Image Transformation & Filtering-

Image transformation refers to the manipulation of an image to alter its appearance or characteristics using mathematical functions or operations. This can involve spatial transformations (like rotation, scaling, and translation) or intensity transformations (like contrast adjustment and color modification). Image transformations are fundamental techniques in image processing and computer vision for improving image quality, extracting information, or preparing images for further analysis.

Applications of Image Transformations

- **Geometric corrections** (e.g., correcting lens distortion or aligning images)
- **Feature extraction** (e.g., detecting edges or corners)
- **Image registration** (aligning multiple images for comparison or fusion)
- **Enhancement** (e.g., improving contrast, brightness, or sharpness)
- **Compression** (e.g., reducing image size while preserving important information)

Example: Using Image Transformation to Enhance Contrast

Let's say you have an image where the pixel intensities are clustered in a narrow range, and you want to enhance the contrast to make details more visible. You can apply contrast stretching or histogram equalization to adjust the intensity distribution. These transformations will stretch the pixel values across a broader range, enhancing the visibility of details in both the dark and light regions.

Filtering-

Filtering in image processing refers to the process of modifying an image by using a mathematical operation, usually by applying a kernel or filter mask to the image. Filters can be used for various purposes like smoothing, sharpening, edge detection, noise reduction, and feature enhancement. Filters are often applied in both the spatial domain (directly on the image pixels) and the frequency domain (after transforming the image using Fourier transforms).

Applications of Filtering:

- **Noise Reduction**: Removing unwanted noise (Gaussian, salt-and-pepper noise).
- **Edge Detection**: Detecting object boundaries (Sobel, Canny).
- **Image Sharpening**: Enhancing fine details (unsharp masking).
- **Smoothing**: Reducing image details for tasks like segmentation (Gaussian, median filters).

Intensity transformations -

Intensity transformations are among the simplest of all image processing techniques. Approaches whose results depend only on the intensity at a point are called point processing techniques or Intensity transformation techniques. Although intensity transformation and spatial filtering methods span a broad range of applications, most of the examples in this article are applications to image enhancement..

In this article we will be going over the following intensity transformation functions:

- Image Negatives
- Log Transformations
- Gamma Transformations

For the demonstration the following images would be used:

Image Negatives:
To find the negative of an image we will be using the negative transformation function which has the form:

$S = L - 1 - r$

Where **S** is the output pixel value, **L** is a number of unique intensities and **r** is the intensity of the image in the range [0, L-1].

The reason why we subtracted 1 from L is that L is the number of unique intensities present in a color channel. Therefore, in the case of an 8-bit image, the value of L would be 256. But since the intensities/color values start from 0, we subtract 1 from L to adjust the range.

The image colors got inverted such that all the blacks turned to whites and vice versa. Reversing the intensity levels of a digital image in this manner produces the equivalent of a photographic negative. This process gives us the complement of the image.

A log transformation maps a narrow range of low-intensity values in the input into a wider range of output levels. The general form of log transformation is

$S = C * \log(1 + r)$

Where **S** is the output pixel value, **C** is a constant and **r** is the intensity of the image in the range [0, L-1].

Example 1:

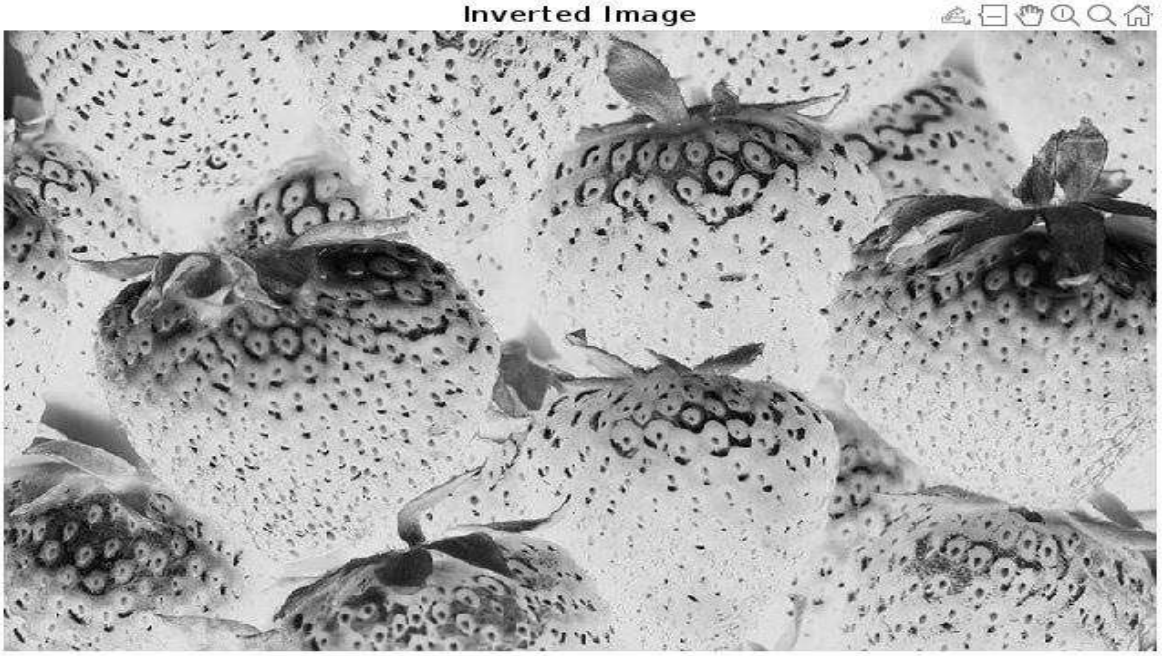
Code for Intensity Transformation

```
  img = imread('Sampleimage.jpg');

% In case of 8 Bit image

  L = 256;

% The below operation is equivalent to 255 - img

  s = (L - 1) - img;

  figure,imshow(s); title('Inverted Image')
```

Output:

Inverted Image

Inverted Image



Log Transformation:
A log transformation maps a narrow range of low-intensity values in the input into a wider range of output levels. The general form of log transformation is

*S = C * log(1 + r)*

Where **S** is the output pixel value, **C** is a constant and **r** is the intensity of the image in the range [0, L-1].

Example 2:

Code for Log Transformation

```
img = imread('Sampleimage.jpg');
% Convert datatype to Double
% (for allowing fractional values)
r = double(img);
% Constant determining the
% nature of the log curve
C = 1;
% Performing log transformation
S = C * log(1 + r);
T = 255/(C * log(256));
% Converting the datatype back
% to integer for displaying
B = uint8(T * S);
figure,imshow(B); title('Log Transformation');
```

Example 2:

```
Code for Log Transformation

img = imread('Sampleimage.jpg');

% Convert datatype to Double

% (for allowing fractional values)

r = double(img);

% Constant determining the

% nature of the log curve

C = 1;

% Performing log transformation

S = C * log(1 + r);

T = 255/(C * log(256));

% Converting the datatype back

% to integer for displaying

B = uint8(T * S);

figure,imshow(B); title('Log Transformation');
```

Output:

The resultant image is one that appears way more bright than the original. This method could be used to brighten images that are dominated by dark regions. The limitation of Log transformations is that they are very limited in usage. Such that the Log transformations can accomplish this compressing of intensity levels in an image, but the power-law transformations discussed later are much more versatile for this purpose.

Power Law Transformation:
Power law transformations have the form

$S = cr^{\gamma}$

Where **S** is the output pixel value, and **c** and **γ** are positive constants. The above equation could also be written as

$s = c(r + \varepsilon)^{\gamma}$

This type of transformation is generally used for gamma correction in Displays or in Images. This is done because our eyes perceive images in a gamma-shaped curve, whereas cameras capture images in a linear fashion.

Example 3:

```
Code for Power Law Transformation

img = imread('Sampleimage.jpg');

% Convert datatype to Double

% (for allowing fractional values)

r = double(img);

% The below value represents gamma

G = 0.6;

% Applying the Power Law Transformation

S = C * (r .^G);

T = 255/(C * (255 .^G));

% Converting the datatype back

% to integer for displaying

O = uint8(T * S);

figure,imshow(O); title('Power Law Transformation');
```

Output:

The resultant images have a bit higher gamma than the original. The power law transformation is very useful, such that it can compress/spread values in higher as well as lower intensity levels. This makes it an ideal choice if wanting to gamma correct an image from variable sources.

Spatial Filtering -

Spatial Filtering technique is used directly on pixels of an image. Mask is usually considered to be added in size so that it has specific center pixel. This mask is moved on the image such that the center of the mask traverses all image pixels.

Classification on the basis of Linearity
There are two types:
1. Linear Spatial Filter
2. Non-linear Spatial Filter

General Classification:

Smoothing Spatial Filter

Smoothing filter is used for blurring and noise reduction in the image. Blurring is pre-processing steps for removal of small details and Noise Reduction is accomplished by blurring.

Types of Smoothing Spatial Filter

1. Linear Filter (Mean Filter)

2. Order Statistics (Non-linear) filter

1. **Mean Filter:** Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. The idea is replacing the value of every pixel in an image by the average of the grey levels in the neighborhood define by the filter mask. Below are the types of mean filter:
    - **Averaging filter:** It is used in reduction of the detail in image. All coefficients are equal.
    - **Weighted averaging filter:** In this, pixels are multiplied by different coefficients. Center pixel is multiplied by a higher value than average filter.
2. **Order Statistics Filter:** It is based on the ordering the pixels contained in the image area encompassed by the filter. It replaces the value of the center pixel with the value determined by the ranking result. Edges are better preserved in this filtering. Below are the types of order statistics filter:
    - **Minimum filter:** 0th percentile filter is the minimum filter. The value of the center is replaced by the smallest value in the window.
    - **Maximum filter:** 100th percentile filter is the maximum filter. The value of the center is replaced by the largest value in the window.
    - **Median filter:** Each pixel in the image is considered. First neighboring pixels are sorted and original values of the pixel is replaced by the median of the list.

Sharpening Spatial Filter

It is also known as derivative filter. The purpose of the sharpening spatial filter is just the opposite of the smoothing spatial filter. Its main focus in on the removal of blurring and highlight the edges. It is based on the first and second order derivative.

First Order Derivative:

- Must be zero in flat segments.
- Must be non zero at the onset of a grey level step.
- Must be non zero along ramps.

First order derivative in 1-D is given by:

$$f' = f(x+1) - f(x)$$

Second Order Derivative:

- Must be zero in flat areas.
- Must be non zero at the onset and end of a ramp.
- Must be zero along ramps.

Second order derivative in 1-D is given by:

$$f'' = f(x+1) + f(x-1) - 2f(x)$$

Histogram :

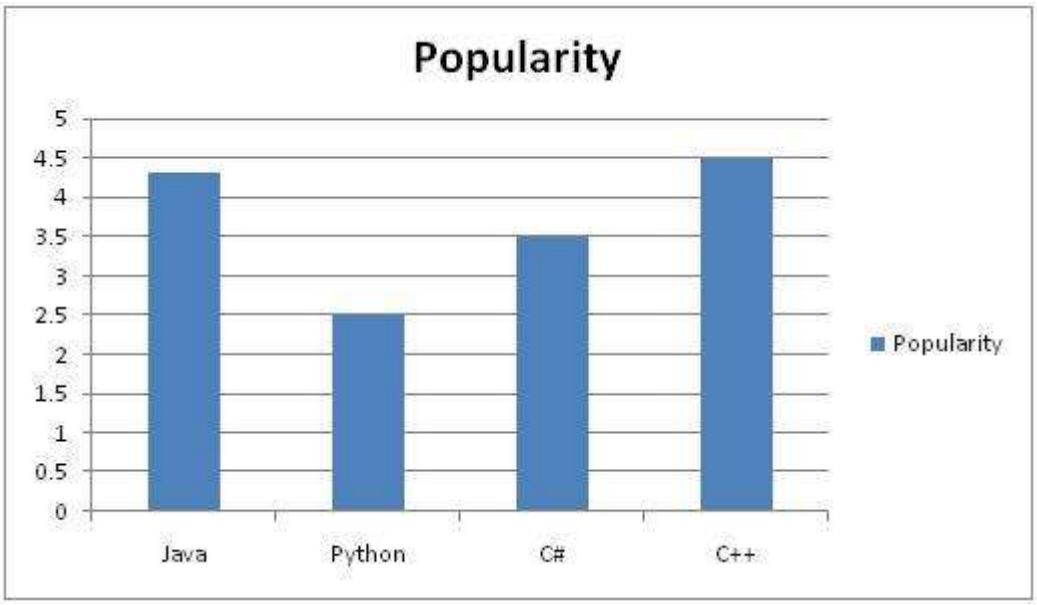
A histogram is a graph. A graph that shows frequency of anything. Usually histogram have bars that represent frequency of occurring of data in the whole data set.

A Histogram has two axis the x axis and the y axis.

The x axis contains event whose frequency you have to count.

The y axis contains frequency.

The different heights of bar shows different frequency of occurrence of data.



Now we will see an example of this histogram is build

Example

Consider a class of programming students.

At the end of the semester, you got this result that is shown in table. But it is very messy and does not show your overall result of class. So you have to make a histogram of your result, showing the overall frequency of occurrence of grades in your class. Here how you are going to do it.

Result sheet

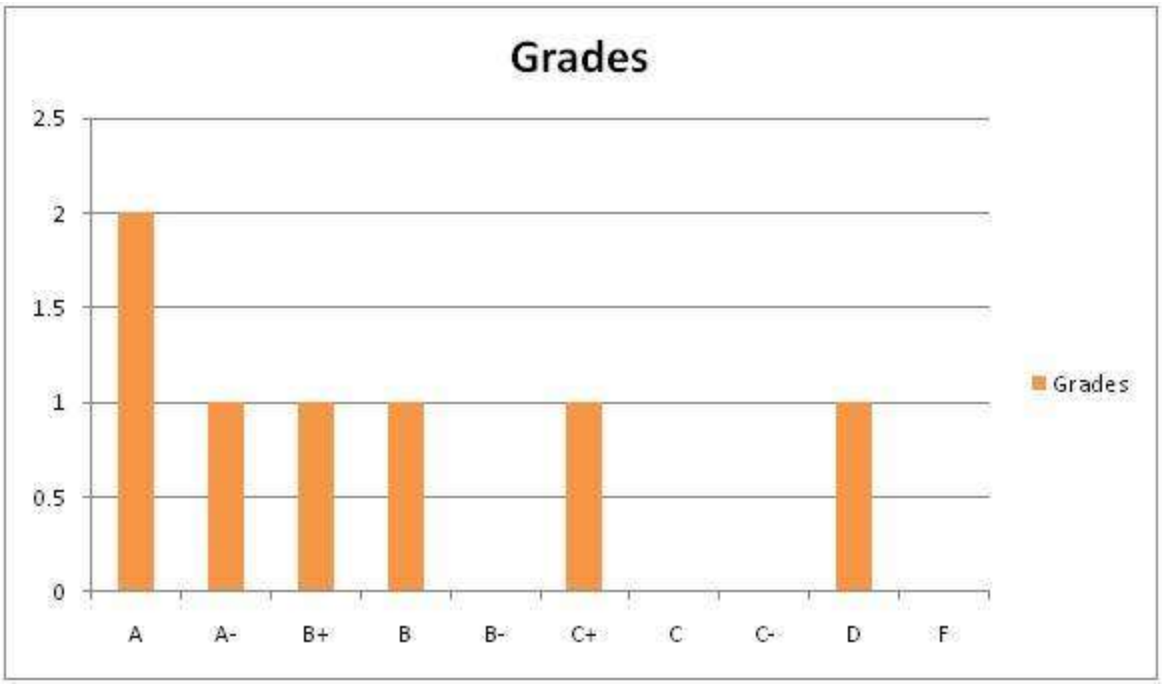
Name	Grade
John	A

Jack	D
Carter	B
Tommy	A
Lisa	C+
Derek	A-
Tom	B+

Histogram of result sheet

Now what you are going to do is, that you have to find what comes on the x and the y axis.

There is one thing to be sure, that y axis contains the frequency, so what comes on the x axis. X axis contains the event whose frequency has to be calculated. In this case x axis contains grades.

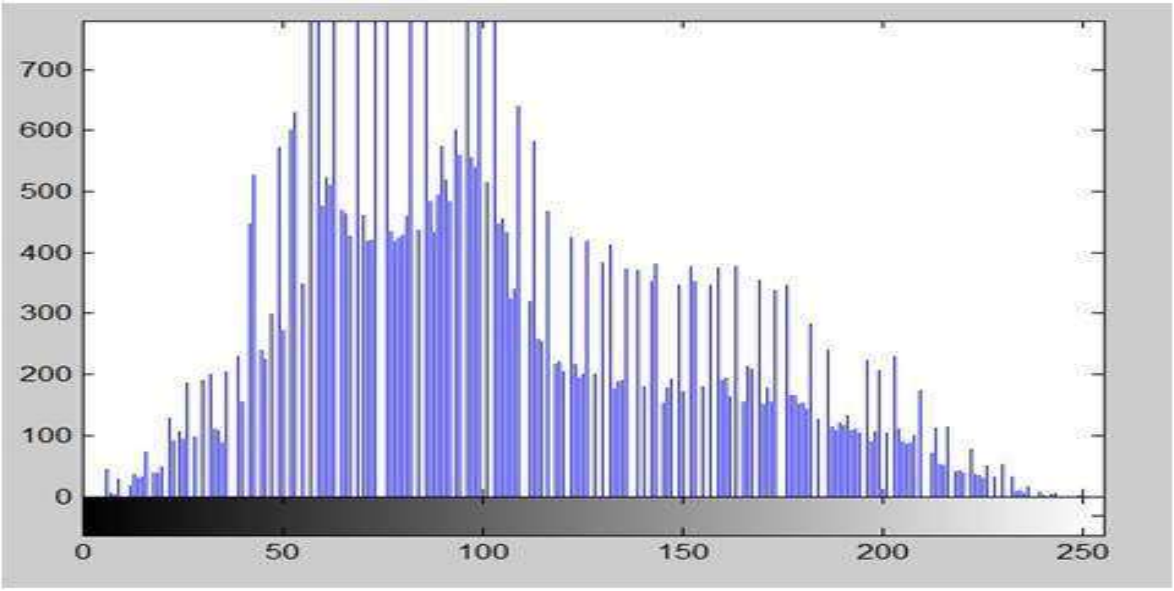


Histogram of an Image

Histogram of an image, like other histograms also shows frequency. But an image histogram, shows frequency of pixels intensity values. In an image histogram, the x axis shows the gray level intensities and the y axis shows the frequency of these intensities.

Example

The histogram of the above picture of the Einstein would be something like this



The x axis of the histogram shows the range of pixel values. Since its an 8 bpp image, that means it has 256 levels of gray or shades of gray in it. Thats why the range of x axis starts from 0 and end at 255 with a gap of 50. Whereas on the y axis, is the count of these intensities.

As you can see from the graph, that most of the bars that have high frequency lies in the first half portion which is the darker portion. That means that the image we have got is darker. And this can be proved from the image too.

Histogram Processing Techniques

Histogram Sliding

In Histogram sliding, the complete histogram is shifted towards rightwards or leftwards. When a histogram is shifted towards the right or left, clear changes are seen in the brightness of the image. The brightness of the image is defined by the intensity of light which is emitted by a particular light source.

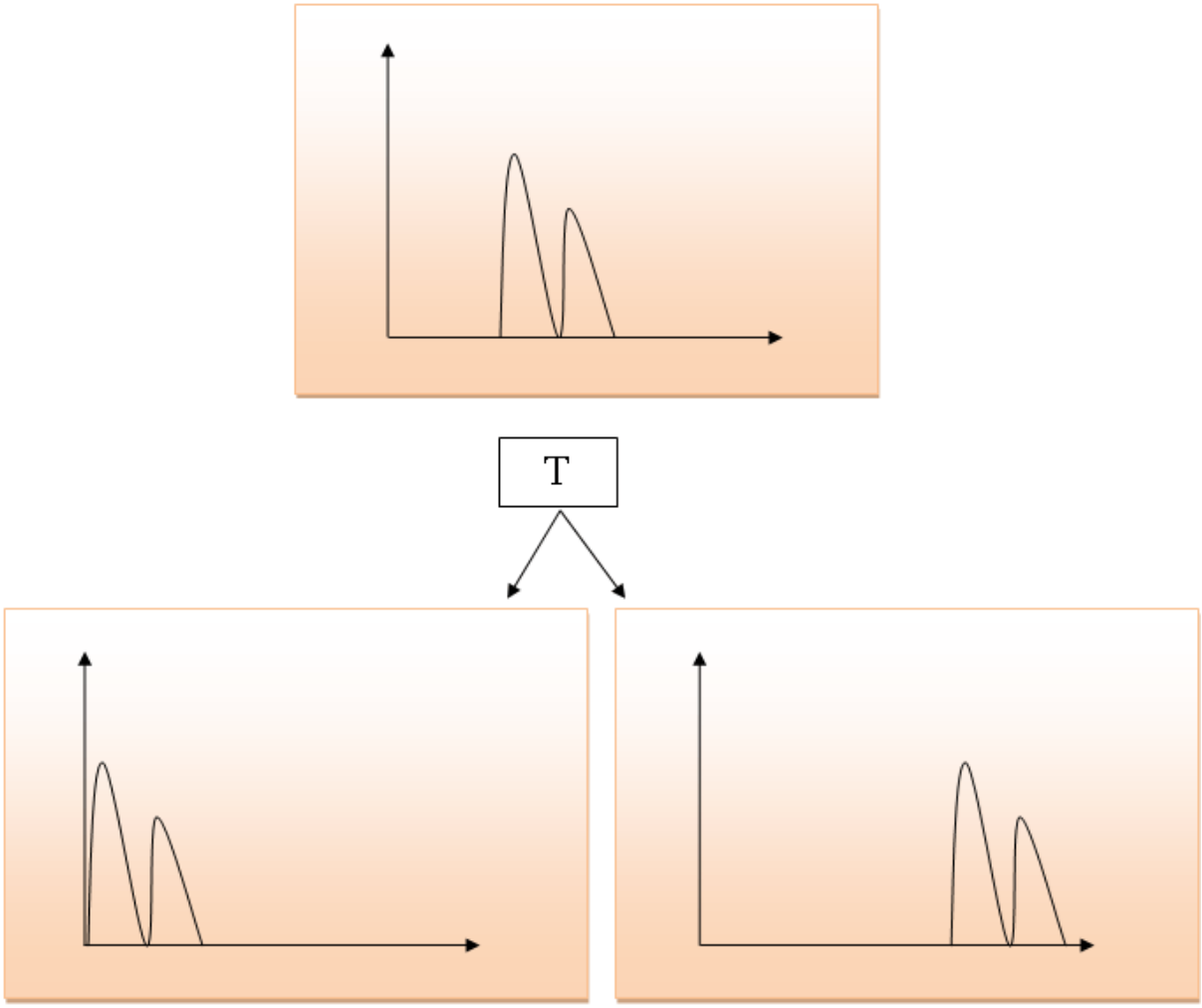


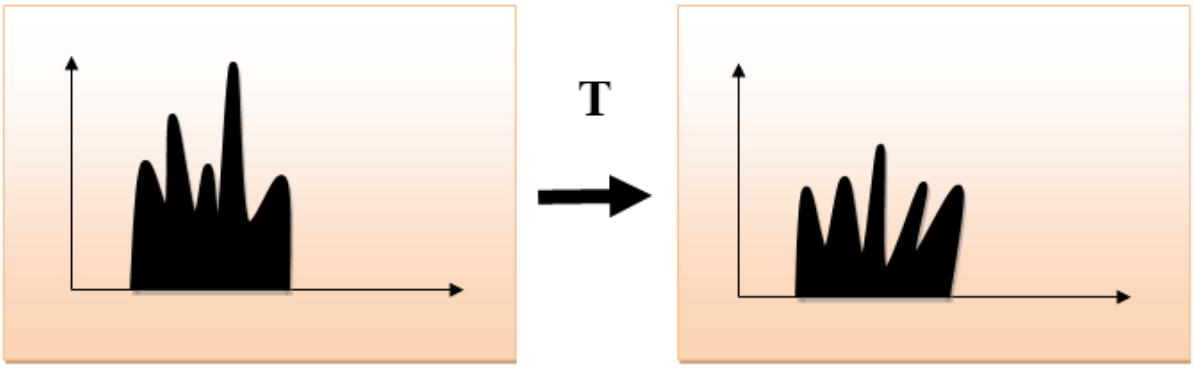
Fig. Histogram Sliding

Histogram Stretching

In histogram stretching, contrast of an image is increased. The contrast of an image is defined between the maximum and minimum value of pixel intensity.

If we want to increase the contrast of an image, histogram of that image will be fully stretched and covered the dynamic range of the histogram.

From histogram of an image, we can check that the image has low or high contrast.

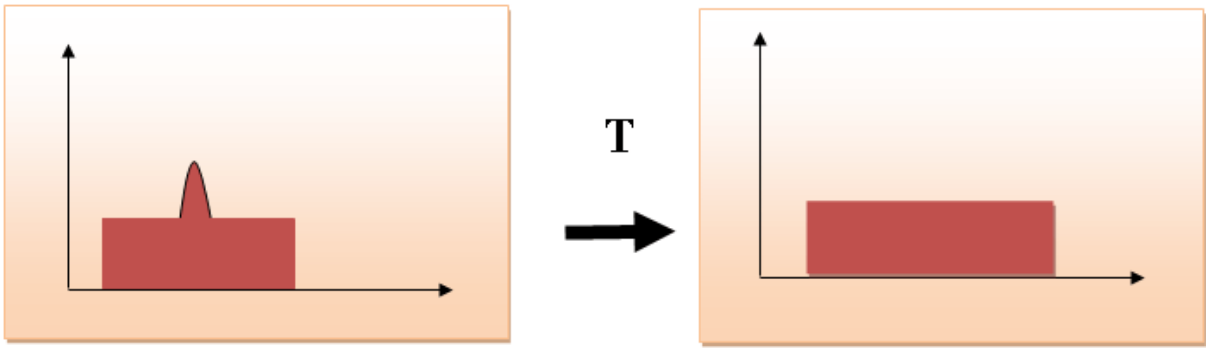


Histogram Equalization

Histogram equalization is used for equalizing all the pixel values of an image. Transformation is done in such a way that uniform flattened histogram is produced.

Histogram equalization increases the dynamic range of pixel values and makes an equal count of pixels at each level which produces a flat histogram with high contrast image.

While stretching histogram, the shape of histogram remains the same whereas in Histogram equalization, the shape of histogram changes and it generates only one image.



Fourier transform-

Fourier transform is mainly used for image processing. In the Fourier transform, the intensity of the image is transformed into frequency variation and then to the frequency domain. It is used for slow varying intensity images such as the background of a passport size photo can be represented as low-frequency components and the edges can be represented as high-frequency components. Low-frequency components can be removed using filters of FT domain. When an image is filtered in the FT domain, it contains only the edges of the image. And if we do inverse FT domain to spatial domain then also an image contains only edges. Fourier transform is the simplest technique in which edges of the image can be fined.

Fourier transform is a mathematical model that decomposes a function or signal into its constituent frequencies. It helps to transform the signals between two different domains like transforming the frequency domain to the time domain**.** It is a powerful tool used in many fields, such as signal processing, physics, and engineering, to analyze the frequency content of signals or functions that vary over time or space.

The generalized form of the complex Fourier series is referred to as the Fourier transform. Fourier transforms are used to represent the mathematical functions and frequency domain. It helps to expand the non-periodic functions and convert them into easy sinusoid functions.

Properties of Fourier Transform

- Fourier transform is a linear transform. It is called linear transform.
- Modulation property is the property in which the function is modulated by other function.
- A shift in the time domain corresponds to a phase shift in the frequency domain in Fourier Transform
- Multiplying a time-domain signal by a complex exponential corresponds to a shift in the frequency domain in Fourier Transform

- In Fourier Transform taking the complex conjugate of the time-domain signal corresponds to taking the complex conjugate of the frequency-domain signal and reversing the frequency.

There are two types of Fourier transform i.e., forward Fourier transform and inverse Fourier transform.

Continuous Fourier Transform (CFT)
For a continuous-time function f(t), the Fourier transform F(ω) is defined as:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{i\omega t}\, dt$$

where:

- **F(ω)** is the Fourier transform of f(t)
- **ω** is the Angular Frequency
- **i** is the Imaginary Number (**i2 = -1**)
- **t** is Time

Fourier Transform Formula
The formula for the Fourier transforms of a function f(x) is given by:

$$f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi i k x}\, dk$$

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x}\, dx$$

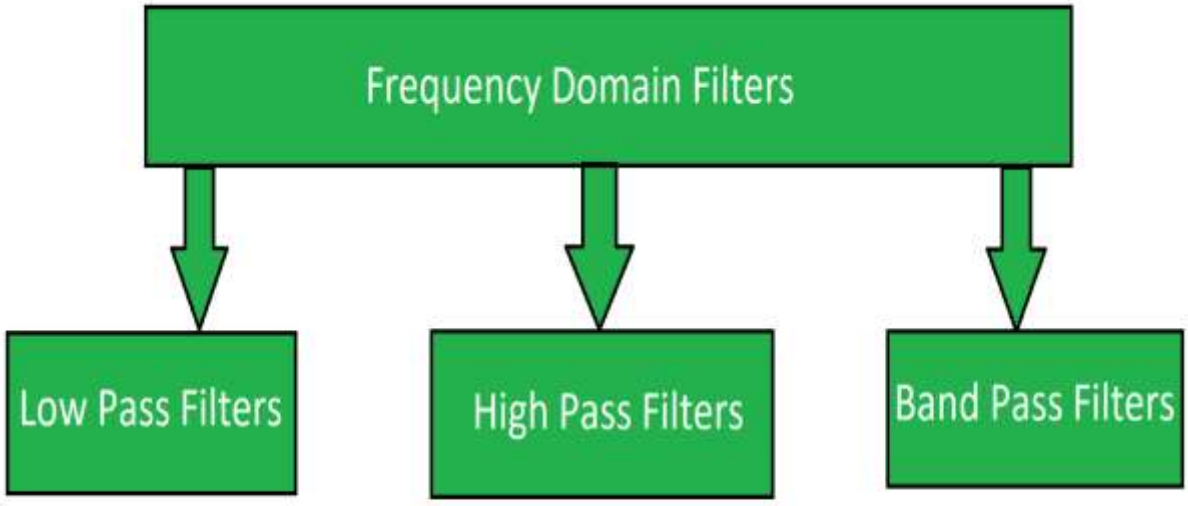
Forward Fourier Transform
The **forward Fourier transform** is a mathematical technique used to transform a **time-domain signal into its frequency-domain representation**. This transformation is fundamental in various fields, including signal processing, image processing, and communications. Forward Fourier Transform is represented by F(k). The symbol for forward Fourier transform is f^*f*^(k) and is defined as:

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x}\, dx$$

Inverse Fourier Transform
The inverse Fourier transform is the process of **converting a frequency-domain representation of a signal back into its time-domain form**. This is the reverse process of the forward Fourier transform. Inverse Fourier Transform is represented by f(x). Symbol for Inverse Fourier transform is f˘*f*(x) and is defined as:

$$f(x) = Fk{-}1Fk{-}1[F(k)]\,(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi i k x}\, dk$$

Frequency Domain Filters -
Frequency Domain Filters are used for smoothing and sharpening of image by removal of high or low frequency components. Sometimes it is possible of removal of very high and very low frequency. Frequency domain filters are different from spatial domain filters as it basically focuses on the frequency of the images. It is basically done for two basic operation i.e., Smoothing and Sharpening.
These are of 3 types:



Classification of Frequency Domain Filters

1. Low pass filter:
Low pass filter removes the high frequency components that means it keeps low frequency components. It is used for smoothing the image. It is used to smoothen the image by attenuating high frequency components and preserving low frequency components. Mechanism of low pass filtering in frequency domain is given by:

$G(u, v) = H(u, v) . F(u, v)$

where $F(u, v)$ is the Fourier Transform of original image

and $H(u, v)$ is the Fourier Transform of filtering mask

2. High pass filter:
High pass filter removes the low frequency components that means it keeps high frequency components. It is used for sharpening the image. It is used to sharpen the image by attenuating low frequency components and preserving high frequency components. Mechanism of high pass filtering in frequency domain is given by:

$H(u, v) = 1 - H'(u, v)$

where $H(u, v)$ is the Fourier Transform of high pass filtering

and $H'(u, v)$ is the Fourier Transform of low pass filtering

3. Band pass filter:
Band pass filter removes the very low frequency and very high frequency components that means it keeps the moderate range band of frequencies. Band pass filtering is used to enhance edges while reducing the noise at the same time.

Colour models -

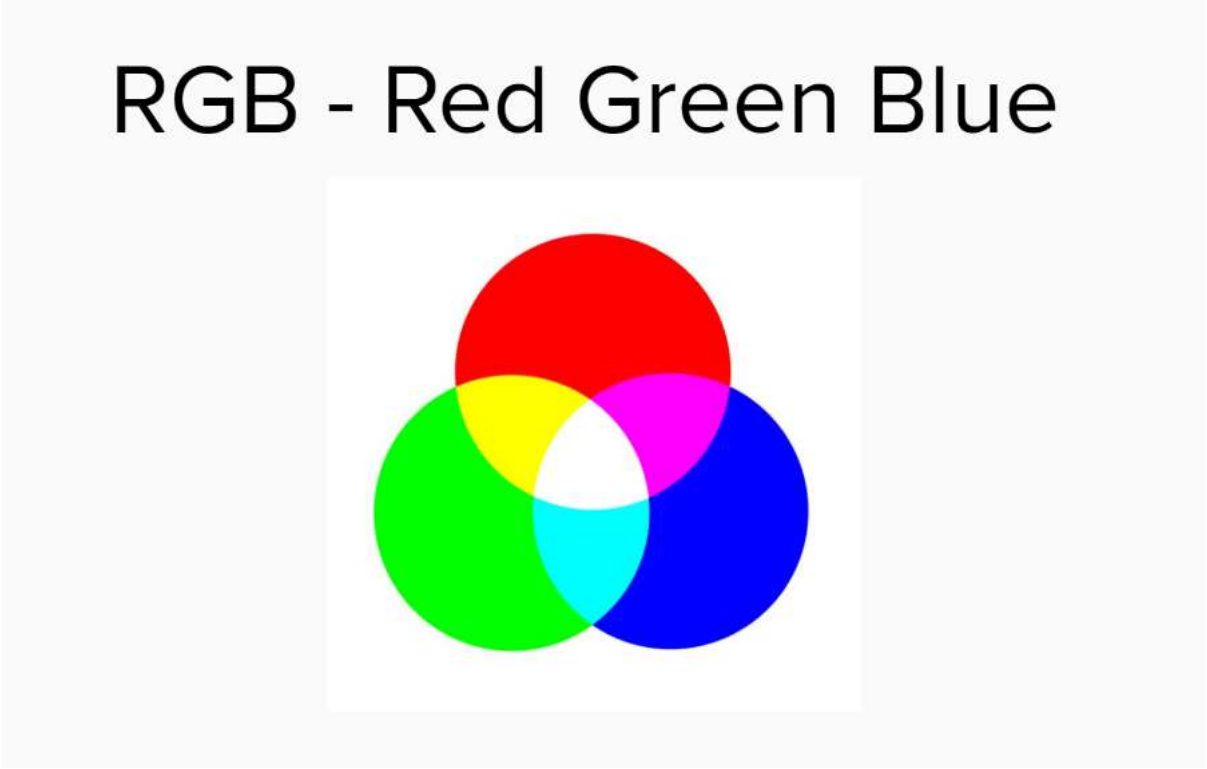
The colour spaces in image processing aim to facilitate the specifications of colours in some standard way.

Different types of colour models are used in multiple fields like in hardware, in multiple applications of creating animation, etc.

RGB: The RGB colour model is the most common colour model used in Digital image processing and openCV. The colour image consists of 3 channels. One channel each for one colour. Red, Green and Blue are the main colour components of this model. All other colours are produced by the proportional ratio of these three colours only. 0 represents the black and as the value increases the colour intensity increases.

Properties:

- This is an additive colour model. The colours are added to the black.
- 3 main channels: Red, Green and Blue.
- Used in DIP, openCV and online logos.



Colour combination:
Green(255) + Red(255) = Yellow
Green(255) + Blue(255) = Cyan
Red(255) + Blue(255) = Magenta
Red(255) + Green(255) + Blue(255) = White

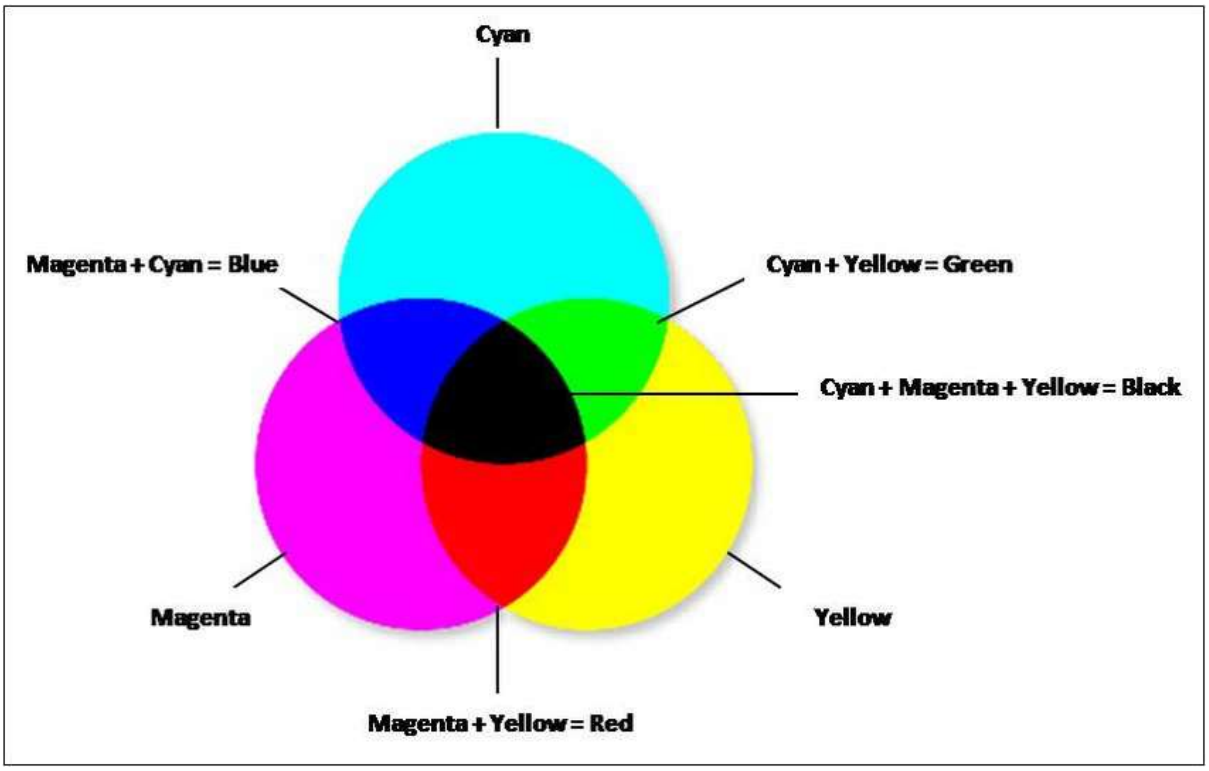
CMYK: CMYK colour model is widely used in printers. It stands for Cyan, Magenta, Yellow and Black (key). It is a subtractive colour model. 0 represents the primary colour and 1 represents the lightest colour. In this model, point (1, 1, 1) represents black, and (0,0,0) represents white. It is a subtractive model thus the value is subtracted from 1 to vary from least intense to a most intense colour value.

1-RGB = CMY

Cyan is negative of Red.

Magenta is negative of Green.

Yellow is negative of Blue.



HSV: The image consists of three channels. Hue, Saturation and Value are three channels. This colour model does not use primary colours directly. It uses colour in the way humans perceive them. HSV colour when is represented by a cone.
Hue is a colour component. Since the cone represents the HSV model, the hue represents different colours in different angle ranges.

Red colour falls between 0 and 60 degrees in the HSV cone.

Yellow colour falls between 61 and 120 degrees in the HSV cone.

Green colour falls between 121 and 180 degrees in the HSV cone.

Cyan colour falls between 181 and 240 degrees in the HSV cone.

Blue colour falls between 241 and 300 degrees in the HSV cone.

Magenta colour falls between 301 and 360 degrees in the HSV cone.

Saturation as the name suggest describes the percentage of the colour. Sometimes this value lies in the 0 to 1 range. 0 being the grey and 1 being the primary colour. Saturation describes the grey colour.

The value represents the intensity of the colour chosen. Its value lies in percentage from 0 to 100. 0 is black and 100 is the brightest and reveals the colour.

HSV model is used in histogram equalization and converting grayscale images to RGB colour images.

YIQ: YIQ is the most widely colour model used in Television broadcasting. Y stands for luminance part and IQ stands for chrominance part. In the black and white television, only the luminance part (Y) was broadcast. The y value is similar to the grayscale part. The colour information is represented by the IQ part.

Pseudo Coloring-

Pseudo Coloring is one of the attractive categories in image processing. It is used to make old black and white images or videos colorful. Pseudo Coloring techniques are used for analysis identifying color surfaces of the sample image and adaptive modeling of histogram black and white image. Selecting different values in the layers R, G, B is the most important achievement of this technique such that this method is based on the analysis of histogram characteristics in the sample image, to assign different values in different layers of a color image, we take action. Pseudo-coloring is also known as a coloring topic in processing digital images.

In this technique, a gray level of type integer unsigned 8-bit (a number between zero and 255) should be considered as input and three outputs must be achieved for three layers of the color digital image graph and each of these three levels should be of type integer unsigned 8-bit (a number between zero and 255). There are many techniques for this conversion that have some differences based on our needs.

- **Grayscale image:** It is a black and white image. The pixels values are shades of gray colour which is the combination of white and black shades. The image is represented in form of one 2-Dimensional matrix. Each value represents the intensity or brightness of the corresponding pixel at that coordinate in the image. Total 256 shades are possible for the grayscale images. 0 means black and 255 means white. As we increase the value from 0 to 255, the white component gets increases and brightness increases.
- **RGB color image:** It is a colored image. It consists of three 2-Dimensional matrices, which are called channels. Red, Green and Blue channels contain the corresponding colour values for each pixel in the image. In integer format, the

range of pixel intensity goes from 0 to 255. 0 means black and 255 represents the highest intensity of the primary colour. There exist 256 shades of each colour.

Steps:

- Read the grayscale image.
- If its bit-depth is 24, then make it 8.
- Create an empty image of the same size.
- Assign some random weight to RGB channels.
- Copy weighted product of grayscale image to each channel of Red, Green, and Blue.
- Display the images after creation.

Functions Used:

- imread( ) inbuilt function is used to read the image.
- imtool( ) inbuilt function is used to display the image.
- rgb2gray( ) inbuilt function is used to convert RGB to gray image.
- uint8( ) inbuilt function is used to convert double into integer format.
- pause( ) inbuilt function is used to stop execution for specified seconds.

Example:

```
% MATLAB code for pseudo colouring

% of grayscale images.

% UTILITY CODE

k=imread("gfglogo.png");

gray2rgb(k);

imtool(grayscale,[]);

function gray2rgb(img)

% Convert into grayscale if not.

[x,y,z]=size(img);

if(z==3)

   grayscale=rgb2gray(img);
```

```
end

gray=double(grayscale./255);

rgb(:,:,1)=gray(:,:)*0.5;

rgb(:,:,2)=gray(:,:)*0.6;

rgb(:,:,3)=gray(:,:)*0.4;

imtool(rgb,[]);

c(x,y,z)=0;

colour=uint8(c);

colour(:,:,1)=grayscale(:,:)*0.5;

colour(:,:,2)=grayscale(:,:)*0.7;

colour(:,:,3)=grayscale(:,:)*0.4;

imtool(colour,[]);

pause(10);

imtool close all;

end
```

Wavelet transforms-

Wavelet transforms are mathematical tools for analyzing data where features vary over different scales. For signals, features can be frequencies varying over time, transients, or slowly varying trends. For images, features include edges and textures. Wavelet transforms were primarily created to address limitations of the Fourier transform.

A wavelet is a function that oscillates between positive and negative values and has **finite duration**. It is used to represent data in both the time domain and frequency domain.

The **wavelet transform** analyzes a signal by decomposing it into **wavelet coefficients** at different scales and shifts. These coefficients represent both the time and frequency components of the signal.

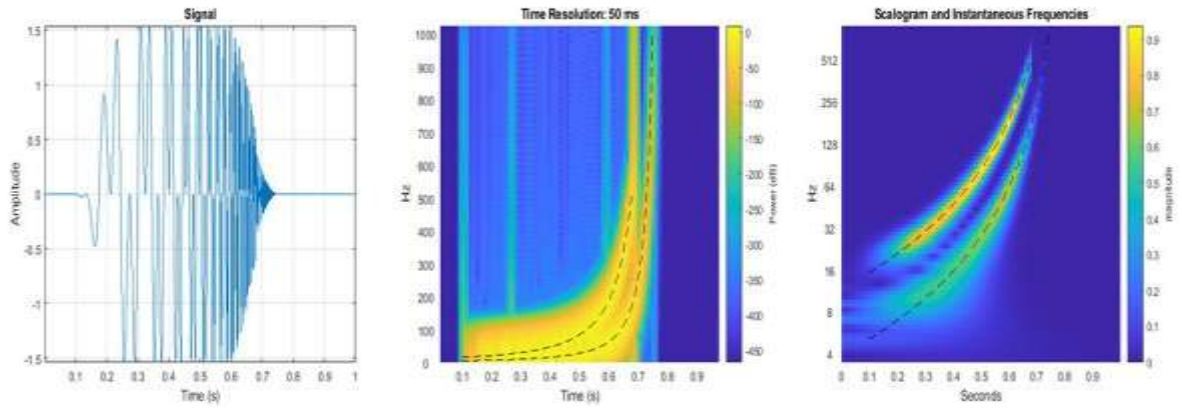
While Fourier analysis consists of decomposing a signal into sine waves of specific frequencies, wavelet analysis is based on decomposing signals into shifted and scaled versions of a *wavelet*. A wavelet, unlike a sine wave, is a rapidly decaying, wave-like oscillation. This enables wavelets to represent data across multiple scales.

Wavelet transforms can be classified into two broad classes: the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT).

The CWT analyzes the signal using **continuous** scales and shifts. It is often used for **time-frequency analysis** of non-stationary signals.

The CWT is similar to the **Short-Time Fourier Transform (STFT)** but uses variable window sizes: wide for low frequencies and narrow for high frequencies.

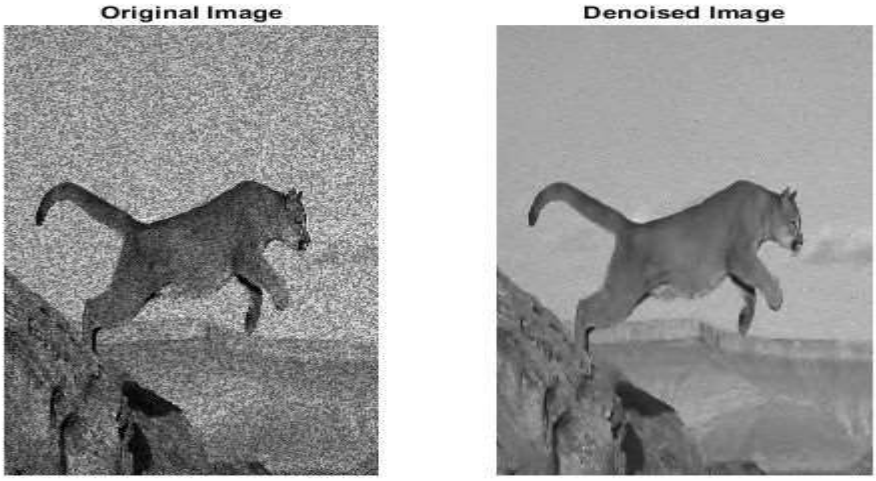
The continuous wavelet transform is a time-frequency transform, which is ideal for analysis of non-stationary signals. A signal being nonstationary means that its frequency-domain representation changes over time. CWT is similar to the short-time Fourier transform (STFT). The STFT uses a fixed window to create a local frequency analysis, while CWT tiles the time-frequency plane with variable-sized windows. The window widens in time, making it suitable for low-frequency phenomena, and narrows for high-frequency phenomena. The continuous wavelet transform can be used to analyze transient behavior, rapidly changing frequencies, and slowly varying behavior.



Analyzing a hyperbolic chirp signal (left) with two components that vary over time in MATLAB. The short-time Fourier transform (center) does not clearly distinguish the instantaneous frequencies, but the continuous wavelet transform (right) accurately captures them.

The DWT is a **discrete** version of the CWT, where the scales and translations are **sampled** rather than continuous. It is computationally more efficient and is widely used in applications like **compression** and **denoising**. DWT decomposes the signal into different frequency bands using **filter banks**. It works by repeatedly applying low-pass and high-pass filters to the signal, followed by downsampling.

With the discrete wavelet transform scales are discretized more coarsely than with CWT. This makes DWT useful for compressing and denoising signals and images while preserving important features. You can use discrete wavelet transforms to perform multiresolution analysis and split signals into physically meaningful and interpretable components.



Applications of Wavelet Transforms

- **Signal Processing**:
  Wavelets are used for analyzing **non-stationary signals** (signals whose frequency characteristics change over time). They are particularly useful for signals like speech, music, and biomedical signals.

- **Image Processing**:
  - **Compression**: Wavelet-based techniques like JPEG2000 are used for image compression, where high-frequency details are discarded, and low-frequency information is preserved.
  - **Denoising**: Removing noise from signals or images while preserving edges and important features.

- **Time-Frequency Analysis**:
  Wavelets are ideal for analyzing signals with **time-varying frequencies**, as they allow for precise localization in both time and frequency.

- **Feature Extraction**:
  In machine learning and data analysis, wavelets can be used to extract important features from signals and data for further analysis.