Digital Image Applications

Lecture Notes

Dr. Taha Basheer Taha

Applied Science Faculty, Information Technology Department

Computer Vision and Digital Image Applications

Computer Vision is a field that enables computers to see and interpret images. Unlike humans, computers need specialized algorithms to understand image content.

Some Applications of Computer Vision:

Automotive – Self-driving cars and traffic monitoring.

Manufacturing – Quality control and defect detection.

Healthcare – Medical imaging and disease diagnosis.

HR & Hiring – AI-powered interviews and candidate assessments.

Oil & Gas – Image classification for geological analysis.

Advanced real life computer vision examples:

A. Video Analysis & Object Recognition

Finding a particular scene in a movie requires manual fast-forwarding which leads security teams struggle to identify suspects in hours of surveillance footage.

Solution:

IBM's AI-powered video tagging:

- Detects objects in video frames.
- Tags scenes with keywords for quick searching.
- Example: Finding a "blue van" in hours of footage instantly.

B. Structural Inspections Using Computer Vision

Problem:

Electric towers & bridges need frequent maintenance to check for rust and defects.

Manual inspection is dangerous and time-consuming.

Drones near high-voltage wires can be unsafe.

Solution:

High-resolution images of structures can be captured from the ground.

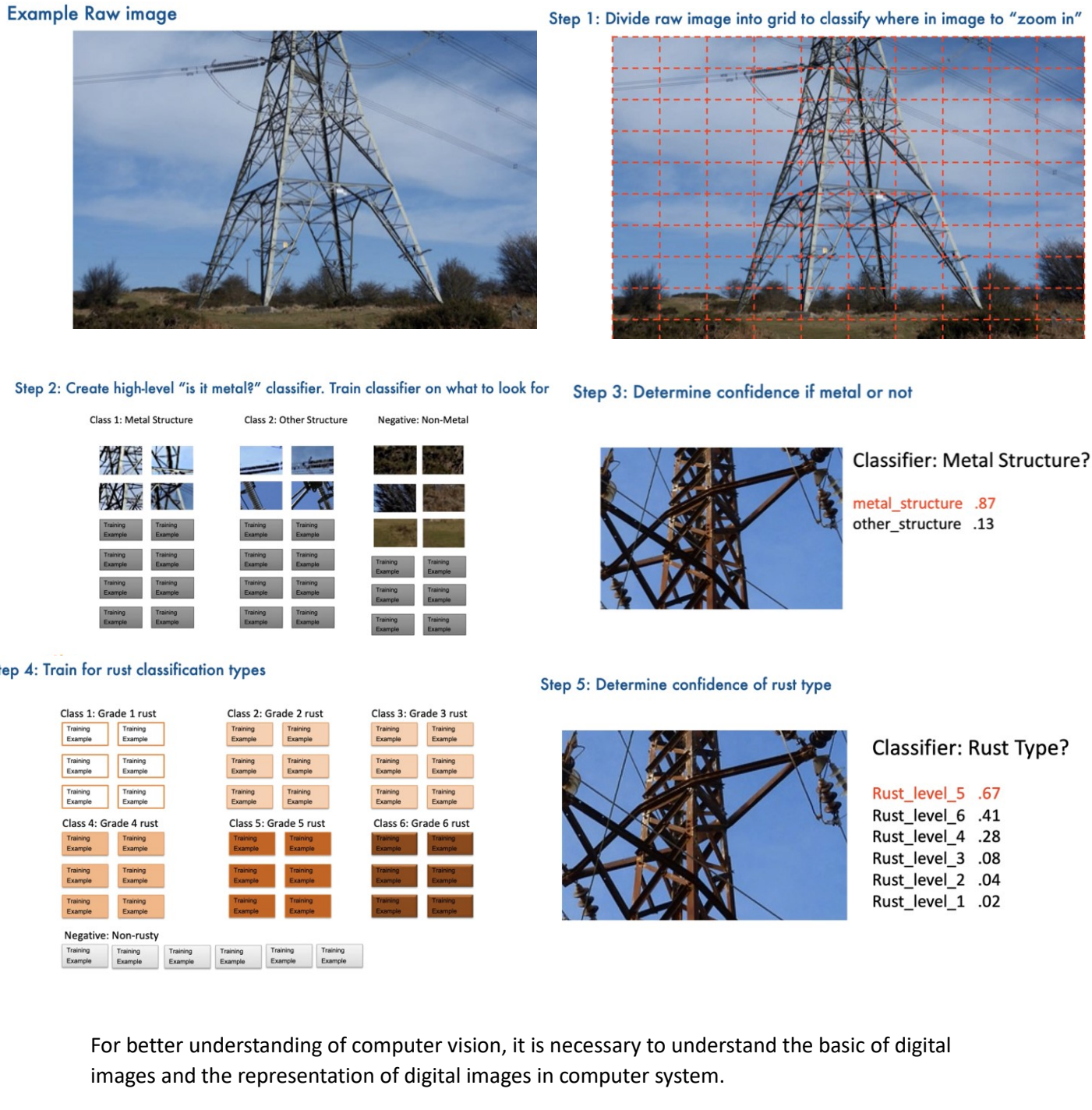
Computer vision steps:

Metal Detection: Classify metal vs. non-metal structures.

Rust Detection: Train a classifier to grade rust levels from 1 (mild) to 6 (severe).

Heatmaps: Overlay rust severity on images to highlight damaged areas

Classification:



For better understanding of computer vision, it is necessary to understand the basic of digital images and the representation of digital images in computer system.

Digital Images

A digital image is a numerical representation of a visual object, stored as a grid of pixels.

- Grayscale images have one intensity value per pixel (0-255).
- Color images use three color channels: Red (R), Green (G), and Blue (B) (RGB Model).
- Images are stored as pixel grids with width × height resolution.

Each pixel in a digital image has an intensity value (0 = black, 255 = white).

Color images store separate values for Red, Green, and Blue (RGB channels).

Higher values indicate brighter pixels; lower values indicate darker pixels.

Example : Imagine a 5×5 grayscale image:

100 120 140 160 180

110 130 150 170 190

120 140 160 180 200

130 150 170 190 210

140 160 180 200 220

The higher the number, the brighter the pixel.

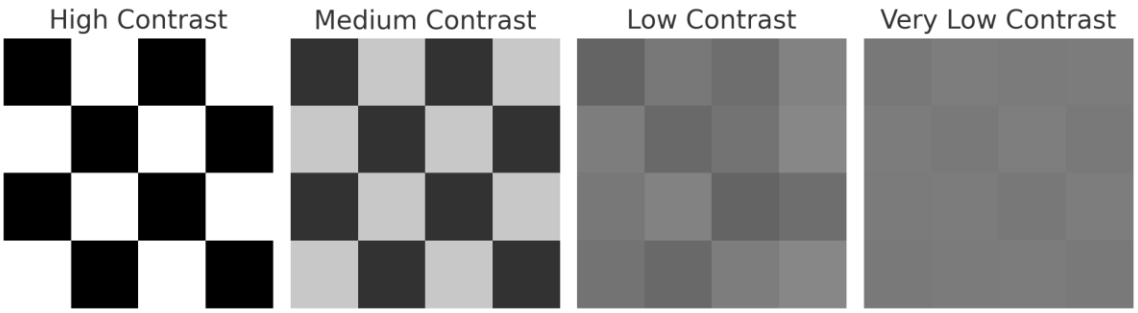
The lower the number, the darker the pixel.

Contrast and Quantization

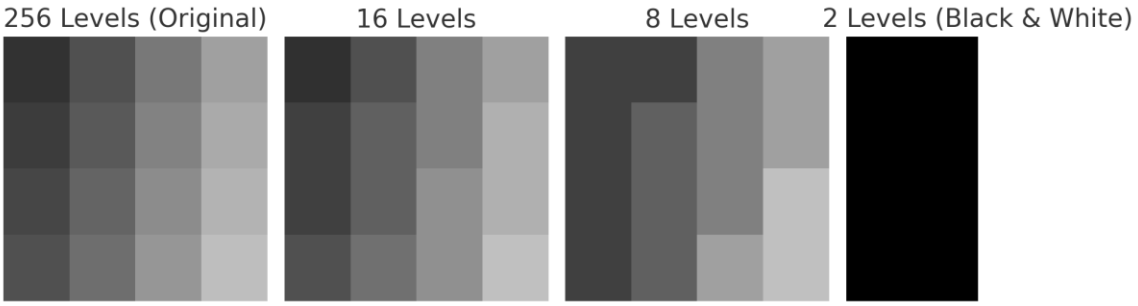
Contrast is the difference between light and dark areas.

High contrast = Clearer edges and better visibility.

Low contrast = Faded, dull image.



Quantization is the process of mapping a large set of values (such as pixel intensities) to a smaller set of values.



OpenCV (Open Source Computer Vision Library) is one of the most widely used libraries for image processing. By using OpenCV, we can manipulate, enhance, and analyze images efficiently. It works with NumPy arrays for fast processing.

OpenCV stores images in BGR format due to historical reasons and hardware compatibility.

Many older image processing systems used BGR for performance efficiency.

To convert BGR to RGB in OpenCV:

image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

Image Rotating:

Each location in pair of values x,y for rows and columns, and it has a value, in binary image either 0 or 1.

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

Draw the letter T
by highlighting the cell.

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

Coordinates		Value
x	y	

For 90^{o} clockwise rotation the equation is :New X = y and New Y = 2-x

Coordinates		Value	New X NewX=y	New Y NewY=2-x	New x,y
x	y				

i.e.: Move the pixel in coordinates (x, y) which is value is (n) to the new coordinates (New X, New Y)

E.g.: Move the pixel in coordinates (0,0) which value is 1, to the new coordinates (0,2)

Homework: Rotate the image anti-clock wise

Spatial Filtering

Spatial filtering is a technique used in image processing where we modify the intensity of a pixel based on its neighboring pixels. It is commonly used for image smoothing, sharpening, and edge detection.

How Does It Work?

A small matrix called a filter/kernel slides over the image with a process called convolution.

Each pixel's new value is calculated as a weighted sum of its surrounding pixels.

Convolution: Is multiplying a small matrix (the kernel) with a region of the image and summing the values is called Convolution. Convolution is the mathematical operation behind spatial filtering.

Convolution Key-steps:

- Add padding and place the kernel on the image.
- Multiply each kernel value by the corresponding pixel value.
- Sum all the multiplied values to get the new pixel.
- Move the kernel across the image, repeating the process.

Average Filter (low-pass filter)

Purpose: Reduces noise and smooths images by averaging the pixel values in a local neighborhood.

Each pixel is replaced with the average of its surrounding pixels.

Effect: Blurs the image, removes small variations, and reduces noise.

Mathematical Representation

The operation can be represented using a convolution kernel.

The 3×3 average filter kernel:

$$H = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

For a general N × N filter, the kernel is:

$$H = \frac{1}{N^2}\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

Effects of Average Filtering

- Removes high-frequency details.
- Reduces noise.
- Causes blurring, which can lead to loss of important details.

Example: Apply average filtering for the image sample (Use zero padding) :

1	2	3
2	3	4
3	4	5

Sharpening Filter (high pass filter)

Purpose: Enhances edges and details in an image.

Concept: Highlights areas where pixel intensity changes sharply (high-frequency regions).

The operation can be represented using a convolution kernel.

The 3×3 common sharpening filter kernel:

$$H = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Effect: Makes images appear more detailed by increasing contrast along edges.

Classwork: Apply the sharpening filter to the following image sample using the zero padding.

1	2	3
2	3	4
3	4	5

Median Filter Kernel

Unlike Mean/Average Filters, the Median Filter does not have a fixed kernel with weights. Instead, it works by sorting pixel values.

Example: 3×3 Median Filter Application

Consider a 3×3 neighborhood:

100 200 255

150 10 180

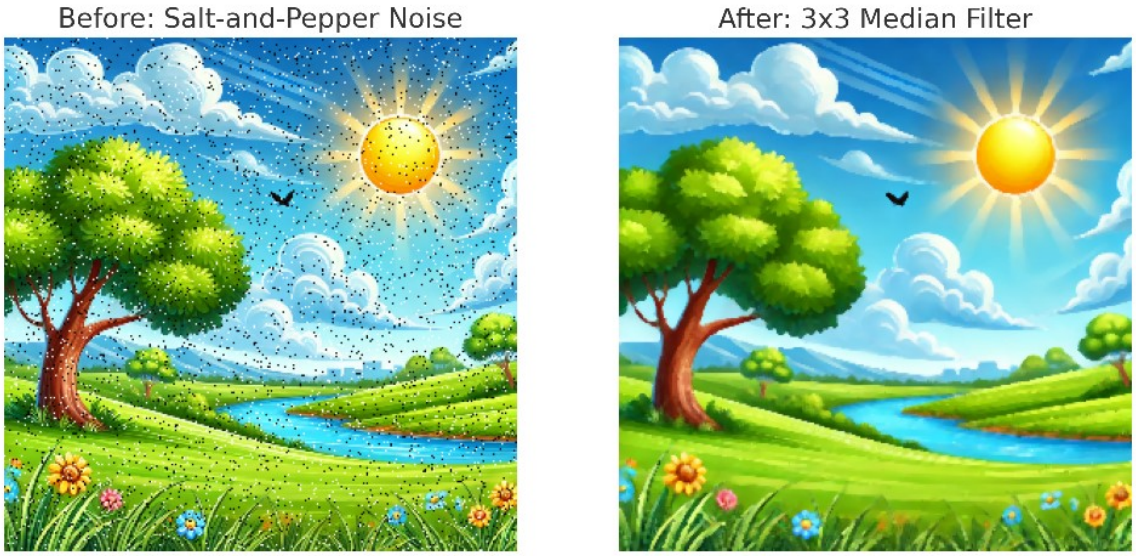
30 250 0

Sorted pixel values in the kernel:

[0, 10, 30, 100, 150, 180, 200, 250, 255]

The median value is 150, so the center pixel (10) is replaced with 150.

Figure below shows the original image, image before salt and pepper, and image after median filter.



How Does Median Filtering (and other filters) Handle Edge Pixels?

When applying a 3×3 filter, the filter needs a neighborhood around each pixel. But what happens at the image edges where there are not enough neighbors?

Possible Edge-Padding Methods:

Zero Padding (Default in Some Cases)

Adds 0s around the image to maintain the size.

Issue: Artificial black border effects.

Not recommended for median filtering.

Mirror Padding (Reflects Edge Pixels)

The pixels at the edge are mirrored outward.

This helps maintain image continuity.

Better for preserving edges.

Replication Padding (Repeats Edge Pixels)

Extends the edge pixels outward instead of using zeros.

Works well to preserve the structure at the edges.

Effect of Different Padding Techniques on Median Filtering

This document explains how different padding techniques affect median filtering using a 3×3 image patch. It provides an analysis of Zero Padding, Mirror Padding, and Replication Padding with numerical examples.

Original 3×3 Pixel Matrix

Consider the following 3×3 pixel neighborhood:

1 2 3
4 5 6
7 8 9

1. Zero Padding (Padding with 0s)

Zero padding adds a border of zeros around the image. The 5×5 zero-padded image is:

0 0 0 0 0
0 1 2 3 0
0 4 5 6 0
0 7 8 9 0
0 0 0 0 0

For the center pixel (5), the sorted values in the 3×3 window are [1, 2, 3, 4, 5, 6, 7, 8, 9]. The median remains 5.

For a border pixel (1), the sorted values are [0, 0, 0, 0, 1, 2, 4, 5], and the median is 0, which introduces artificial black borders.

2. Mirror Padding (Reflecting the Edges)

Mirror padding reflects the border pixels outward. The 5×5 mirror-padded image is:

5 4 5 6 5
2 1 2 3 2
5 4 5 6 5
8 7 8 9 8
5 4 5 6 5

For a border pixel (1), the sorted values in the 3×3 window are [1, 2, 2, 4, 4, 5, 5, 5, 5]. The median is 4, preserving structure.

Mirror padding prevents artificial dark edges and maintains edge continuity.

3. Replication Padding (Repeating Edge Pixels)

Replication padding extends the edge pixels outward. The 5×5 replication-padded image is:

1 1 2 3 3
1 1 2 3 3
4 4 5 6 6
7 7 8 9 9
7 7 8 9 9

For a border pixel (1), the sorted values in the 3×3 window are [1, 1, 1, 1, 2, 2, 4, 4, 5]. The median is 2.

Replication padding is good for preserving edges but may introduce a blocky effect in sharp structures.

The following table summarizes the effects of each padding technique on median filtering:

Padding Type	Edge Handling	Effect on Image
Zero Padding	Adds black pixels at edges	Creates artificial dark borders
Mirror Padding	Reflects edge values	Preserves edges smoothly
Replication Padding	Extends the edge pixels	Can create a blocky effect in some cases

Best Choice for Median Filtering?

Mirror Padding is the best option because:

- It prevents artificial borders (unlike zero padding).

- It smooths out noise without disrupting edges (better than replication padding).