Machine Vision

Applying the Machine Vision Pipeline in MATLAB

Lab Activity Sheet 6

Author: Dr. Mostapha Kalami Heris

Introduction

This lab session extends the concepts introduced in the lecture on the machine vision pipeline. The aim is to give you an opportunity to experiment with different stages of image processing and see how they connect in practice.

The machine vision pipeline includes several important stages:

- **Preprocessing** to prepare the image,
- **Segmentation** to separate meaningful regions,
- **Morphological operations** to clean and refine results,
- **Measurement** to extract useful information.

Each of these stages influences the next, and small changes in one part of the pipeline can affect the final outcome. In this lab you will explore these ideas through short, open-ended activities.

This activity sheet is self-contained. Each activity includes:

- a short explanation of the goal,
- suggested MATLAB functions,
- space for you to experiment with different parameters and approaches.

You are encouraged to:

- use an image of your choice,
- try different methods and compare results,

- read MATLAB documentation for any unfamiliar functions,
- ask questions during the session whenever needed.

The three activities in this lab focus on:

1. illumination correction and segmentation,
2. object detection and measurement,
3. recognising text or repeated patterns.

These tasks reflect common problems in machine vision and will help you understand how early processing choices affect later stages. You may begin with any activity and explore them in any order.

Activity 1: Illumination Correction and Segmentation

Objective

To explore how uneven illumination affects segmentation and to experiment with simple methods for correcting background variations before applying thresholding.

Key Idea

Segmentation works best when the background is uniform. Shadows, gradients, and reflections can cause global thresholding to fail. By estimating and removing the background first, you can achieve more reliable results.

What You Will Do

- Choose or create a meaningful image with noticeable changes in brightness.
- Try one or more methods for illumination correction.
- Apply global or adaptive thresholding.
- Compare different results and observe how preprocessing influences segmentation.

Suggested MATLAB Functions

- `im2double`, `rgb2gray`
- `strel`, `imopen`, `imtophat`
- `imsubtract`, `imadjust`
- `imbinarize`, `adaptthresh`
- `bwareaopen`, `imshowpair`

Guided Steps

1. Load your image and convert it to grayscale if needed.
2. Estimate the background using a large structuring element or top-hat filtering.
3. Subtract the estimated background and observe the corrected image.
4. Apply global thresholding and adaptive thresholding.
5. Compare the segmentation results visually.

Things to Explore

- How does the size of the structuring element affect the correction?
- When does adaptive thresholding perform better than global thresholding?
- How sensitive are the results to noise or shadows in your chosen image?

Activity 2: Object Detection and Measurement

Objective

To experiment with detecting meaningful objects in an image and to explore how simple measurements can be extracted after segmentation or feature detection.

Key Idea

Object detection can be based on shape, contrast, or region properties. Once objects are identified, MATLAB can calculate features such as area, centroid, circularity, or bounding boxes. These measurements depend strongly on the quality of preprocessing and segmentation.

What You Will Do

- Select an image that contains several objects, either circular or irregular.
- Try different detection methods, such as circle detection or region-based analysis.
- Examine the measurements returned by MATLAB for each object.
- Compare how different preprocessing choices affect detection.

Suggested MATLAB Functions

- `imfindcircles`, `viscircles`
- `imbinarize`, `edge`
- `bwlabel`, `bwconncomp`
- `regionprops` (area, centroid, circularity, bounding box)
- `bwboundaries`, `imshowpair`

Guided Steps

1. Load your image and inspect its structure and contrast.
2. If the image contains round shapes, try using `imfindcircles` to detect them.
3. Alternatively, create a binary mask using thresholding and refine it with basic morphology.
4. Label the connected components and extract measurements using `regionprops`.
5. Overlay the detected objects and their measurements on the image.

Things to Explore

- How do different radii or sensitivity settings influence `imfindcircles`?
- What happens when the segmentation contains noise or small gaps?
- Which measurements are most meaningful for the objects you selected?

Activity 3: Text and Pattern Recognition

Objective

To explore how text or repeated patterns can be recognised in images, and to observe how preprocessing steps influence the quality of recognition.

Key Idea

Optical Character Recognition (OCR) and pattern detection rely on clear, well-segmented features. Uneven illumination, noise, or shadows can cause missing or incorrect detections. Preprocessing often plays a crucial role in improving recognition results.

What You Will Do

- Choose an image that contains meaningful text or repeated symbols.
- Apply preprocessing steps to enhance contrast and remove background variations.
- Run OCR or attempt simple pattern detection.
- Inspect the confidence values or detection output and compare results before and after preprocessing.

Suggested MATLAB Functions

- `ocr`, `insertObjectAnnotation`
- `imbinarize`, `adaptthresh`
- `imtophat`, `imreconstruct`
- `bwareaopen`, `edge`
- `imshowpair`

Guided Steps

1. Load your image and convert it to grayscale if needed.
2. Experiment with illumination correction or contrast enhancement.
3. Apply thresholding to create a clearer foreground.

4. Run `ocr` and examine the recognised text or confidence scores.

5. Compare the results from different preprocessing choices.

Things to Explore

- Which preprocessing steps make the greatest improvement to OCR?
- How does the placement or font of text influence recognition?
- Can you reduce false detections by removing small artifacts?
- What happens when the image contains patterns instead of text?

Summary

This lab session allows you to explore several stages of the machine vision pipeline through short, practical activities. By working with images of your choice, you can see how preprocessing, segmentation, morphology, and measurement are connected and how each stage influences the next.

In this session, you:

- experimented with illumination correction and observed how it affects segmentation quality,
- applied region-based and shape-based methods to detect and measure objects,
- explored text or pattern recognition and examined how preprocessing influences the results.

These activities introduce ideas that are common in many computer vision applications. You are encouraged to continue experimenting with different images, parameters, and functions to deepen your understanding. Small changes in preprocessing can produce large differences in the final output, and exploring these effects is an important part of developing practical machine vision skills.

The functions and techniques introduced here form a useful foundation for future work in this module. You may revisit any of the activities as you progress through later topics, especially when analysing more complex images or constructing your own vision workflows.