
AI-generated picture

CHULA ΣNGINEERING
Innovation toward Sustainability | ACTNOW

Image Processing for Transportation Engineering Applications – Part 1

2101553: Computer Applications in Transportation

Siwarak Unsiwilai, Ph.D.
Department of Civil Engineering
Chulalongkorn University

Outline

- Section I: Introduction to Image Processing and Its Relevance in Transportation
- Section II: What is an image?
- Section III: What is image processing?
- Section IV: Basic Image Processing Techniques (with MATLAB Demonstration)

Learning Objectives

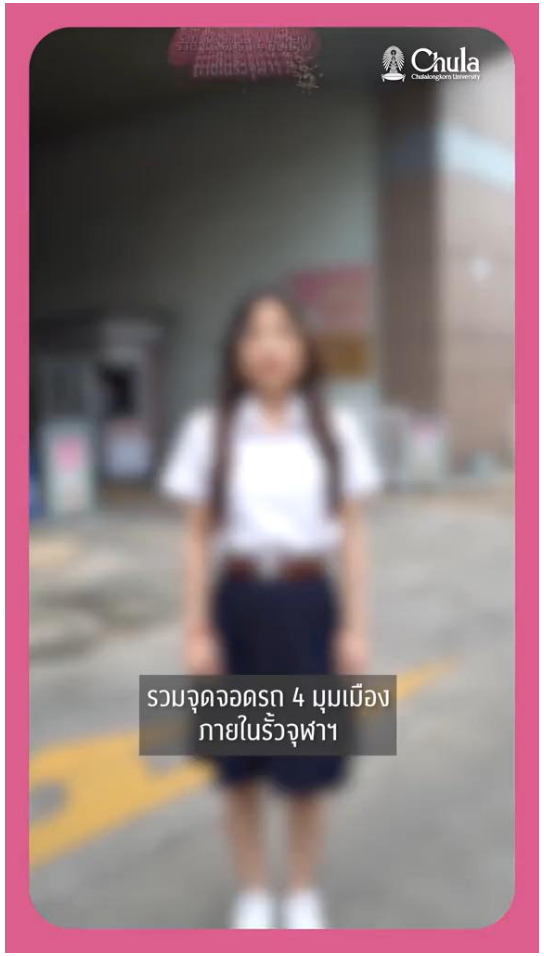
By the end of this class, you will be able to:

- Explain fundamental concepts of image processing
- Recognize real-life and transportation-specific image processing use cases
- Perform basic image processing tasks using MATLAB

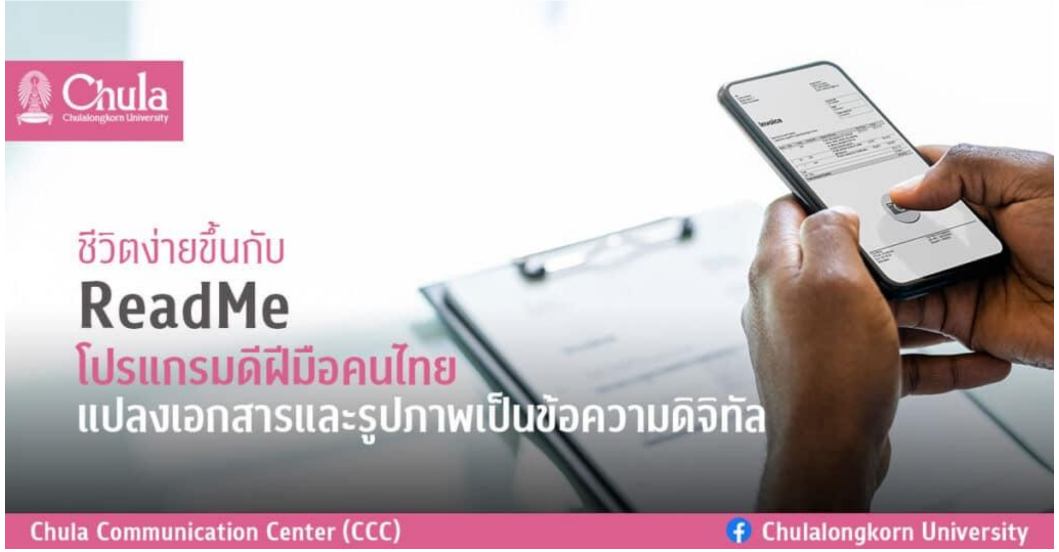
Section I: Introduction

I: How Image Processing Touches Our Everyday Lives

https://blog.google/products/translate/google-translates-instant-camera-translation-gets-upgrade/



https://www.tiktok.com/@chulalongkornuniversity/video/7319155935065410824



https://www.chula.ac.th/highlight/75113/

Optical Character Recognition (OCR)

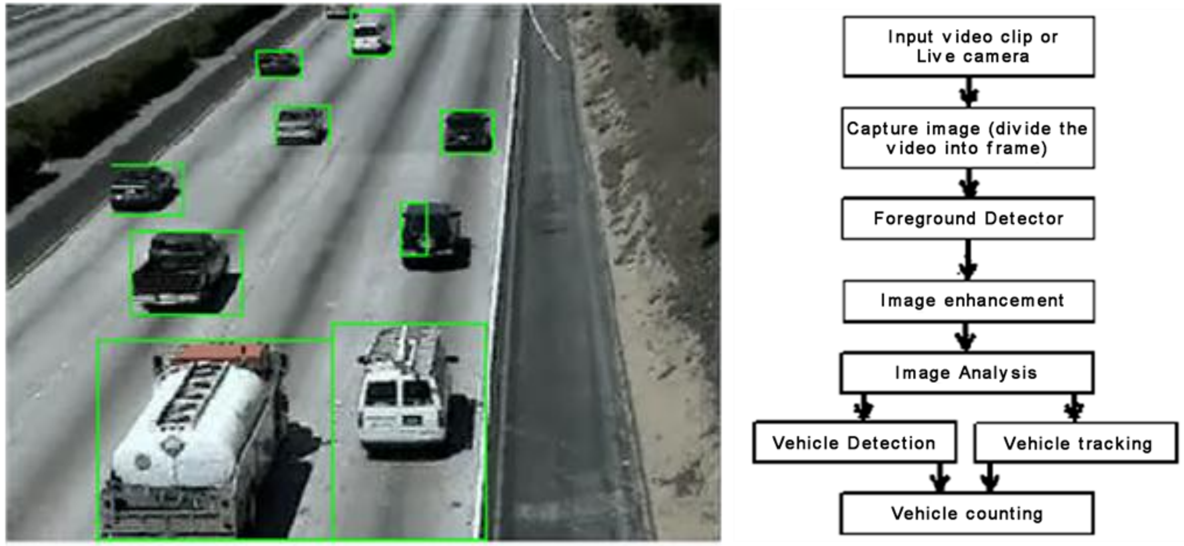
I: How Image Processing Touches Our Everyday Lives

Example use cases:

- Instantly translating signs and menus with your phone’s camera.
- Unlocking smartphones with facial recognition.
- Depositing checks or scanning receipts with banking apps.
- Tagging friends and organizing photos automatically on social media.
- Virtual “try-on” for glasses, makeup, or furniture in shopping apps.

I: How is Image Processing Used in Engineering?

- Traffic Monitoring & Control
  - Camera video feeds are processed to count vehicles, detect congestion, and identify incidents in real time for traffic management and planning.

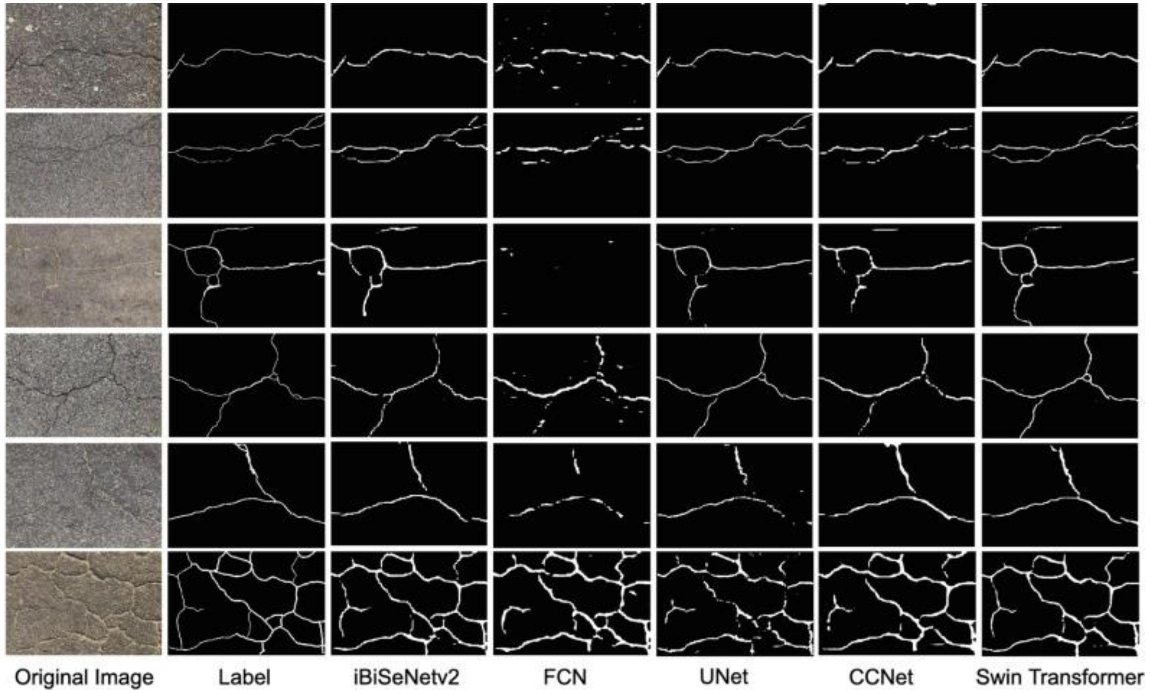


https://www.scirp.org/journal/paperinformation?paperid=78431

https://journal-digitallife.com/publications/article/measurement-of-motor-vehicle-traffic-volume-using-camera-images-and-artificial-intelligence/

I: How is Image Processing Used in Engineering?

- Road & Pavement Inspection
  - UAV/drone or vehicle-mounted cameras automatically detect pavement cracks, potholes, and line markings.

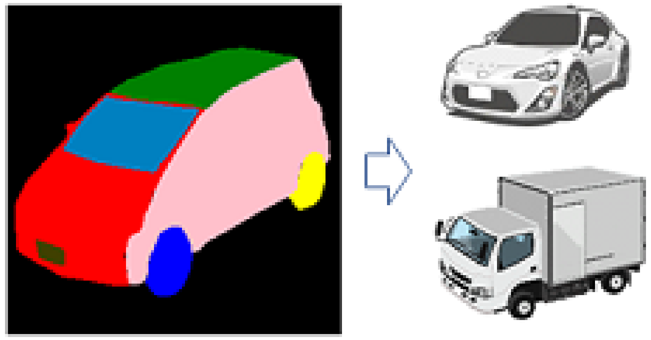


https://www.sciencedirect.com/science/article/pii/S2214391224001569

คู่มือการตรวจสอบและประเมินสภาพทาง ระบบบริหารงานซ่อมบำรุงทางหลวงท้องถิ่น กรมทางหลวงชนบท กระทรวงคมนาคม

I: How is Image Processing Used in Engineering?

- Vehicle Recognition & Classification
  - Roadside or intersection cameras use image processing to classify vehicle types (car, truck, bus, motorcycle) for tolling, surveys, or enforcement.



https://journal-digitallife.com/publications/article/measurement-of-motor-vehicle-traffic-volume-using-camera-images-and-artificial-intelligence/

https://pmc.ncbi.nlm.nih.gov/articles/PMC8914976/

I: How is Image Processing Used in Engineering?

- Asset Management
  - Street and infrastructure visuals are integrated with maps to track condition, plan upgrades, and communicate with stakeholders.

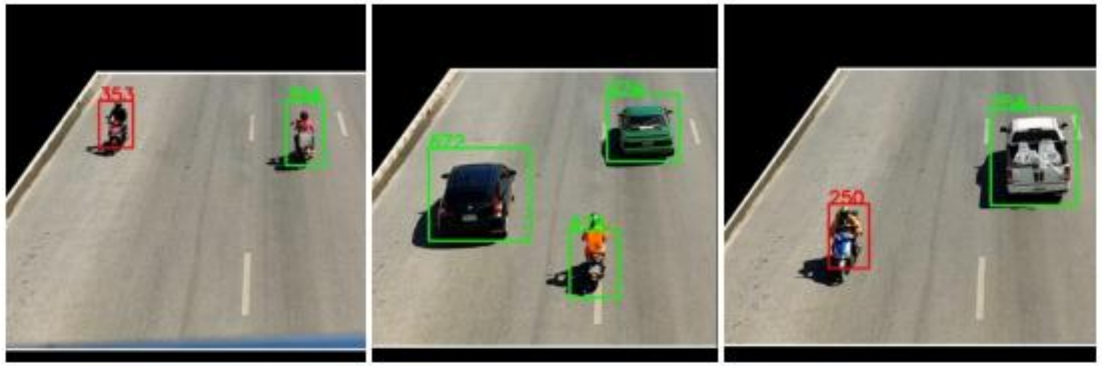
https://www.scielo.br/j/bcg/a/vL8ggk5BQ95bftBhYdjJF6J/?format=html&lang=en

I: How is Image Processing Used in Engineering?

- Safety, Security, and Hazard Detection
  - Surveillance or smart sensors detect hazards (e.g., stopped vehicles, accidents) to send automatic alerts.

(a) (b)

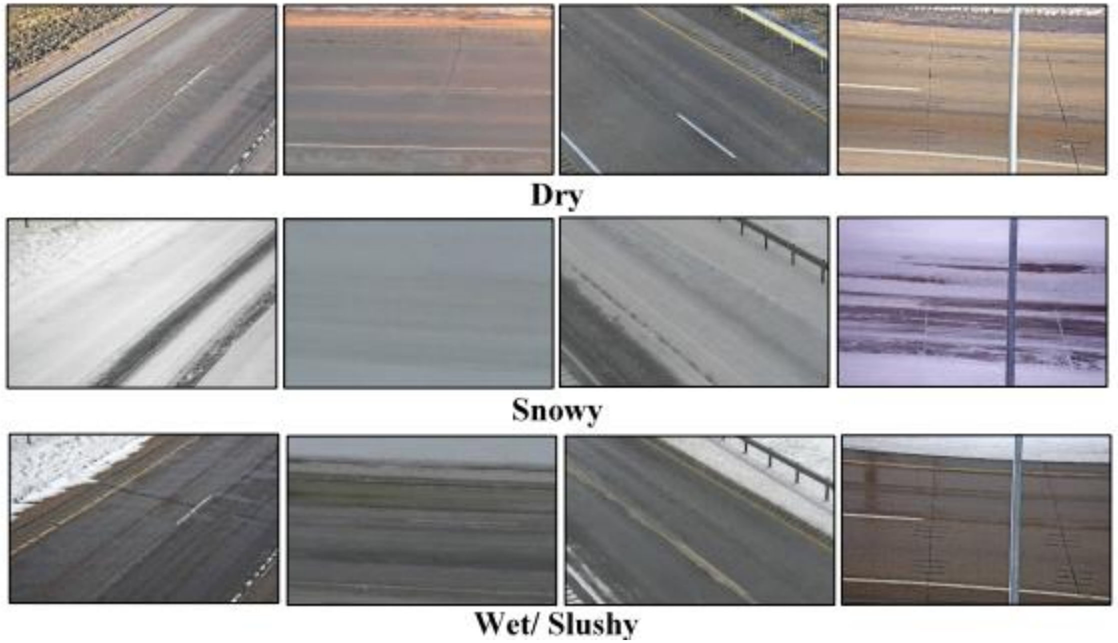
https://its.isr.uc.pt/publications/RECPAD08-GMonteiro.pdf



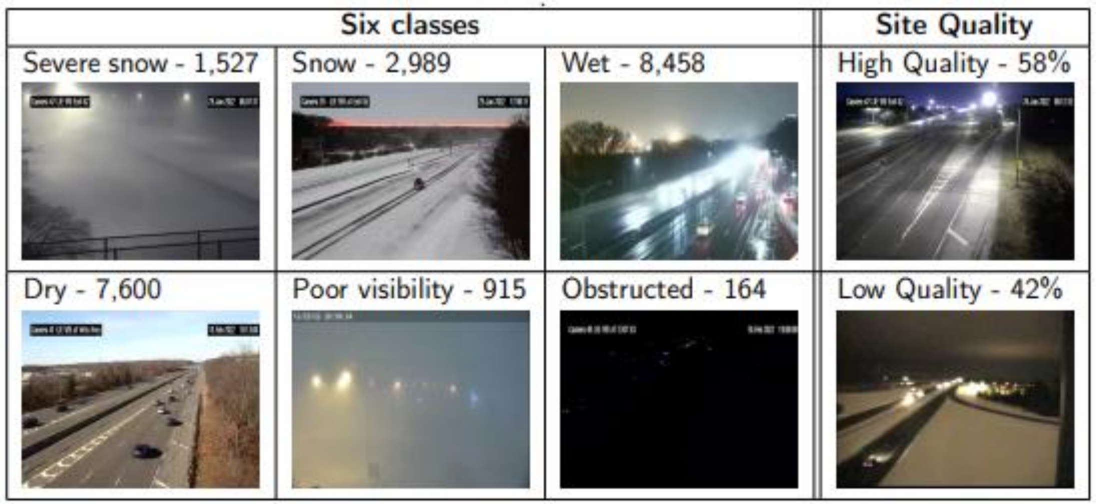
https://ethesisarchive.library.tu.ac.th/thesis/2021/TU_2021_6322040400_15943_20298.pdf

I: How is Image Processing Used in Engineering?

- Environmental & Weather Monitoring
  - Roadside/remote cameras monitor surface status (e.g., snow, flooding) for maintenance and driver safety alerts.



https://www.ugpti.org/resources/reports/downloads/mpc22-485.pdf



https://arxiv.org/pdf/2510.06440v1

I: Why do we care about image processing in everyday life?

- To save time and resources by automating previously manual tasks (e.g., infrastructure inspections, traffic monitoring).
- To make smarter, data-driven decisions by extracting actionable information from complex visual data (e.g., prioritizing repairs, planning for future growth).
- To improve safety in transportation systems through instant detection of hazards and incidents.
- To increase reliability and accuracy over human observation, especially for large-scale or high-speed scenarios (e.g., automated defect detection).
- To enable innovation, such as self-driving vehicles, smart cities, and predictive maintenance, by leveraging image-based analytics and AI.
- To reduce human error and enhance objectivity in engineering evaluations.

Section II:
What is an image?

II: What is an image?

- Common Understanding
  - An image is a **visual representation of objects**, scenes, or phenomena that we perceive through our eyes or capture using cameras, sensors, or other devices.

- Perceptual Definition
  - It's a 2D (two-dimensional) array of visual information containing **spatial details and intensity variations** that our brain interprets to understand the world around us.

II: What is an image?

- Everyday Examples:
  - A photograph you take with your smartphone
  - A screenshot from your computer
  - A medical X-ray or CT scan
  - A satellite image of Earth
  - A traffic camera feed
  - A thermal image from an infrared camera
- Key Characteristic
  - An image captures **information** about **light** intensity and **color** at different spatial locations, creating a representation of reality that preserves spatial relationships between objects.

II: What is an image?

- Technical Definition
  - A digital image is a discrete (sampled and quantized) 2D representation of a continuous scene, stored as a rectangular array of numerical values called **pixels** (picture elements).
- Mathematical Representation
  - An image *I* can be represented as a matrix or 2D function

I(x, y) = [intensity value at position (x, y)]

Where:

x	=	column index (horizontal position, typically 0 to width-1 for Python, 1 to width for MATLAB)
y	=	row index (vertical position, typically 0 to height-1 for Python, 1 to width for MATLAB)
I(x,y)	=	pixel intensity or color value at that location

II: What is an image?

- Mathematical Representation
  - Image as a Matrix: For an image with dimensions M×N:

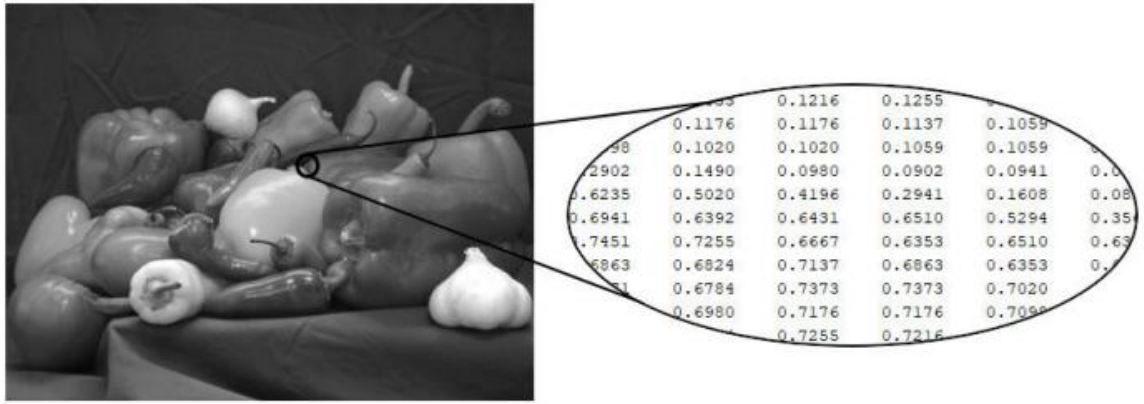
$$I = \begin{vmatrix} I(0,0) & I(0,1) & I(0,2) & \cdots & I(0,N-1) \\ I(1,0) & I(1,1) & I(1,2) & \cdots & I(1,N-1) \\ I(2,0) & I(2,1) & I(2,2) & \cdots & I(2,N-1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ I(M-1,0) & I(M-1,1) & I(M-1,2) & \cdots & I(M-1,N-1) \end{vmatrix}$$

Where:

M = number of rows (height in pixels)
N = number of columns (width in pixels)
Total number of pixels = M × N

II: Key Technical Components - Pixels

- The smallest unit of a digital image. Each pixel represents the intensity or color value at a specific spatial location in the image.
- **Pixel Value Range** (depends on bit depth):
  - **8-bit grayscale**
    - 0 to 255 (256 possible values)
    - 0 = black (minimum intensity)
    - 255 = white (maximum intensity)
    - 1-254 = various shades of gray
  - **16-bit grayscale**
    - 0 to 65,535 (65,536 possible values)
  - **32-bit floating point**
    - 0.0 to 1.0 (or any range)



MATLAB Image Processing Toolbox User's Guide

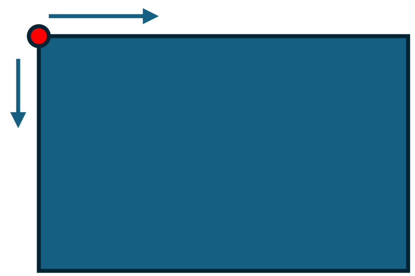
II: Key Technical Components - Pixels

- **Pixel Value Range** (depends on bit depth):
  - **RGB color image (8-bit per channel)**
  - Each pixel has 3 values (R, G, B), each 0-255
  - Example:
    - (255, 0, 0) = pure red
    - (0, 255, 0) = pure green
    - (0, 0, 255) = pure blue
    - (255, 255, 255) = white
    - (0, 0, 0) = black

MATLAB Image Processing Toolbox User's Guide

II: Key Technical Components - Pixels

- **Pixel Location**
  - Each pixel has a unique coordinate (x, y)
  - Origin (0, 0) is typically at the top-left corner of the image
  - x increases moving right (column index)
  - y increases moving down (row index)



- **Spatial Resolution**
  - The number of pixels determines image resolution
  - 1920×1080 pixels means 1920 columns × 1080 rows = 2,073,600 total pixels

II: Image Sampling and Quantization

Real-world scenes are **continuous** (infinite detail and infinite color variations), but digital images are **discrete** (finite number of pixels and finite color values).

- **Sampling**: The process of measuring the continuous scene at discrete spatial locations
  - A camera sensor takes measurements at specific pixel locations
  - Sampling rate determines image resolution (how many measurements per unit area)
- **Quantization**: The process of assigning discrete intensity values to continuous light measurements
  - Continuous light intensity (0.0 to ∞) is mapped to discrete levels (0 to 255 for 8-bit)
  - Bit depth determines how many levels are available

II: Image Sampling and Quantization

- **Trade-offs**

  - Higher sampling (more pixels) = better spatial detail but larger file size
  - Higher bit depth (more levels) = smoother gradations but larger file size
  - Lower sampling/bit depth = faster processing and smaller storage, but loss of quality

II: Image File Formats and Compression

- **Uncompressed Formats**
  - **RAW**: Raw sensor data, no processing, largest file size
    - Contains all captured information
    - Preferred in professional photography and medical imaging
  - **BMP (Bitmap)**: Simple format, each pixel stored separately
    - No compression = large file sizes
    - Used mainly for simple applications

II: Image File Formats and Compression

- **Lossy Compression** (some information discarded for a smaller file size)
  - **JPEG**: Joint Photographic Experts Group
    - Exploits human eye's lower sensitivity to color variations
    - Divides image into 8×8 blocks, applies DCT (Discrete Cosine Transform)
    - Quality factor (0-100%) controls compression level
    - Suitable for natural photographs, surveillance footage
    - NOT suitable when pixel-perfect accuracy is needed

II: Image File Formats and Compression

- **Lossless Compression** (no information lost)
  - **PNG**: Portable Network Graphics
    - Lossless compression using LZ77 algorithm
    - Preserves exact pixel values
    - Good for technical images, screenshots
  - **TIFF**: Tagged Image File Format
    - Flexible format supporting various bit depths
    - Can be compressed or uncompressed
    - Standard in professional/medical imaging
  - **GIF**: Graphics Interchange Format
    - Supports animation and limited color palettes
    - Good for simple graphics and web animation

II: Image File Formats and Compression

- **Comparison**
  - File Size: RAW > BMP > TIFF (uncompressed) > PNG ≈ TIFF (lossless) > JPEG
  - Quality: RAW = BMP ≈ TIFF ≈ PNG > JPEG (if high quality setting)
  - Processing Speed: JPEG (faster, fewer pixels to process) > RAW (slower)

II: Image Resolution and Size

- **Spatial Resolution**: Number of pixels in an image
  - M (height) × N (width) pixels
- **Common resolutions**:
  - HD (High Definition): 1280 × 720 pixels
  - Full HD: 1920 × 1080 pixels
  - 4K: 3840 × 2160 pixels
  - Mobile phone: typically, 2000-4000 × 3000-4000 pixels
- Higher resolution captures more detail
- Higher resolution = more data to store and process

II: Image Resolution and Size

- **Calculating file size**
  - File Size (bytes) = Height × Width × Number of Channels × Bytes per Channel
  - Example 1: Grayscale image (8-bit), 1920 × 1080
    - File Size = 1920 × 1080 × 1 × 1 = 2,073,600 bytes ≈ 2 MB
  - Example 2: RGB color image (8-bit per channel), 1920 × 1080
    - File Size = 1920 × 1080 × 3 × 1 = 6,220,800 bytes ≈ 6 MB
  - Example 3: RGB color image (16-bit per channel), 1920 × 1080
    - File Size = 1920 × 1080 × 3 × 2 = 12,441,600 bytes ≈ 12 MB

II: Wrap Up - Anatomy of a Digital Image

- **A digital image is**
  - A 2D array of numbers (matrix)
  - Where each number represents the **intensity** or color at a spatial **location**
  - With discrete spatial sampling (finite resolution) and discrete intensity values (finite bit depth)
- **Key parameters**
  - **Spatial dimensions**: M × N (or M × N × C for color)
  - **Bit depth**: How many bits per pixel/channel (8, 16, 32, etc.)
  - **Color space**: Grayscale, RGB, HSV, CMYK, etc.
  - **File format**: How it's stored (RAW, PNG, JPEG, etc.)
  - **Coordinate system**: How we index pixels (typically top-left origin)

II: Wrap Up - Anatomy of a Digital Image

- **images are data**
  - Images are structured data we can analyze, manipulate, and process
  - All image processing operations are mathematical transformations on these numerical matrices
  - Processing complexity depends on image size and the operations applied
- **Considerations**
  - Sampling: Must be fine enough to capture relevant details
  - Quantization: Must have sufficient levels to represent tonal variations
  - Compression: Must balance file size with quality requirements
  - Format choice: Depends on application needs (medical, surveillance, printing, display, etc.)

Section III:
What is image processing?

III: What is Image Processing?

- **Image processing** is the computational manipulation of digital images to extract useful information, enhance quality, or transform images into a more suitable form for analysis or display.

- **Core Concept**
  - Images are numerical matrices; image processing is fundamentally the application of mathematical operations and algorithms to these matrices to achieve specific goals.

III: What is Image Processing?

- **Two Main Goals**

  - **Image Enhancement:** Improve image quality or make certain features more visible
    - Example: Reducing noise, increasing contrast, sharpening edges

  - **Image Analysis & Extraction:** Extract meaningful information from images
    - Example: Detecting objects, counting items, measuring distances

All image processing operations are mathematical transformations applied to pixel values

III: Image Processing vs Image Analysis

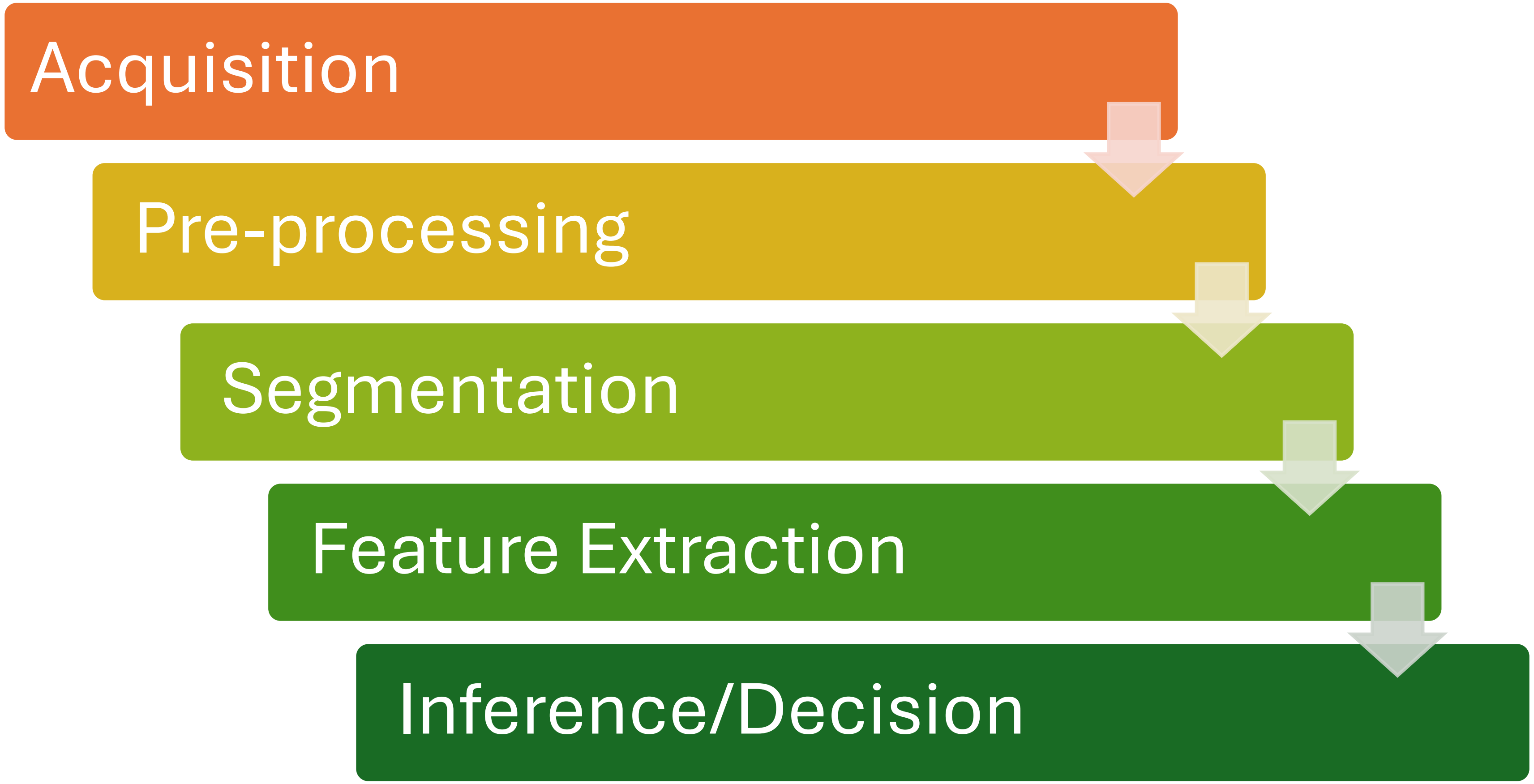
- **Image Processing (Lower-level operations)**
  - Works directly on pixel values
  - Focuses on **transforming** image data
  - Output is typically another image
  - Examples: Filtering, smoothing, sharpening, resizing, rotating
- **Image Analysis (Higher-level operations)**
  - Works on processed images or extracted features
  - Focuses on **understanding** what's in the image
  - Output is typically data or decisions (numbers, classifications, detections)
  - Examples: Object detection, counting, measuring, classification

In Practice: Often combined in a pipeline

Image Processing: prepare the image → Image Analysis: extract meaning from it

III: The Image Processing Pipeline

- **Typical Image Processing Workflow** (the pipeline)



III: The Image Processing Pipeline

- **Stage 1: Acquisition**
  - Capture the image using cameras, sensors, or other devices
  - Choose appropriate resolution, color space, file format
  - **Transportation example**: Traffic camera captures video frame

- **Stage 2: Pre-processing**
  - Prepare the image for analysis
  - Common operations: Noise reduction, contrast enhancement, resizing
  - Goal: Improve image quality and remove unwanted artifacts
  - **Transportation example**: Remove noise from surveillance footage, enhance contrast to see vehicles better

III: The Image Processing Pipeline

- **Stage 3: Segmentation**
  - Divide the image into meaningful regions or objects
  - Separate what you want to analyze from the background
  - Techniques: Thresholding, edge detection, clustering
  - **Transportation example**: Separate vehicles from the road background

- **Stage 4: Feature Extraction**
  - Identify and measure important characteristics of segmented objects
  - Extract quantifiable information from each region
  - Examples: Size, shape, color, texture, position
  - **Transportation example**: For each detected vehicle, extract: size, color, position, speed

III: The Image Processing Pipeline

- **Stage 5: Inference/Decision**
  - Make final decisions or predictions based on extracted features
  - Apply machine learning, rules, or algorithms
  - Produce actionable output (classifications, counts, alerts)
  - **Transportation example**: Classify vehicle type, count traffic volume, detect congestion, trigger alerts

III: Common Image Processing Techniques

- **Pixel-level Operations** (simplest level)
  - Operate on individual pixels independent of neighbors
  - Change brightness, contrast, or color values
  - Examples: Thresholding, intensity scaling, histogram equalization
- **Local Operations** (neighborhood-based)
  - Operate on each pixel and its neighboring pixels
  - Considers spatial relationships
  - Most common category of image processing
  - Examples: Filtering, edge detection, morphological operations
- **Global Operations** (entire image)
  - Consider all pixels in the image
  - Modify entire image structure or content
  - Examples: Fourier transform, image resizing, rotation

III: When and Why to Use Image Processing

- **Image Processing is ideal when**
  - You have visual data that needs analysis
  - You need to automate visual inspection or monitoring
  - Manual processing is time-consuming or impractical
  - Consistency and speed are important
  - You want objective, repeatable results

III: When and Why to Use Image Processing

- **Advantages**
  - **Automation**: Process thousands of images quickly
  - **Consistency**: Same algorithm applied uniformly (no human fatigue)
  - **Scalability**: Once developed, apply to new data at minimal cost
  - **Objectivity**: Mathematical, not subjective like human observation
  - **Integration**: Results can feed into automated systems and decisions
  - **24/7 Operation**: Continuous monitoring without human presence

III: When and Why to Use Image Processing

- **Limitations and Challenges**
  - Requires careful algorithm tuning for different conditions
  - Lighting, weather, season changes affect results
  - Shadows, occlusion, reflections cause difficulties
  - Can be computationally expensive for high-resolution or real-time applications
  - Requires ground truth data for validation/training
  - Not suitable for tasks requiring human judgment or context

III: Image Processing vs Other Approaches

- **Why Image Processing Instead of Other Methods?**
  - **vs Point Sensors/Detectors**:
    - Sensors: Detect presence/absence at fixed location
    - Image processing: Provides spatial information, can track movement
    - **Transportation**: Cameras see entire intersection; sensors only detect at one point
  - **vs. Manual Inspection**
    - Manual: Time-consuming, subjective, inconsistent, limited scale
    - Image processing: Fast, objective, consistent, scalable
    - **Transportation**: Inspect entire highway vs. manual pothole inspection

III: Image Processing vs Other Approaches

- **Why Image Processing Instead of Other Methods?**
  - **vs Point Sensors/Detectors**:
    - Sensors: Detect presence/absence at fixed location
    - Image processing: Provides spatial information, can track movement
    - **Transportation**: Cameras see entire intersection; sensors only detect at one point
  - **vs. Manual Inspection**
    - Manual: Time-consuming, subjective, inconsistent, limited scale
    - Image processing: Fast, objective, consistent, scalable
    - **Transportation**: Inspect entire highway vs. manual pothole inspection

Section IV:
Basic Image Processing Techniques

IV: MATLAB Advantages for Image Processing

- Extensive Image Processing Toolbox with pre-built functions
- Visual results: see exactly what each operation does
- Fast prototyping and experimentation
- Integration with other tools for analysis
- Industry standard in engineering

IV: Hands-On Experience

- We will explore key MATLAB functions for image processing
- The concepts and tools we learn today will pave the way for next week's hands-on applications in transportation engineering.

Thank you for your attention!

siwarak.u@chula.ac.th

CE Building, Room 534