Introduction to Machine Vision

Machine Vision

Dr. Mostapha Kalami Heris

m.k.heris@shu.ac.uk

School of Engineering and Built Environment

Sheffield Hallam University

Module Overview

- **Level 7 Module**
- **20 credits**
- Delivered via **Lectures + Labs**
- Module Team:
  - Dr. Mostapha Kalami Heris (Module Leader)
  - Dr. Hajar “Helga” Razaghi
- Coursework: **100%** (submitted after Week 12)
- Learning Outcomes:
  - **LO1:** Basics of image processing & machine vision (today's focus)
  - **LO2 and LO3:** Applications + Hardware/Software (later)

Coursework Alignment

- **Part A** (early weeks)
  - I/O, display literacy
  - Justify methods
- **Part B** (later)
  - Connected components
  - Measurement & analysis

⚠️ Habits from Week 1 carry forward

Sheffield Hallam University

About Me

- My name is **Mostapha Kalami Heris**.
- **PhD in Electrical Engineering** (Control Systems)
- Expertise in **Artificial Intelligence, Machine Learning, and Intelligent Control Systems**
- **20+ years industry experience** in roles such as:
  - Senior Software Engineer
  - Machine Learning Engineer
  - AI Scientist and Researcher
- Currently **Lecturer at Sheffield Hallam University**

Why This Module Matters

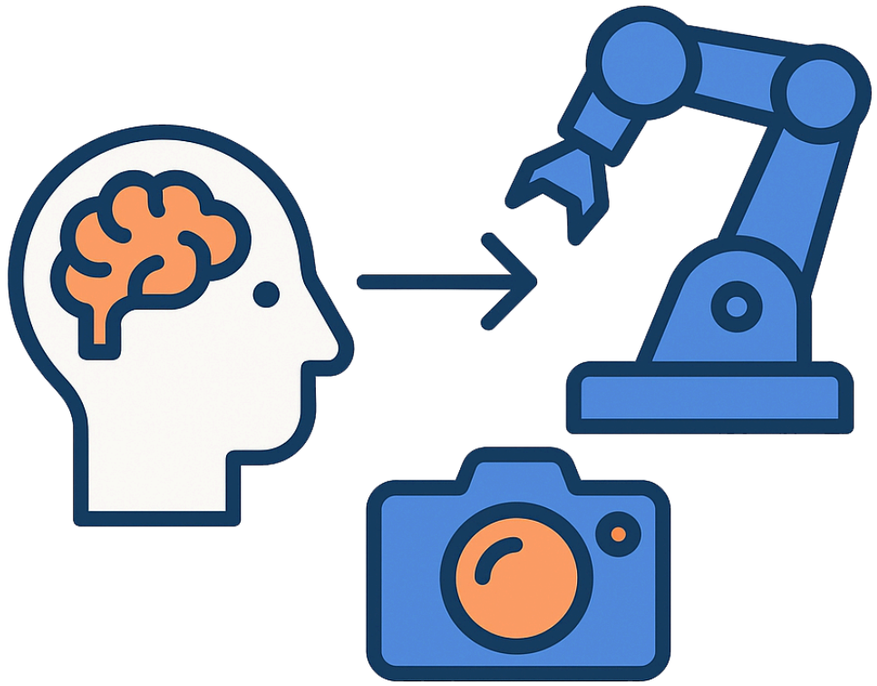
- Machines now **see, analyze, decide**
- From **robots** to **healthcare**
- Vision = **bridge** between perception & action
- Core skill for **automation, AI, robotics**

Sheffield Hallam University

Motivation & Applications

Why Machine Vision, Why Now?

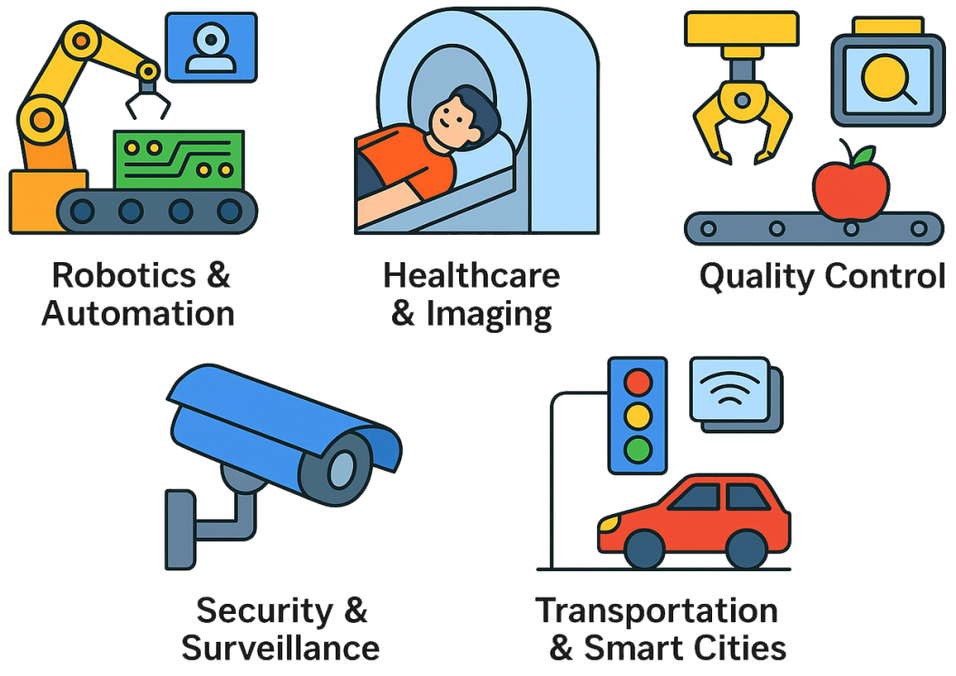
- Humans: **context, flexibility**
- Machines: **speed, scale, precision**
- Machine Vision links **perception → action**



Sheffield Hallam University

Applications of Machine Vision

- Robotics & Automation
- Healthcare & Imaging
- Quality Control in factories
- Security & Surveillance
- Transportation & Smart Cities



Quick Prompt

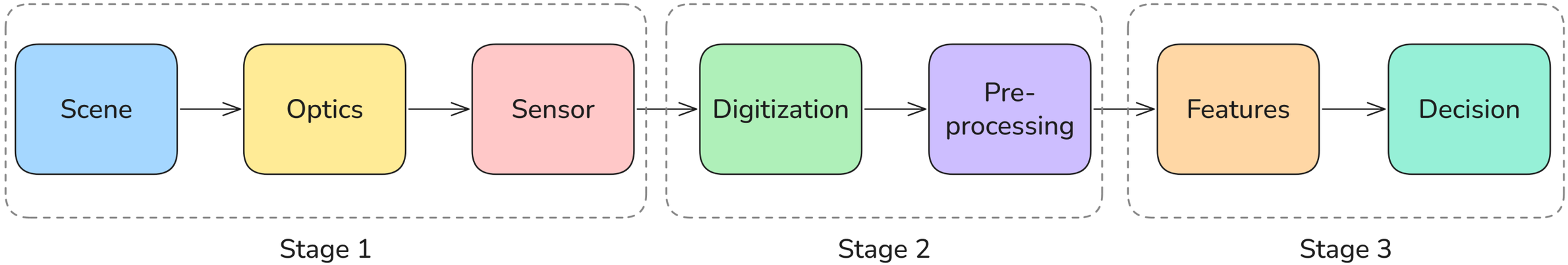
What is the decision each system must output?

- Robot in a factory → ?
- Medical scan system → ?
- Self-driving car → ?

Think about: **classification, detection, measurement**.

The Imaging Pipeline

Pipeline Overview

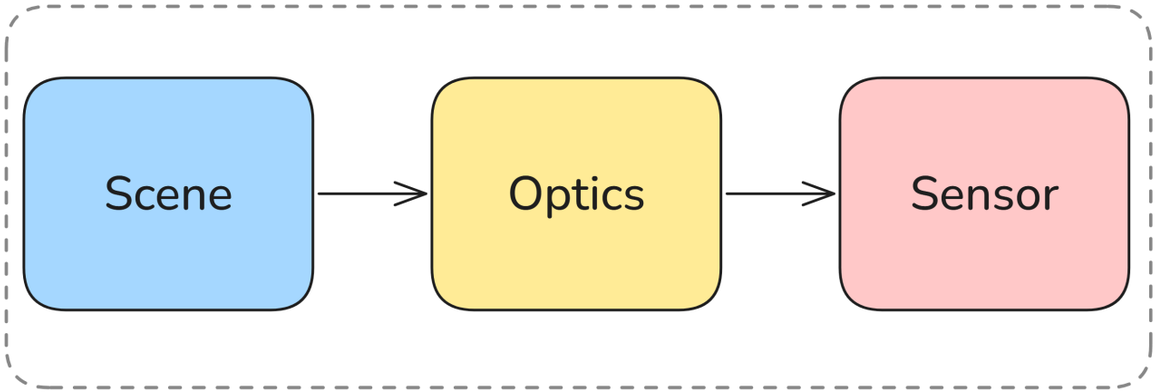


- From **Scene → Decision**
- Standard scaffold across the module
- Each stage has **strengths & failure modes**

Stage 1: Scene → Optics → Sensor

- Light from **scene**
- Modified by **lenses / optics**
- Captured by **sensor**

⚠️ Failure mode: **blur / poor focus**

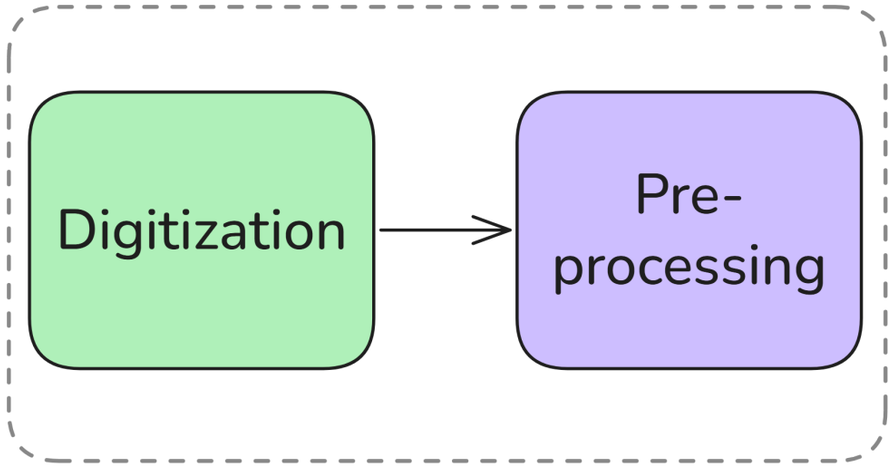


Stage 1

Stage 2: Digitization → Pre-Processing

- **Digitization**: analog → digital
- **Pre-processing**: noise reduction, scaling
- Prepares data for **analysis**

⚠️ Failure mode: **quantization error**

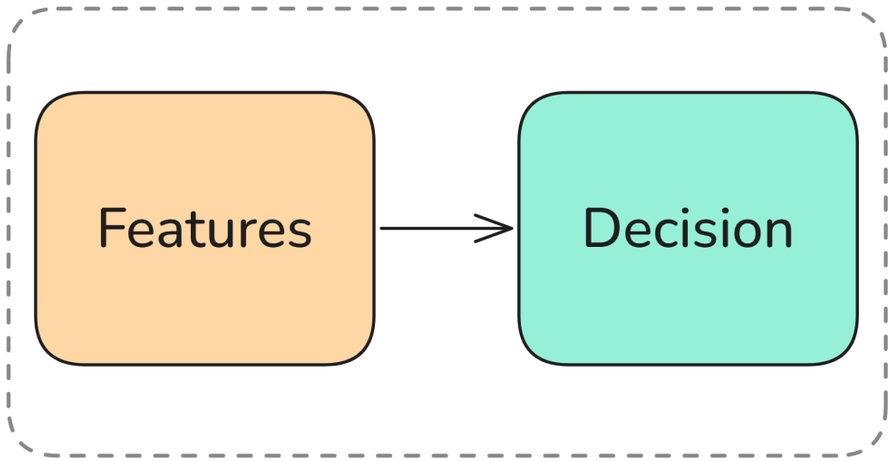


Stage 2

Stage 3: Features → Decisions

- Extract **edges, regions, textures**
- System outputs a **decision / label**
- Bridge to **control & action**

⚠️ Failure mode: **wrong features = wrong decision**



Stage 3

Quick Prompt

> **Where in the pipeline would you expect most errors in practice?**

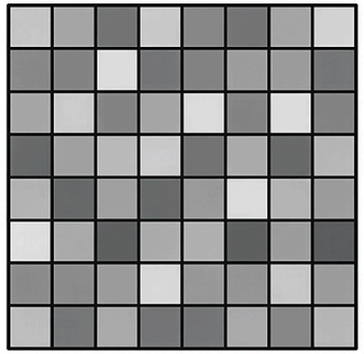
- Optics?
- Digitization?
- Features?

Think about **hardware vs software challenges**.

What is an Image in MATLAB?

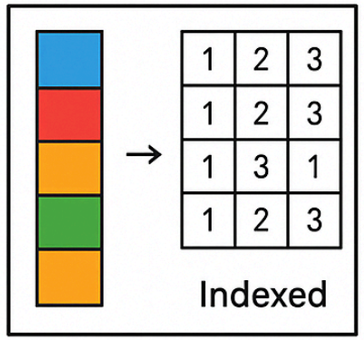
Image Representations

- **Grayscale** (M×N)
- **True Color** (M×N×3)
- **Indexed** (map + index array)
- **Binary** (0/1 only)

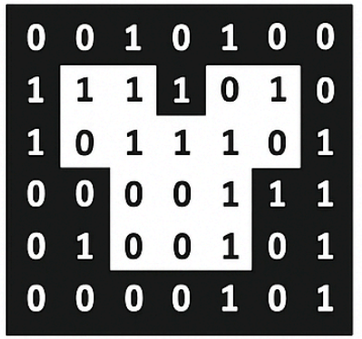


Grayscale

True Color



Indexed

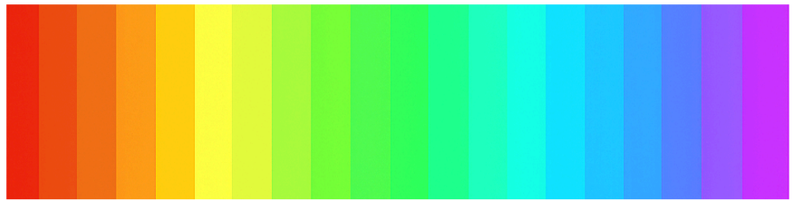


Binary

Data Types & Ranges

- **uint8** → integers from 0 to 255
- **uint16** → integers from 0 to 65,535
- **double** → numbers between 0 and 1 (scaled)

⚠️ Wrong type = wrong display



Quick Prompt

What happens if you cast an image to double but do not scale it?

- Think about **range vs display**.
- Hint: MATLAB assumes **0 – 1 for doubles**.

Hands-On with MATLAB

Basics: Read, Display, and Save

- `imread` → read image file
- `imshow` → uses raw data values
- `imagesc` → rescales to display range
- `imwrite` → save image file

> You can add **colorbar** for clarity.

Key MATLAB Image Functions

- `im2gray` → Convert image to grayscale (recommended)
- `rgb2gray` → RGB (M×N×3) to grayscale (M×N)
- `mat2gray` → Rescale data to [0,1] (for safe display of doubles)
- `rgb2hsv` → Convert RGB to HSV (hue, saturation, value)
- `hsv2rgb` → Convert HSV to RGB

👉 Use carefully: conversions may change **appearance & meaning**

Micro-Demo Script

MATLAB Code:

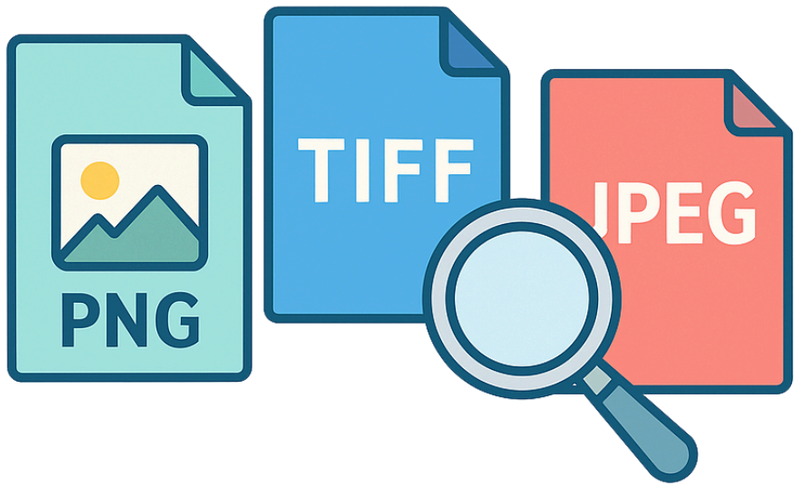
```MATLAB
I = imread('ngc6543a.jpg');   % Read image
size(I)                       % Check dimensions
class(I)                      % Check data type

imshow(I);                    % Display raw data
imagesc(I);                   % Display scaled data
imshow(rgb2gray(I));          % Convert to grayscale
imshow(I(:,:,1));             % Show red channel

imwrite(I, 'example.tif');    % Save image as TIFF
```

Formats & Compression

- **PNG** and **TIFF** → lossless (keep detail)
- **JPEG** → lossy (OK for presentation)
- `imfinfo` function → check metadata
- Record format choice in the **lab**



Quick Prompt

Which format preserves 16-bit detail?

- PNG
- JPEG
- TIFF

Wrap-Up & Support

Sheffield Hallam University

Key Takeaways

- Why machine vision matters today
- Imaging pipeline: scene → decision
- Images as **data + types + ranges**
- Safe display & scaling (`imshow` vs `imagesc`)
- MATLAB basics: read, inspect, save safely
- File formats & compression (lossless vs lossy)

Exit Ticket

✍️ Before you leave, write down:

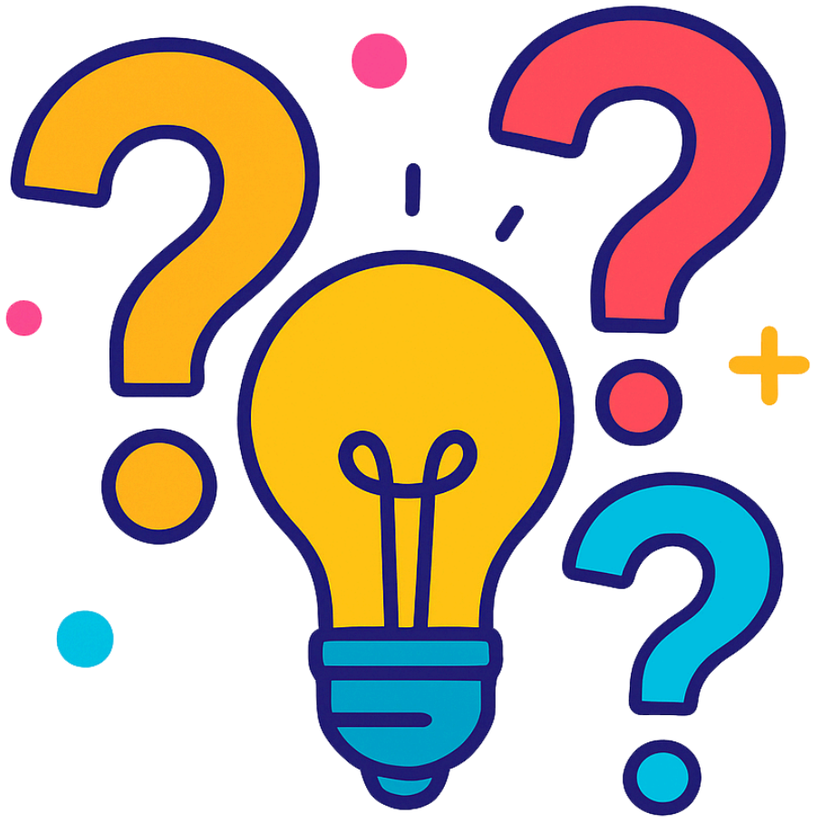
1. **One clear concept** from today
2. **One confusing concept**
3. **One thing to try in MATLAB** this week

Sheffield Hallam University

Questions & Support

- You are encouraged to **ask questions anytime** 💡
  - During the lecture
  - In lab sessions
  - By email → m.k.heris@shu.ac.uk
- **No question is too simple** — asking questions:
  - Helps you learn faster
  - Builds confidence
  - Improves understanding for the whole class

👉 If something is unclear, **just ask!**



Sheffield Hallam University