{Image 2} \\ \cdot \\ \cdot \\ \cdot \\ \text{Image N} \end{pmatrix}$$

PCA-it!

Tutorial on Tuesday 10th Feb

Eigenfaces for Recognition

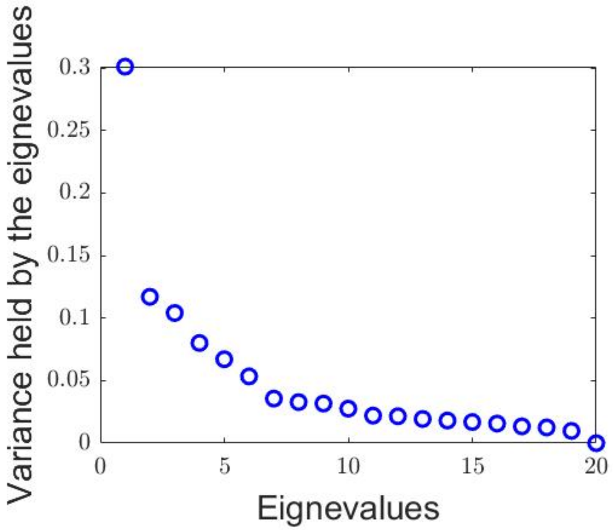
1st Eigenvector

2nd Eigenvector

3rd Eigenvector

4th Eigenvector

5th Eigenvector



Tutorial on Tuesday 10th Feb

Eigenfaces for Recognition

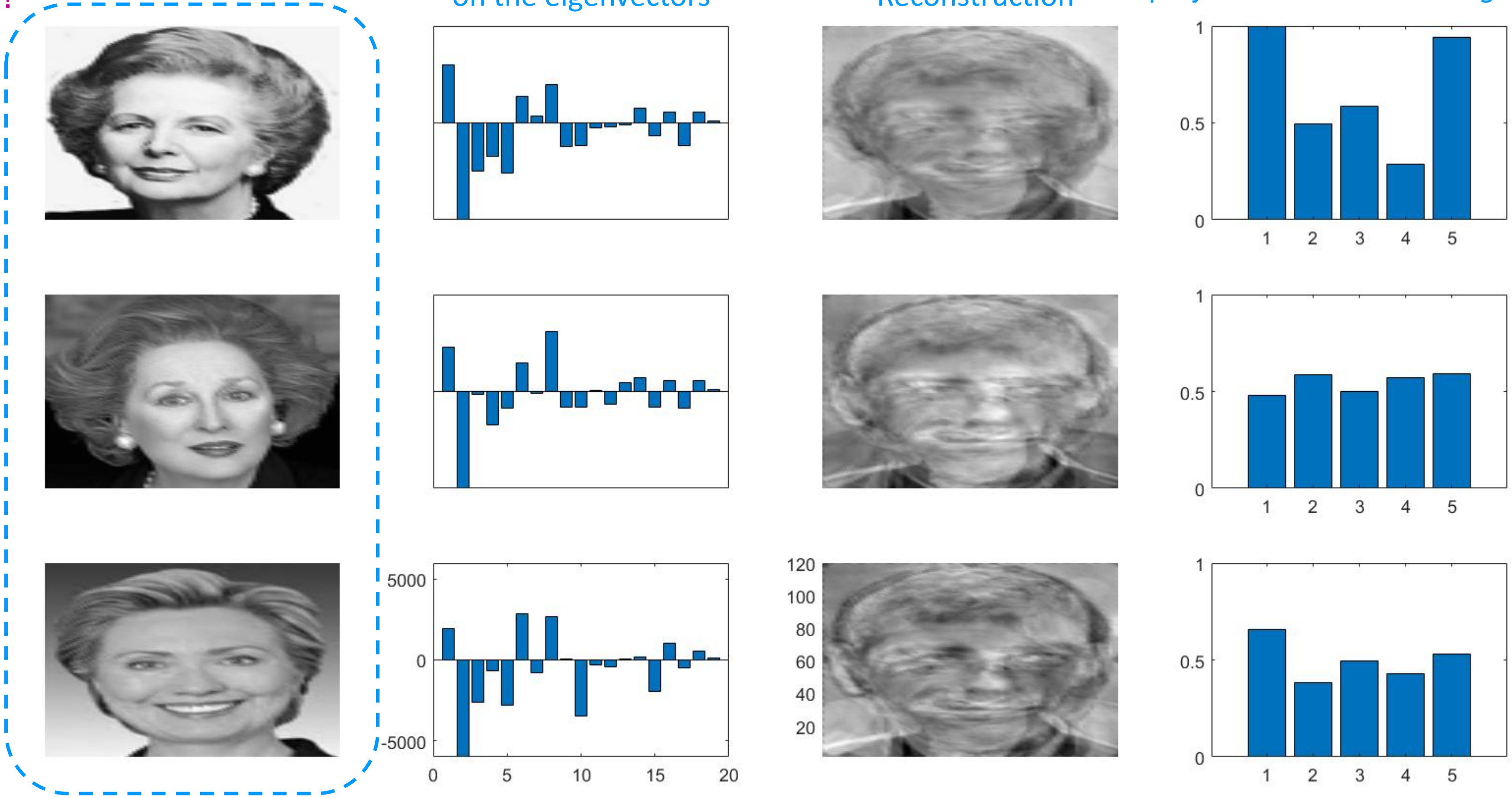
Use your own photo and see which celebrity you look like the most!

Project of each test image on the eigenvectors

Image Reconstruction

The Error is computed between the projection of the test image and the projection of *each training image*

Testing Images



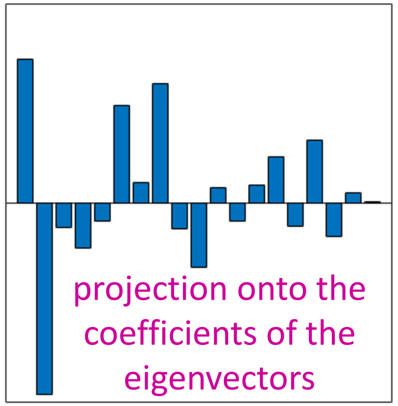
It seems that Meryl Streep did a good job resembling Margret Thatcher!

Tutorial on Tuesday 10th Feb

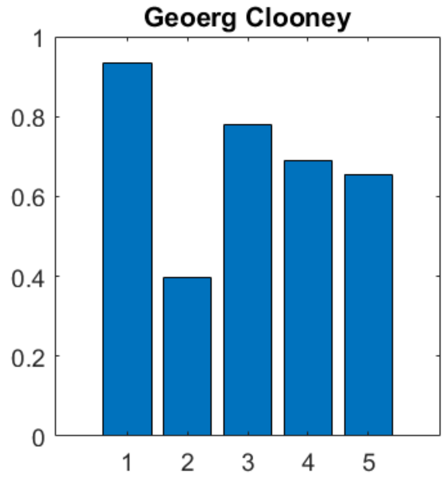
Eigenfaces for Recognition

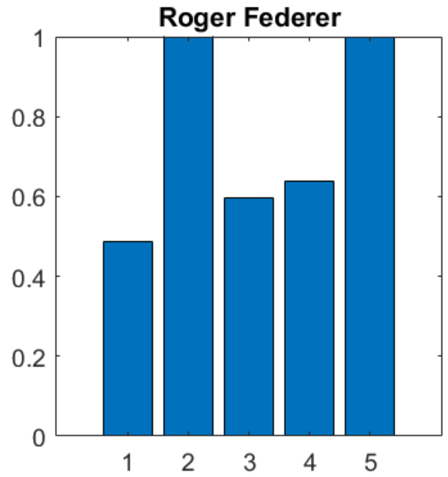
I did it …

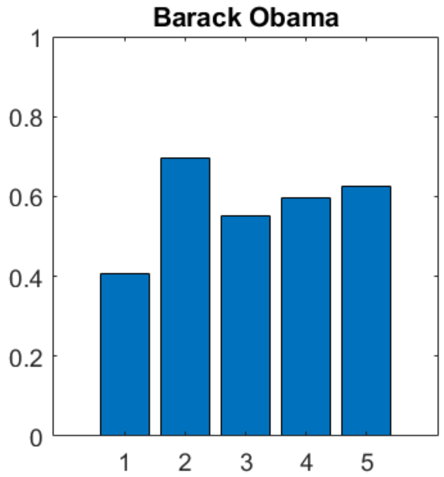
Test Image

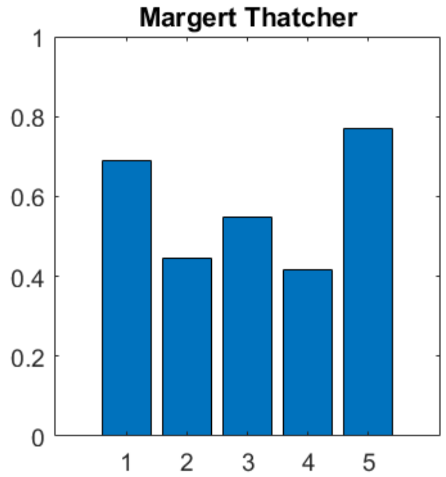


The reconstruction using 20 eigenvectors.









First Coursework (40%)

CW1 Deadline

JANUARY

WK	M	T	W	T	F	S	S
19				1	2	3	4
20	5	6	7	8	9	10	11
21	12	13	14	15	16	17	18
22	19	20	21	22	23	24	25
23	26	27	28	29	30	31	

FEBRUARY

WK	M	T	W	T	F	S	S
23							1
24	2	3	4	5	6	7	8
25	9	10	11	12	13	14	15
26	16	17	18	19	20	21	22
27	23	24	25	26	27	28	

CW1 Release

MARCH

WK	M	T	W	T	F	S	S
27							1
28	2	3	4	5	6	7	8
29	9	10	11	12	13	14	15
30	16	17	18	19	20	21	22
31	23	24	25	26	27	28	29
32	30	31					

APRIL

WK	M	T	W	T	F	S	S
32			1	2	3	4	5
33	6	7	8	9	10	11	12
34	13	14	15	16	17	18	19
35	20	21	22	23	24	25	26
36	27	28	29	30			

You can use either in your CW

MATLAB

jupyter python™

To wrap up with a few reminders…

- **Your third tutorial will be on Tuesday 10th Feb**. Four PGTAs will be with you during the session. Don’t spare any question! Ask them and they will be happy to help.
- All Lecture Material will be uploaded to Moodle later this day, along with the questions of the first tutorial .
- See you next Thursday 12th Feb ***Online*** for our guest lecture!