Autonomy: Machine vision

Prof. Mirko Kovač
Dr. Kevin Holdcroft
Laboratory of Sustainability Robotics
EPFL, Switzerland

About me

BASc: Mechatronic Systems Engineering

Jr. Application Engineer: Smart buildings

MSc: Systems, Control and Robotics

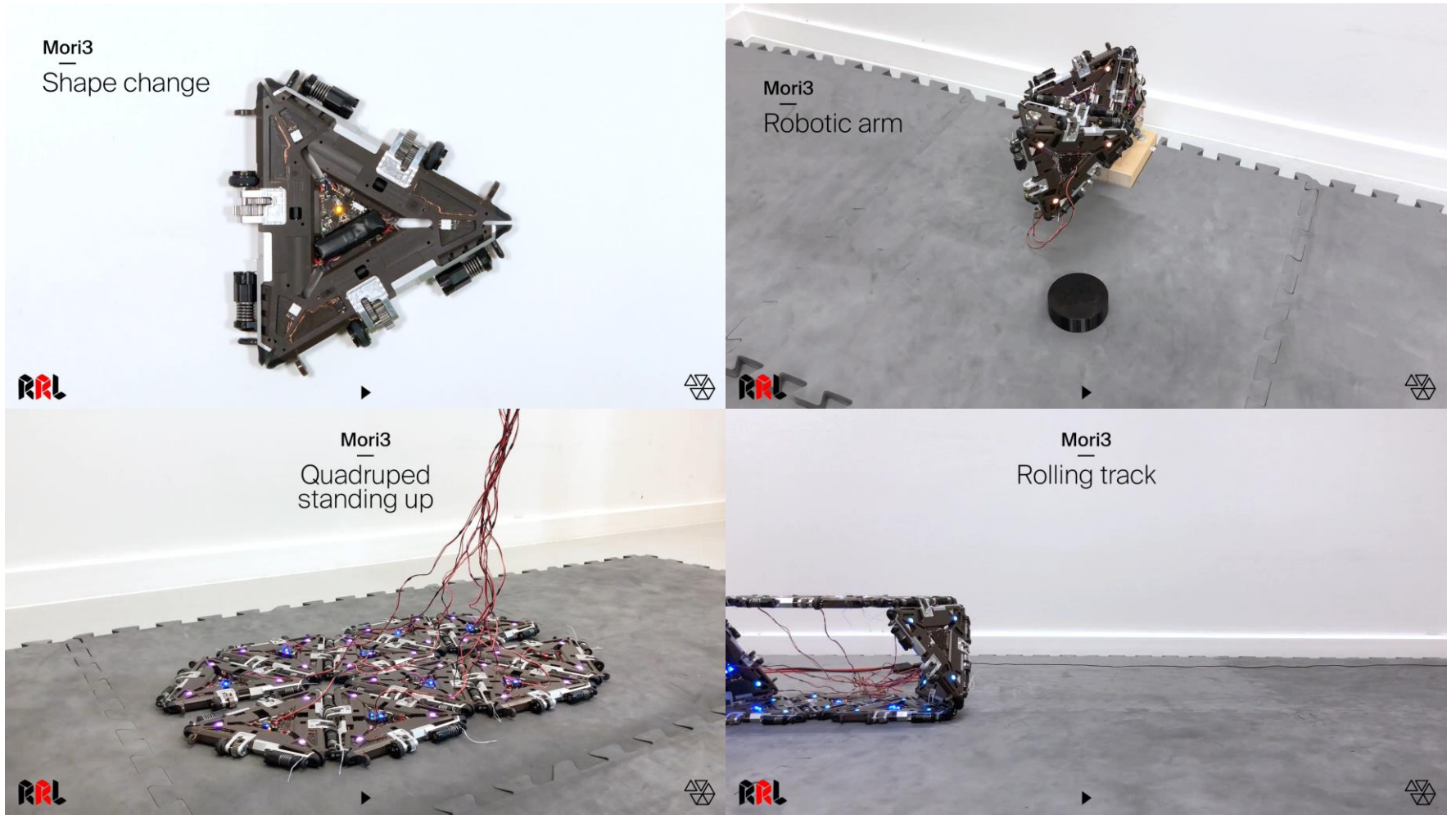
Intern: Porus media group (fuel cells)

PhD: Robotics, Control, Intelligent Systems

PostDoc: Sustainability Robotics

ENV-530 Sustainability Robotics

Mori3: A polygon-based modular robot



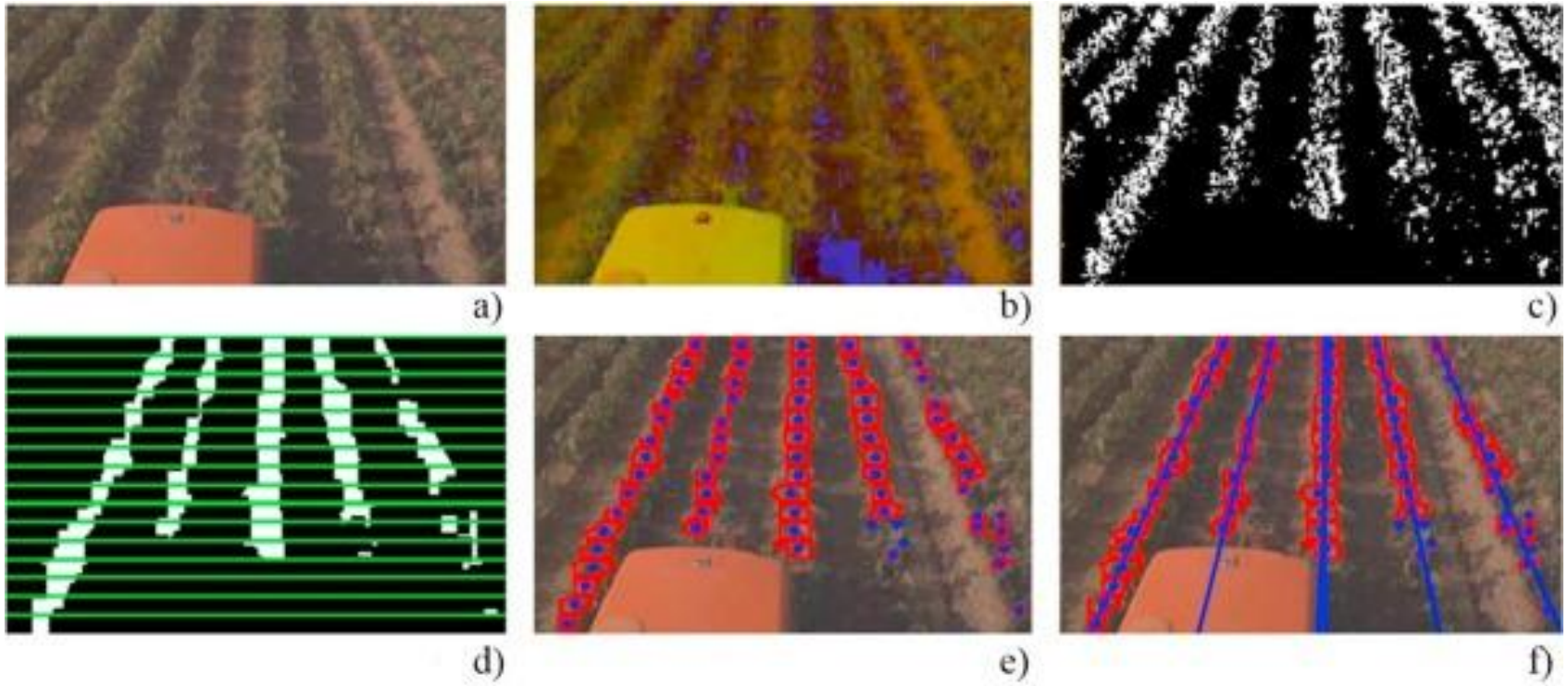
C. H. Belke, K. A. Holdcroft, A. Sigrist, and J. Paik, “Morphological flexibility in robotic systems through physical polygon meshing”, *Nature Machine Intelligence*, pp. 1–7, 2023

Learning goals

- Learn different colour models and how they are applied
- Identify the underlying principles for basic image manipulation techniques
- Understand the core concept of edge and line detection
- Discuss applications of computer images within agriculture

Motivation

- Understand the steps between each of these images



Why colour imaging

- Standard sensor, similar to eye
  - Cheap, multipurpose
  - *Pigments change in plants for harvest*

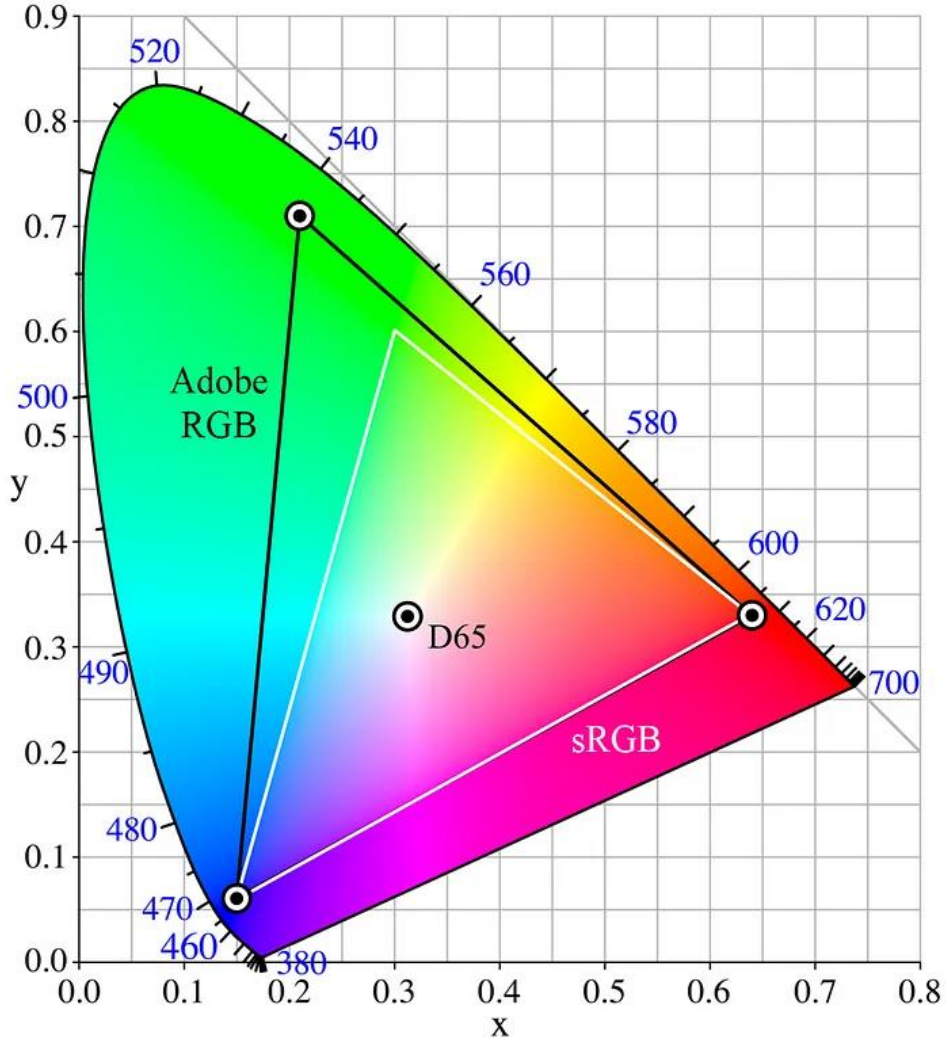
- Challenges
  - Lighting
  - Occlusion
  - Object movement

1) https://ucanr.edu/datastoreFiles/234-2818.pdf

Section: Colour Models

Colour models

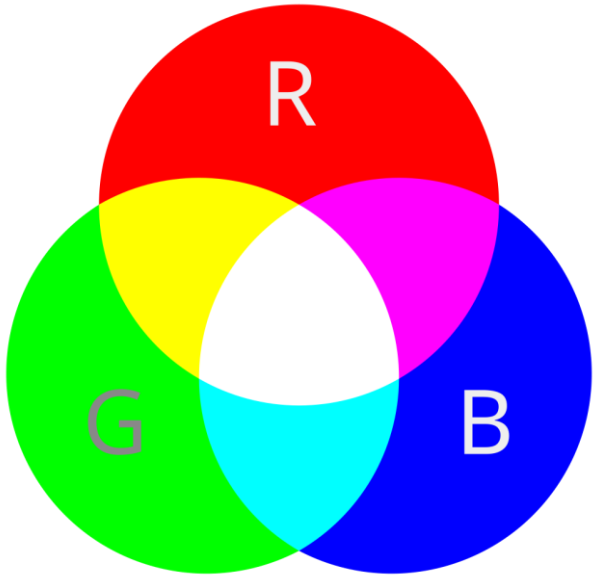
- What are colour models?
  - Different ways to represent a colour
  - Different ways to *digitise* an image
- Why different models?
  - Different functions
    - Imitate human eyes, imitate inks
    - Different understandings
  - Each model captures different parts of colour space

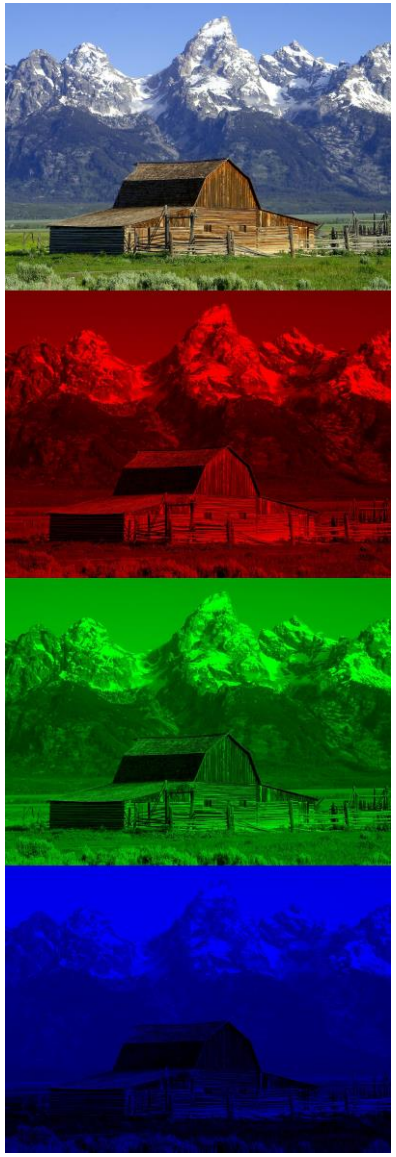


1) https://en.m.wikipedia.org/wiki/File:CIE1931xy_gamut_comparison.svg
2) https://medium.com/hipster-color-science/a-beginners-guide-to-colorimetry-401f1830b65a

Colour models – RGB model

- Red, Green, Blue
  - *Additive* colour model
  - Similar to cones in an eye
- Wavelength of primary colours arbitrary
  - Related to human physiology
  - Or related to acquisition device



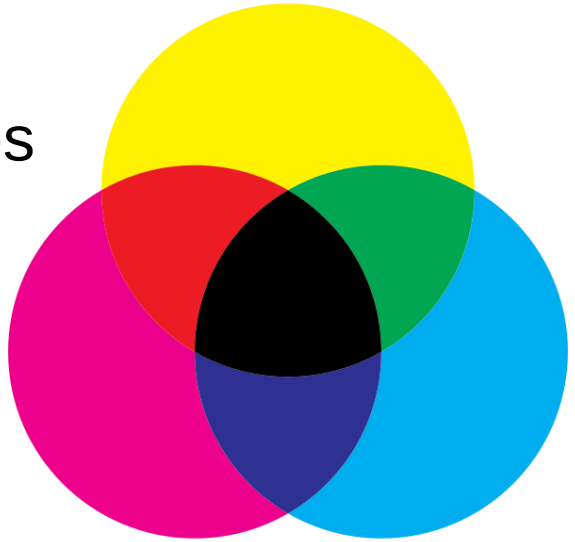


1) https://en.wikipedia.org/wiki/RGB_color_model

Colour models – CMYK model

- Cyan, Magenta, Yellow, and Key (black)
  - *Subtractive* colour model
  - CMY: without key

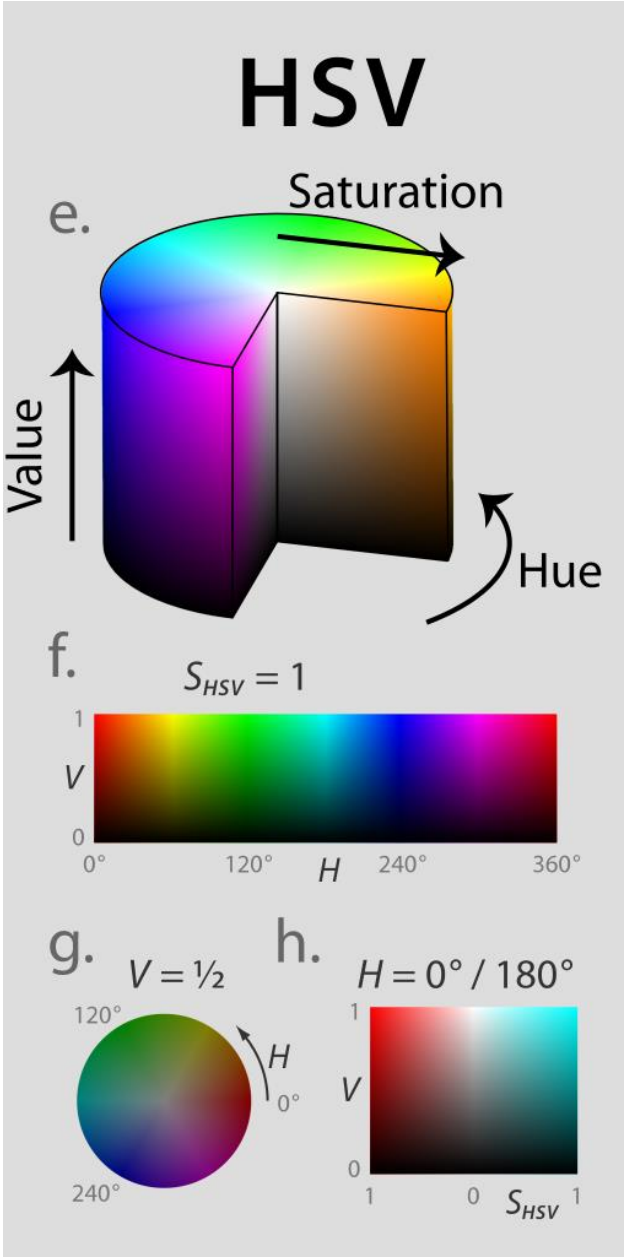
- Used for colour printing
  - Based on how paint / ink mixes



1) https://en.wikipedia.org/wiki/CMYK_color_model

Colour models – HSV model

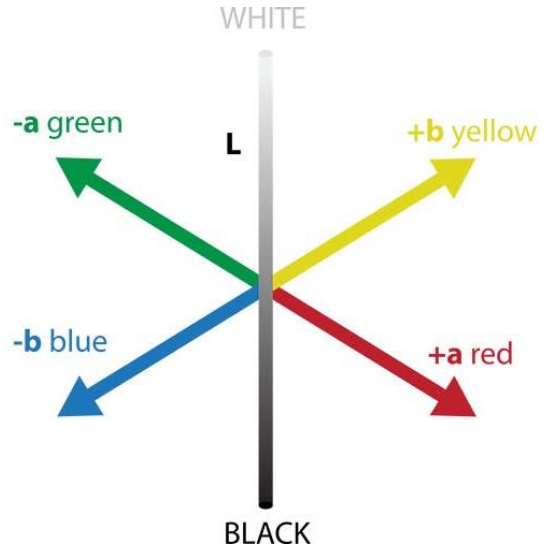
- Hue, Saturation, Value
  - E.g., Colour, Intensity, !Lightness
- Wraps RGB model around a cylinder
- Motivation
  - Reduce intensity without changing 3 different colour values

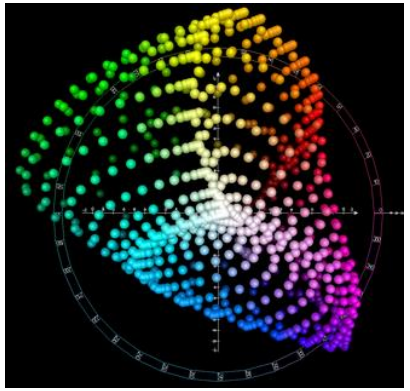


1) https://en.wikipedia.org/wiki/HSL_and_HSV

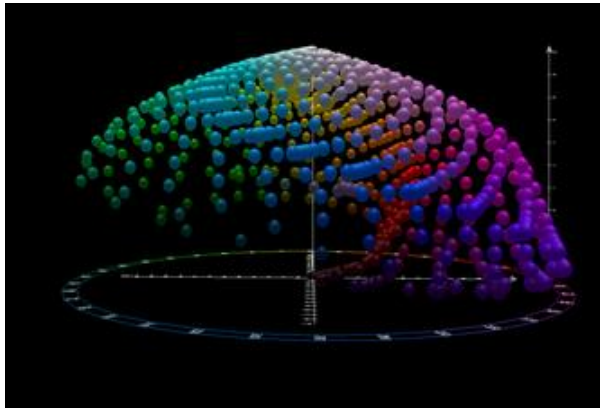
Colour models – CIELAB model (L*a*b*)

- International Commission on Illumination (CIE)
- Colour relative to CIE standard observer
  - Designed to approximate human vision
  - Better for small differences in colour
- Three coordinates
  - L* Lightness
  - a* red to green
  - b* blue to yellow





Top view

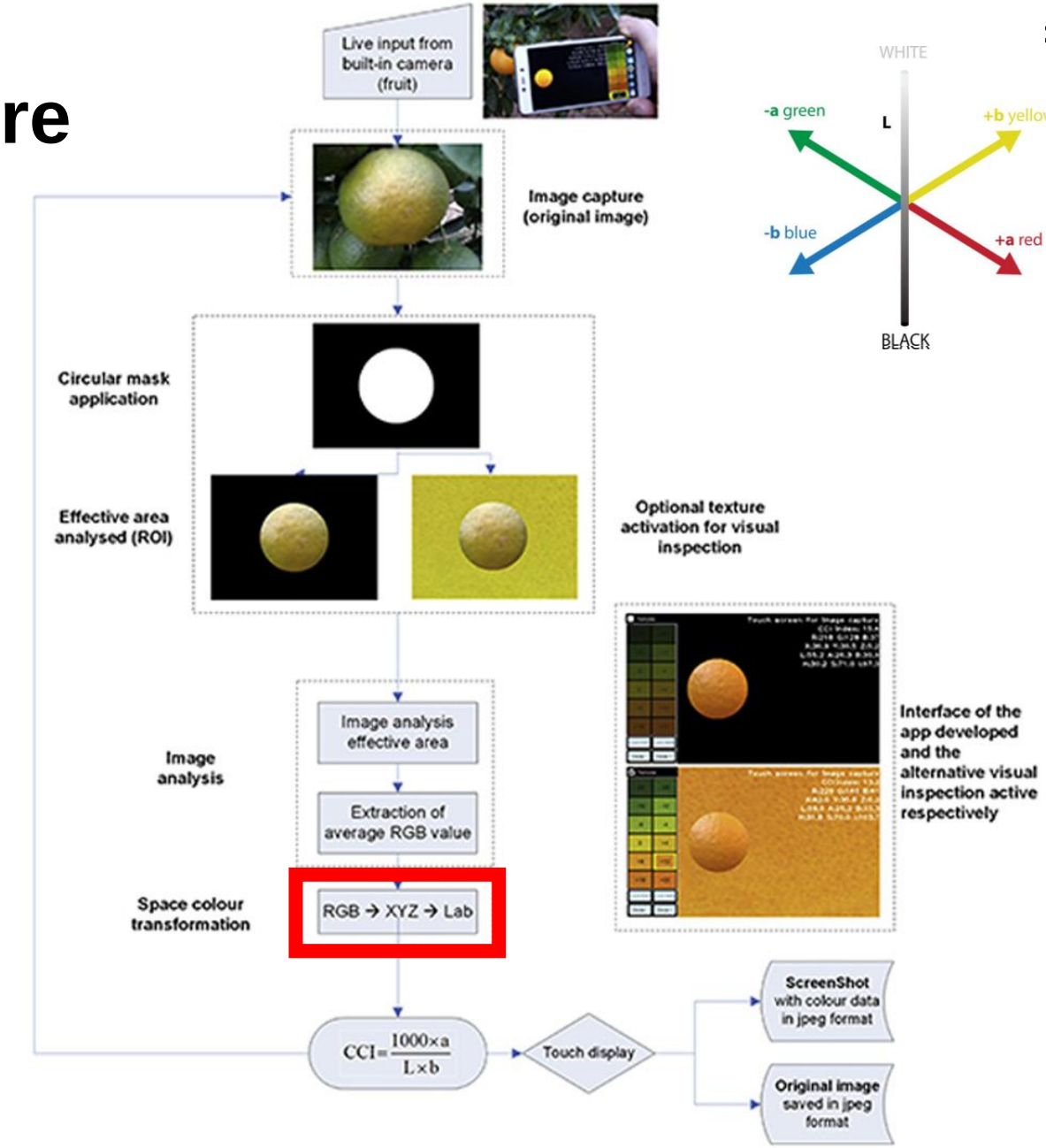


Side view

1) https://www.datacolor.com/business-solutions/blog/what-is-cielab/

CIELAB model in agriculture

- Changes from green to red
  - Maturity of fruits
- Applications
  - Citrus colour index (CCI)
  - Colour of tomatoes
  - Browning index
  - Compost stability

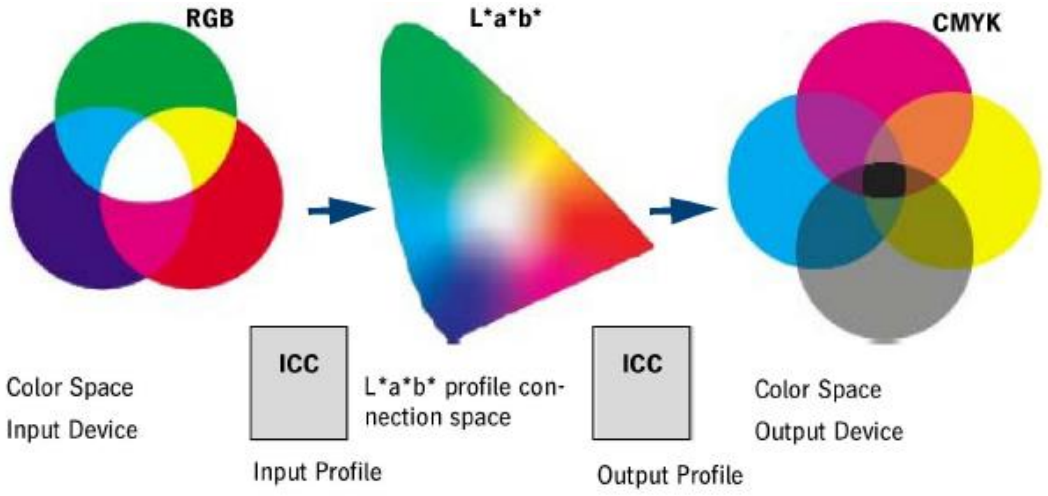


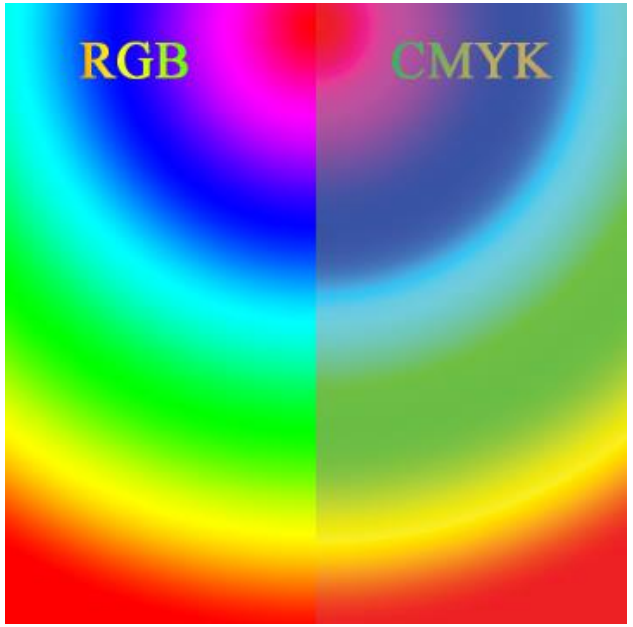
Citrus colour index to classify orange ripeness

1) https://www.sciencedirect.com/science/article/abs/pii/S1537511017310103

Colour models – conversion

- Color space conversion
  - Colours in one representation are not necessary available in the other ones
- Color conversion usually uses CIELAB



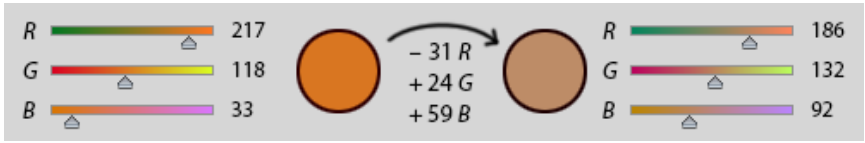


1) https://en.wikipedia.org/wiki/Color_space

Colour models – Comparison

- Comparison between two colours: Delta E (ΔE)
  - Euclidian distance
  - Not useful for nonperceptual colour spaces (e.g., RGB)

- CIE formula for comparison:



$$\Delta E* = \left(\left(L*_1 - L*_2 \right)^2 + \left(a*_1 - a*_2 \right)^2 + \left(b*_1 - b*_2 \right)^2 \right)^{1/2}$$

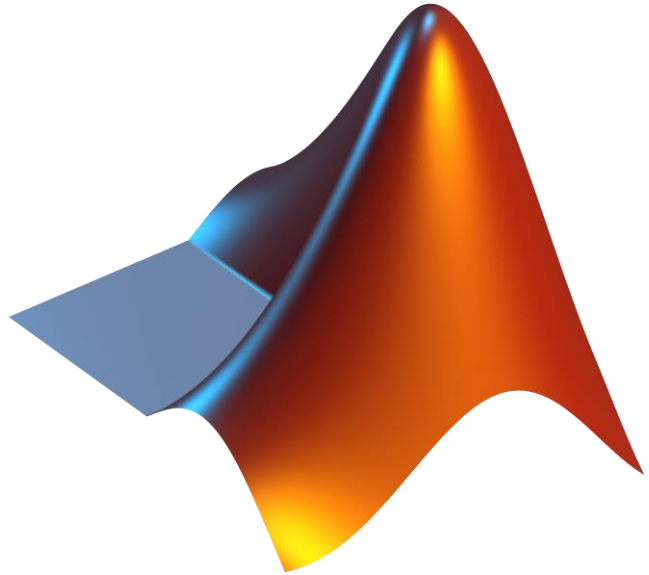
- ΔE* < 3 : non-distinguishable by human eye
- ΔE* > 5 : clearly distinguishable

1) https://en.wikipedia.org/wiki/HSL_and_HSV
2) https://zschuessler.github.io/DeltaE/learn/

Section: Image processing

Image processing

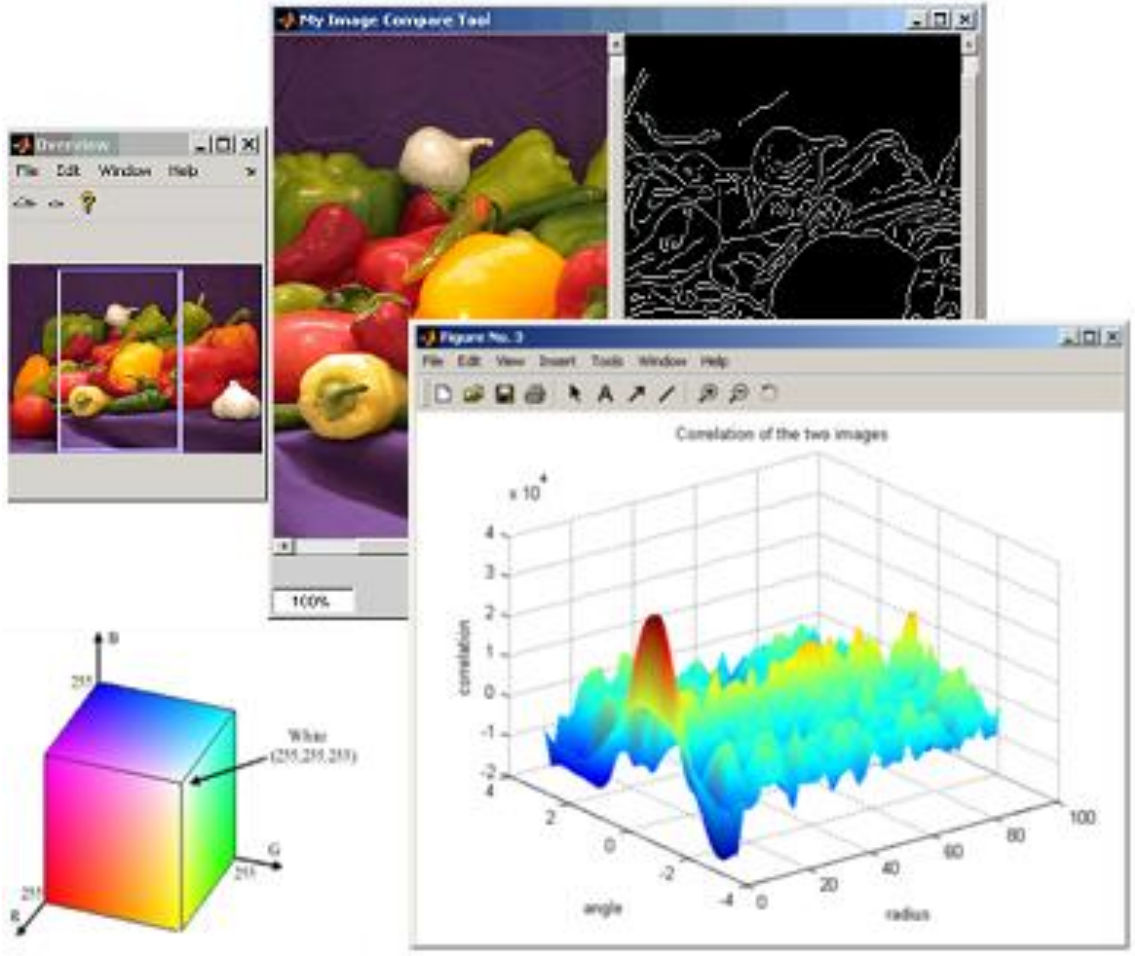
- How to take a photo and make it more useful
- There are programming libraries with functions for image processing
  - OpenCV2 -> Python, C++, Java
  - MATLAB embedded
- *All of the following examples can be done through the above libraries*



1) https://en.wikipedia.org/wiki/OpenCV
2) https://en.wikipedia.org/wiki/MATLAB

Image processing - Contents

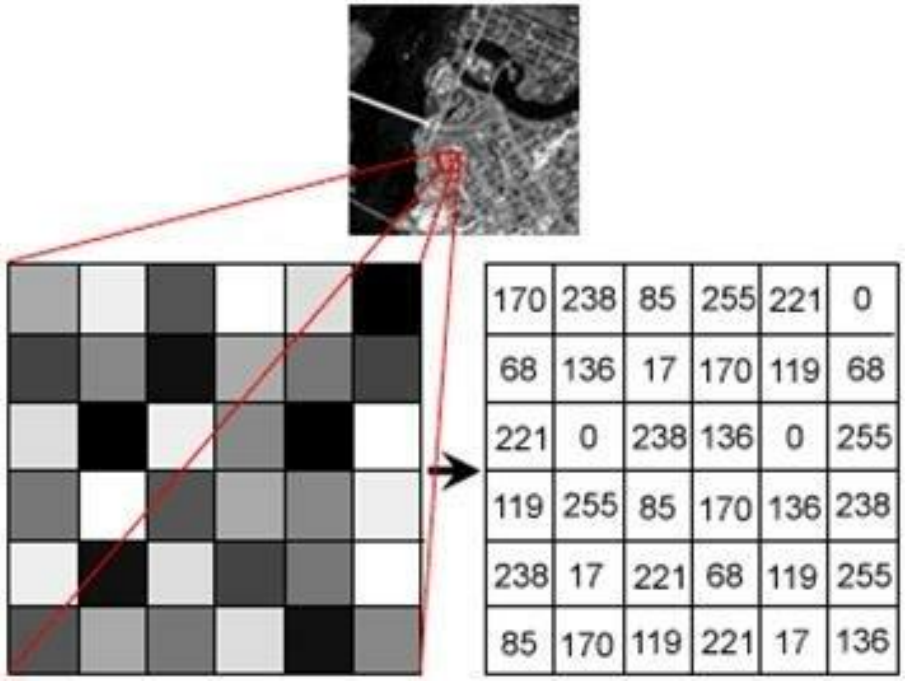
- Normalisation
- Histogram equalisation
- Image segmentation
- High / Low pass filtering
- Kernels
- Erosion / Dilation
- Edge detection

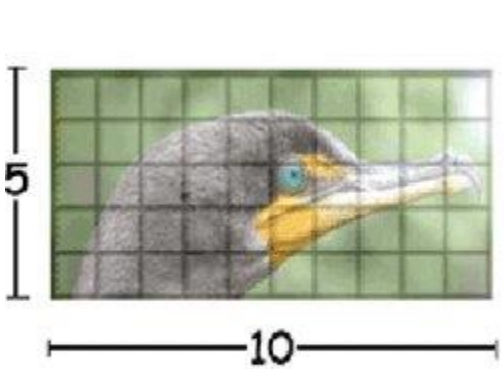


1) https://www.solutions4u-asia.com/emailc/digitalimageprocessing.html

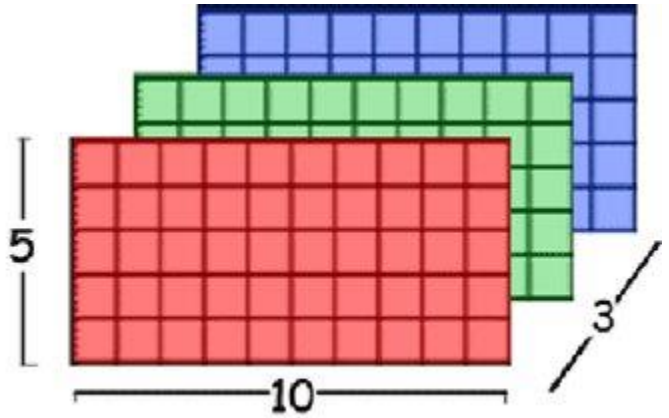
Image representation

- Images are just matrices
  - High value -> more intensity
- RGB: three stacked matrices
  - Width x height x 3
  - Can address as combination or individual “channels”
- Images just numbers
  - Can do matrix operations across whole image





Original Color Image



Matlab RGB Matrix

1) Barbosa B, Neves AJ, Soares SC, Dimas ID. Analysis of emotions from body postures based on digital imaging. SIGNAL. 2018 May 20:81.
2) https://www.researchgate.net/publication/282798184_The_project_is_based_on_emerging_field_Image_Processing_In_this_project_A_Graphical_User_Interface_has_been_designed_using_the_software_Labwindows_which_can_process_both_type_of_Real_time_Image_Proces/figures?lo=1

Image normalisation

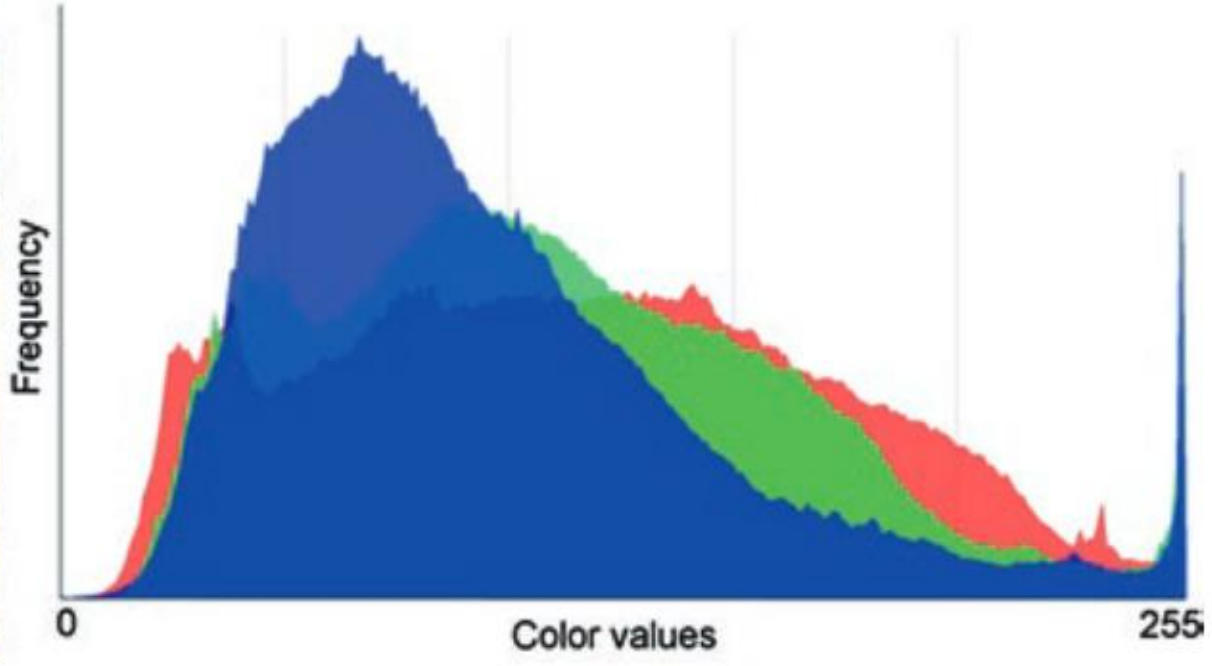
- Changes values of pixels to set range
  - Necessary as preprocessing
- E.g., greyscale image
  - Change 60-165 to 0-255
- Often normalised between 0-1 and 0-255
  - Formula often linear interpolation

None

1) https://ieeexplore.ieee.org/document/5411802

Histogram

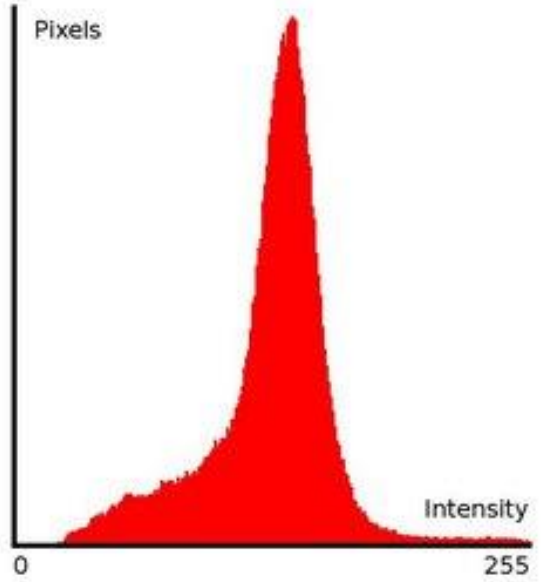
- Calculates abundance of colour in each channel

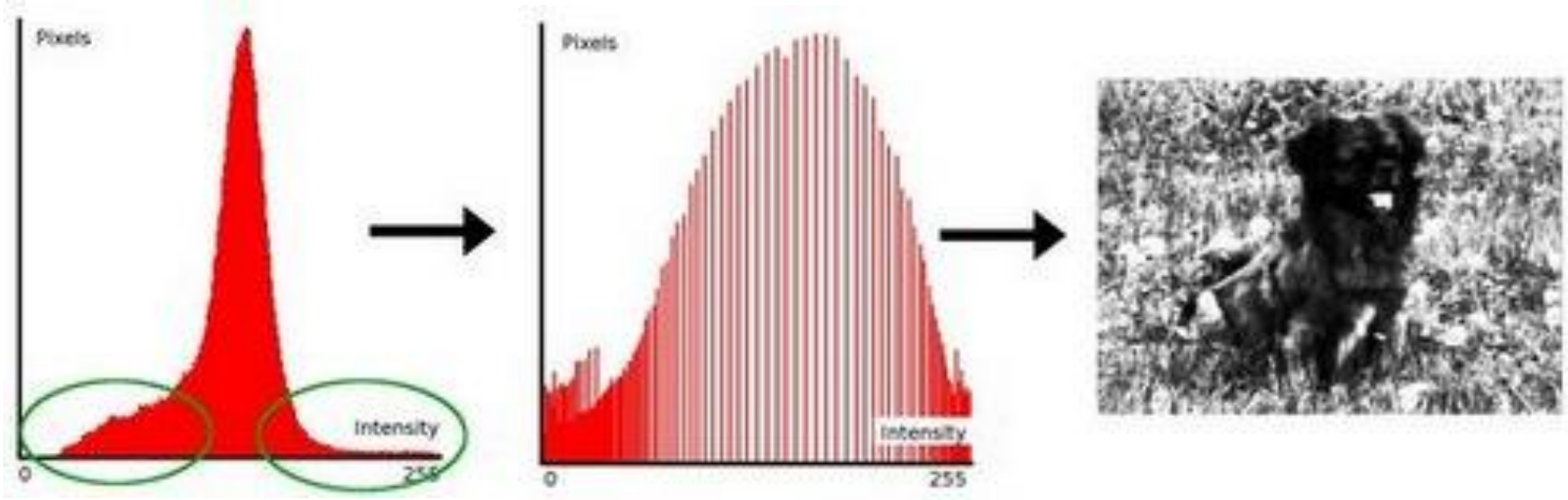


1) Fundamentals of agricultural and field robotics

Histogram Equalization

- Improves contrast
- Clip histogram
  - Interpolate pixels

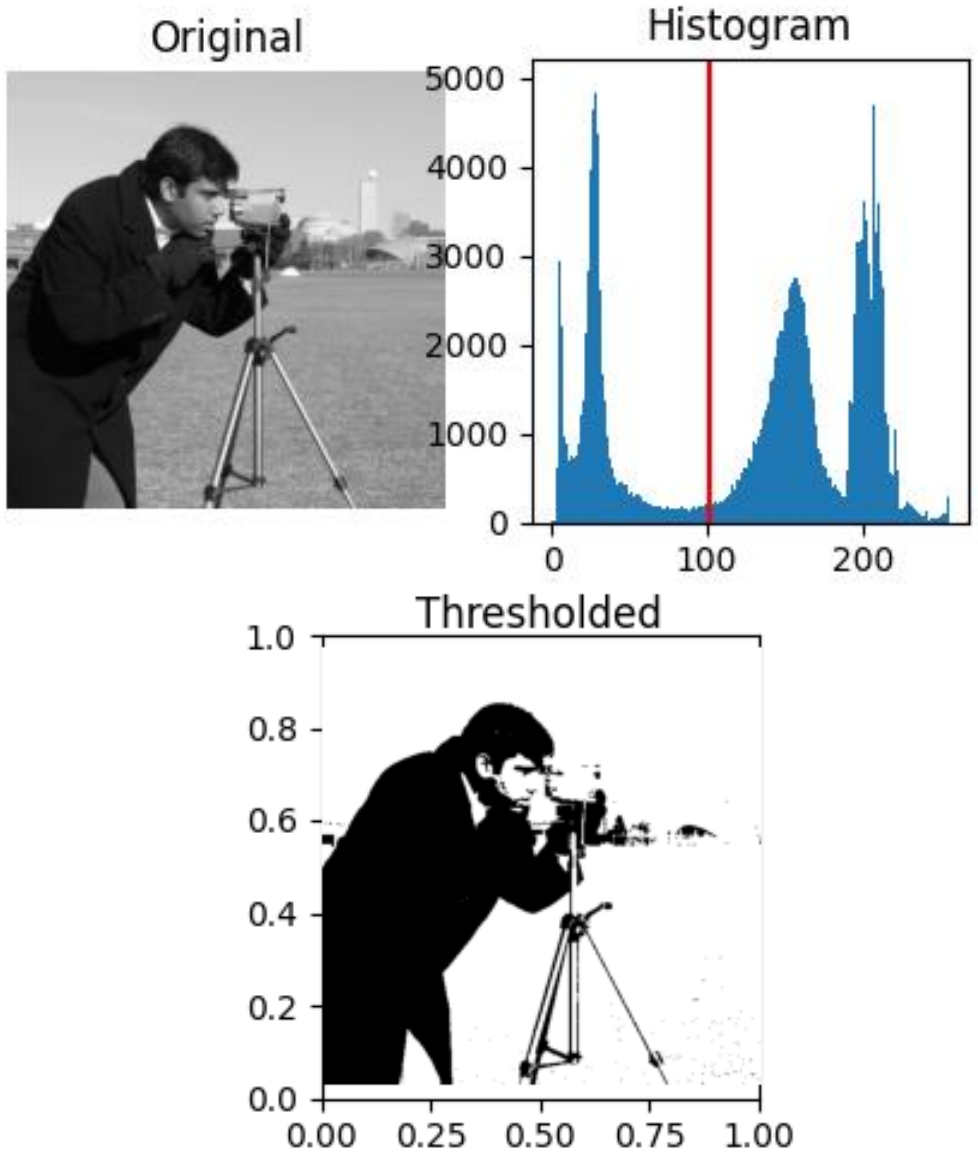




1) https://docs.opencv.org/4.x/d4/d1b/tutorial_histogram_equalization.html

Image segmentation

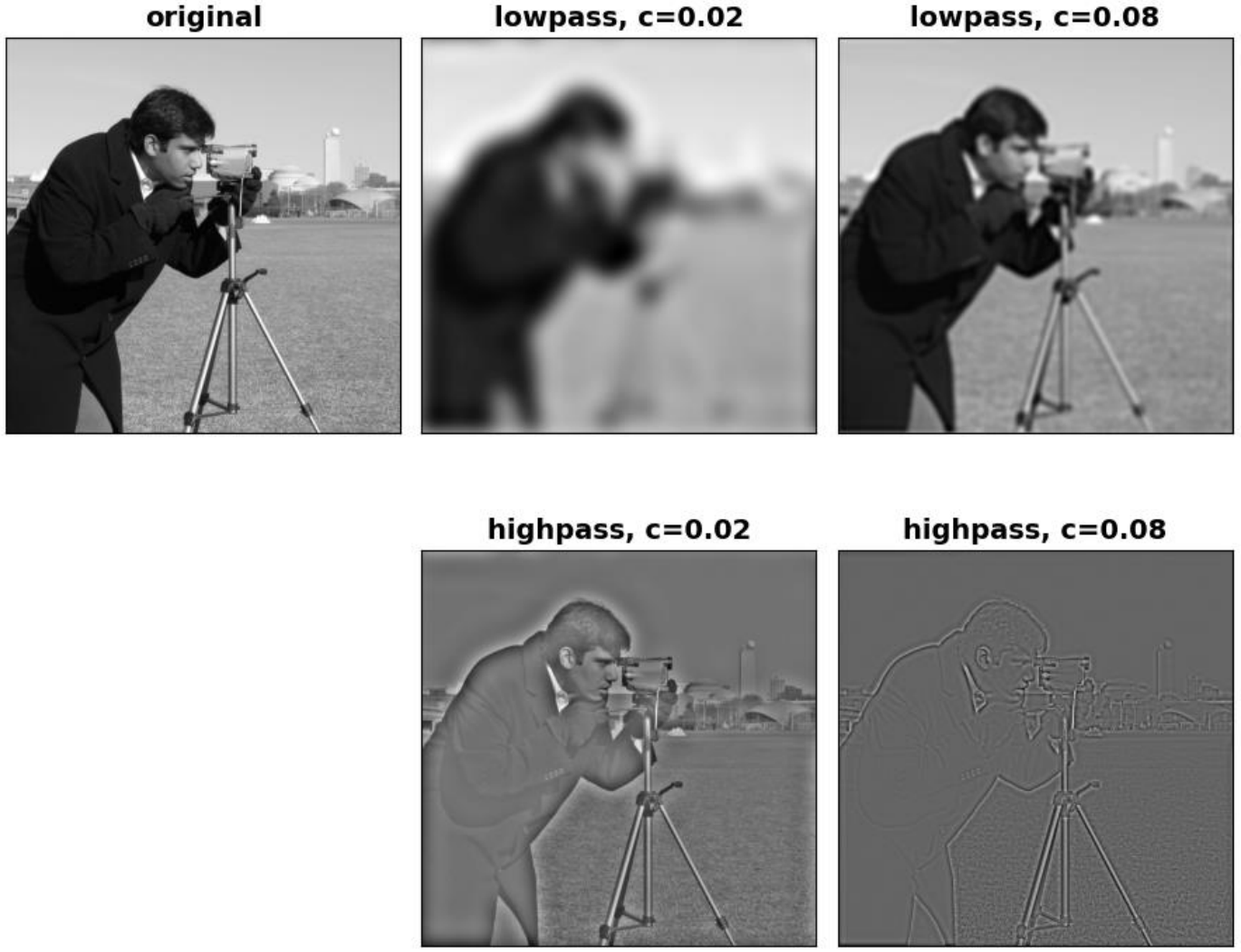
- Threshold segmentation
  - If pixel above an arbitrary “intensity” threshold
    - Classify as 1, otherwise 0
  - Binarisation
- Thresholds operate on
  - Greyscale image
  - Individual channels (e.g., red)
  - Combination of a few
- How to determine threshold?
  - Histogram analysis
  - Peak analysis, algorithms (Isodata thresholding)
  - Manual tuning



1) https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_thresholding.html

High and low pass filtering

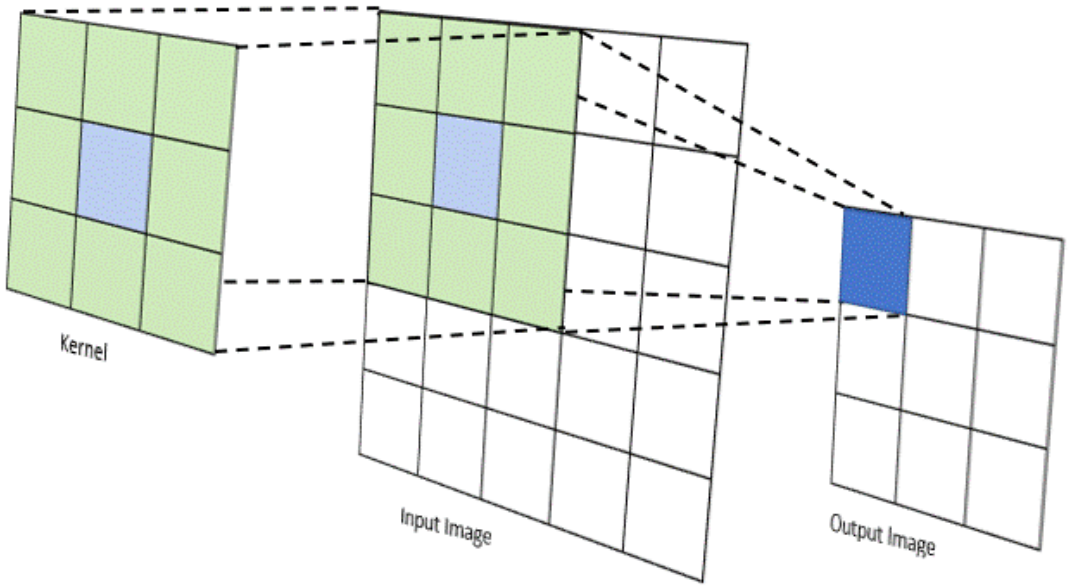
- Low pass filter
  - Blur
  - Removes noise
- High pass filter
  - Detail extraction
  - Edge detection
- Done via *kernels*



1) Source: https://scikit-image.org/docs/stable/api/skimage.filters.html

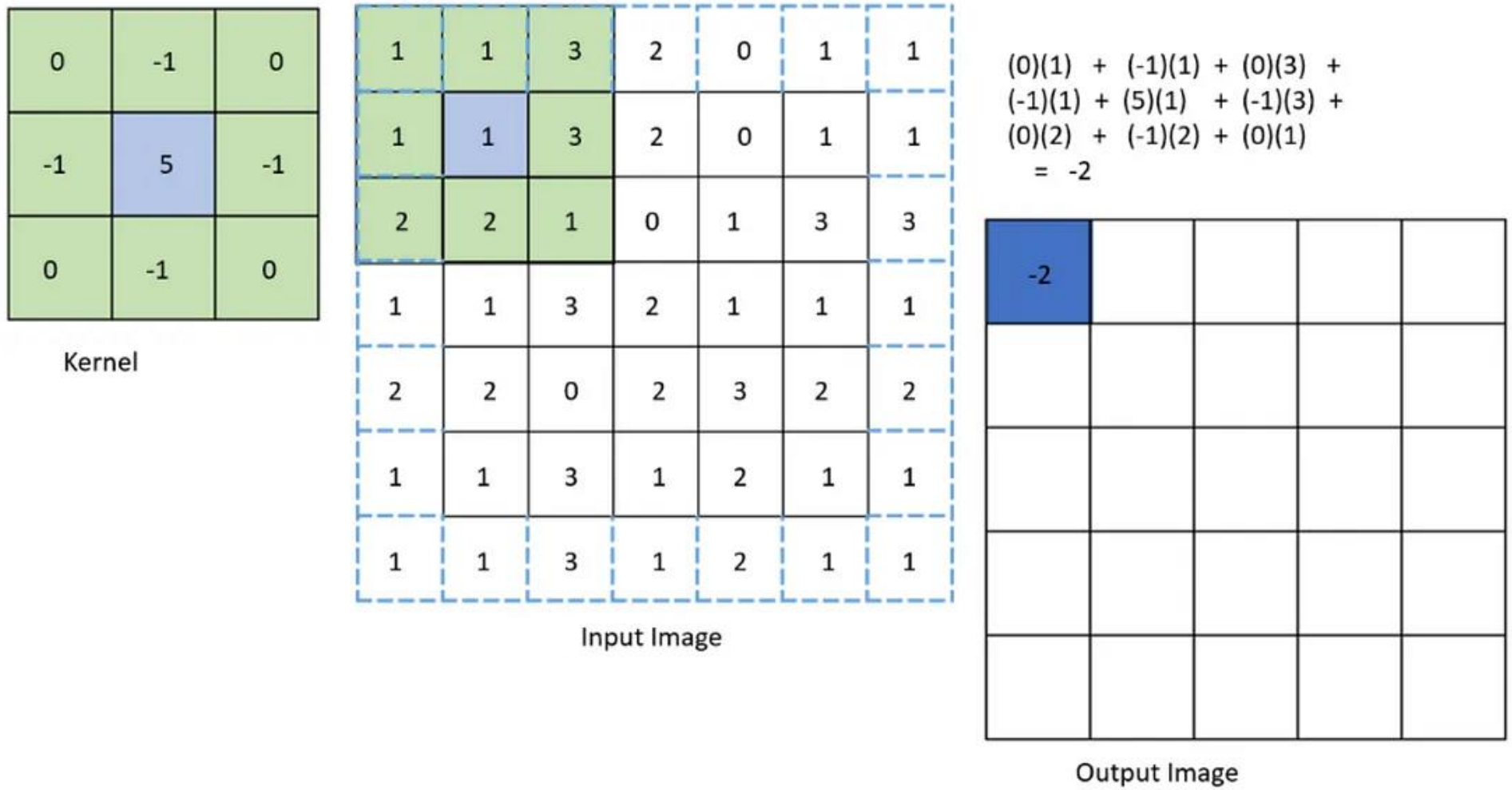
Image filtering - kernels

- Kernels
  - Small, fixed matrices
- Kernel scanned across image
  - Matrix operation results in a new pixel
  - Convolution
- *Choice of kernel does different actions to image*
  - Note: Reduces image size



1) Source: https://medium.com/@timothy_terati/image-convolution-filtering-a54dce7c786b

Image filtering – kernels example



1) Source: https://medium.com/@timothy_terati/image-convolution-filtering-a54dce7c786b

Image filtering - kernels

Original

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

Sharpen

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Gaussian Blur

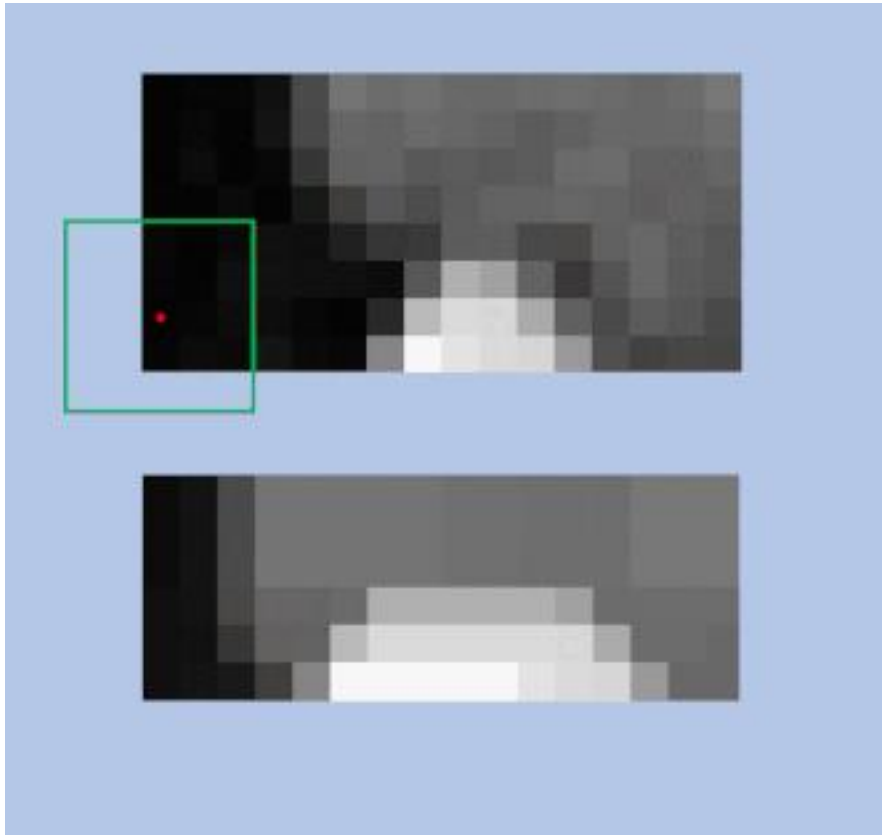
$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

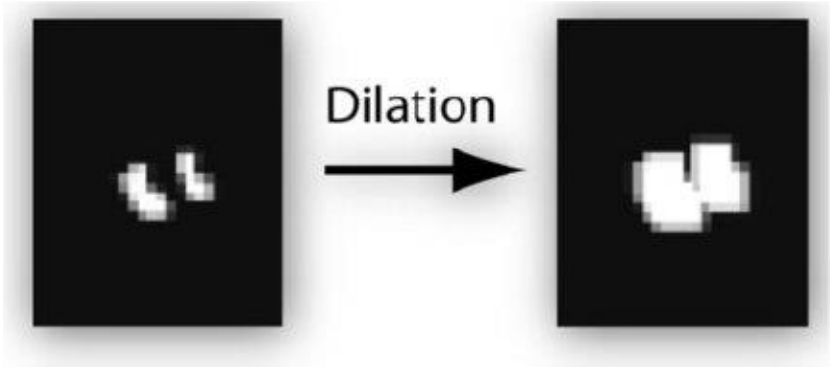
Diagonal edge detection

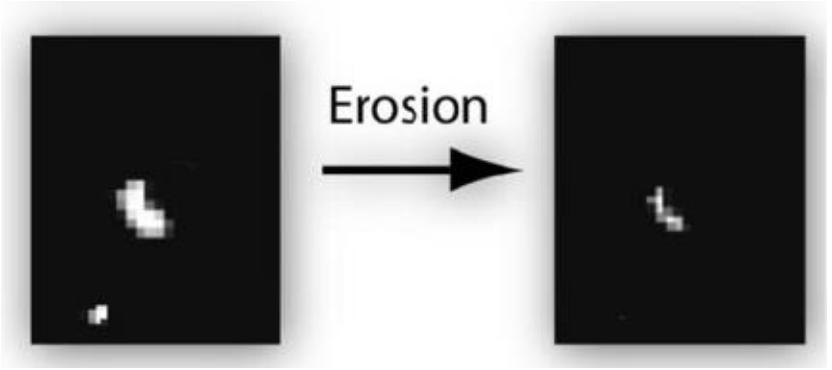
1) Source: https://medium.com/@timothy_terati/image-convolution-filtering-a54dce7c786b

Erosion / dilation

- Dilation:
  - Each pixel is maximum of all nearby pixels
- Erosion:
  - Each pixel is minimum of all nearby pixels



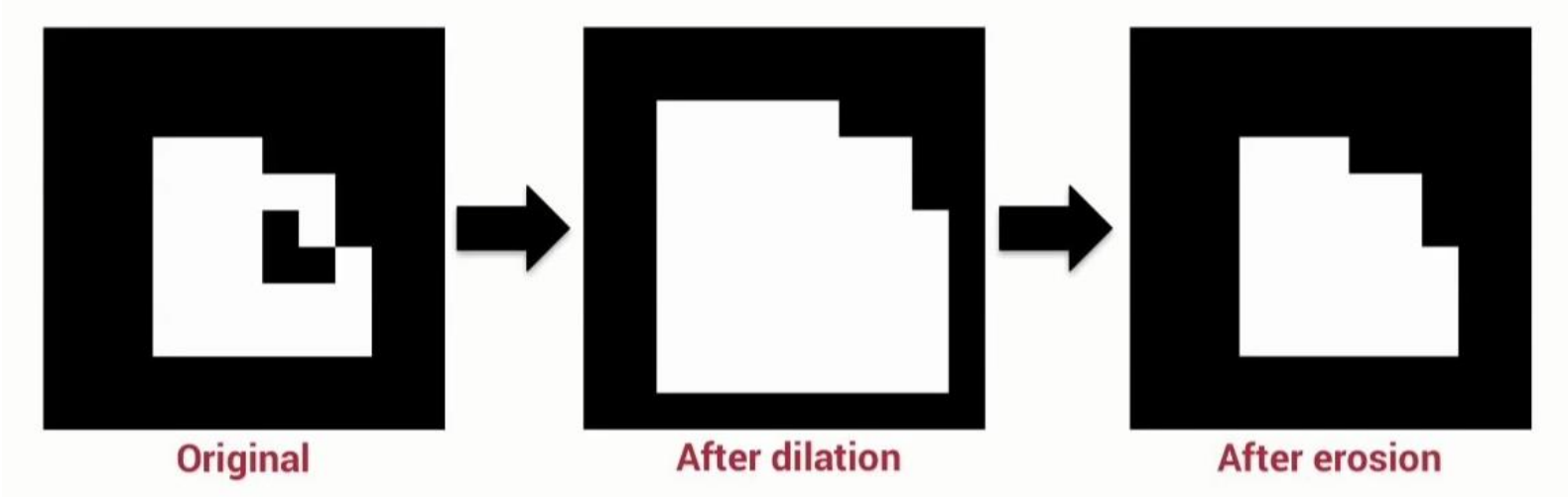




1) https://www.researchgate.net/publication/274454691_A_human-computer_collaborative_workflow_for_the_acquisition_and_analysis_of_terrestrial_insect_movement_in_behavioral_field_studies/figures?lo=1
2) https://en.wikipedia.org/wiki/Dilation_(morphology)

Dilation then erosion

- Dilation then erosion
  - Removes noise



1) Source: https://robotacademy.net.au/lesson/mathematical-morphology/

Section: Image processing algorithms

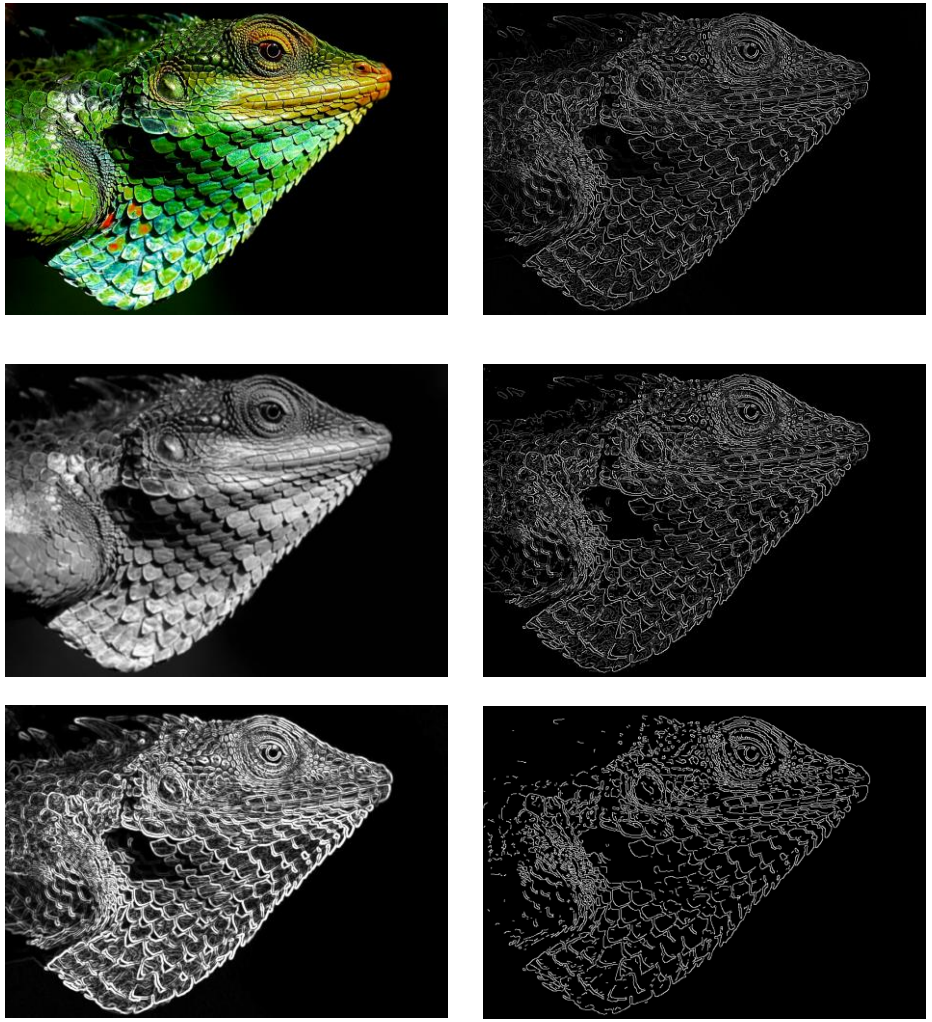
Canny edge detection

- Takes images and finds edges
  - Image into black + white edges

1) Source: https://en.wikipedia.org/wiki/Canny_edge_detector

Canny edge detection

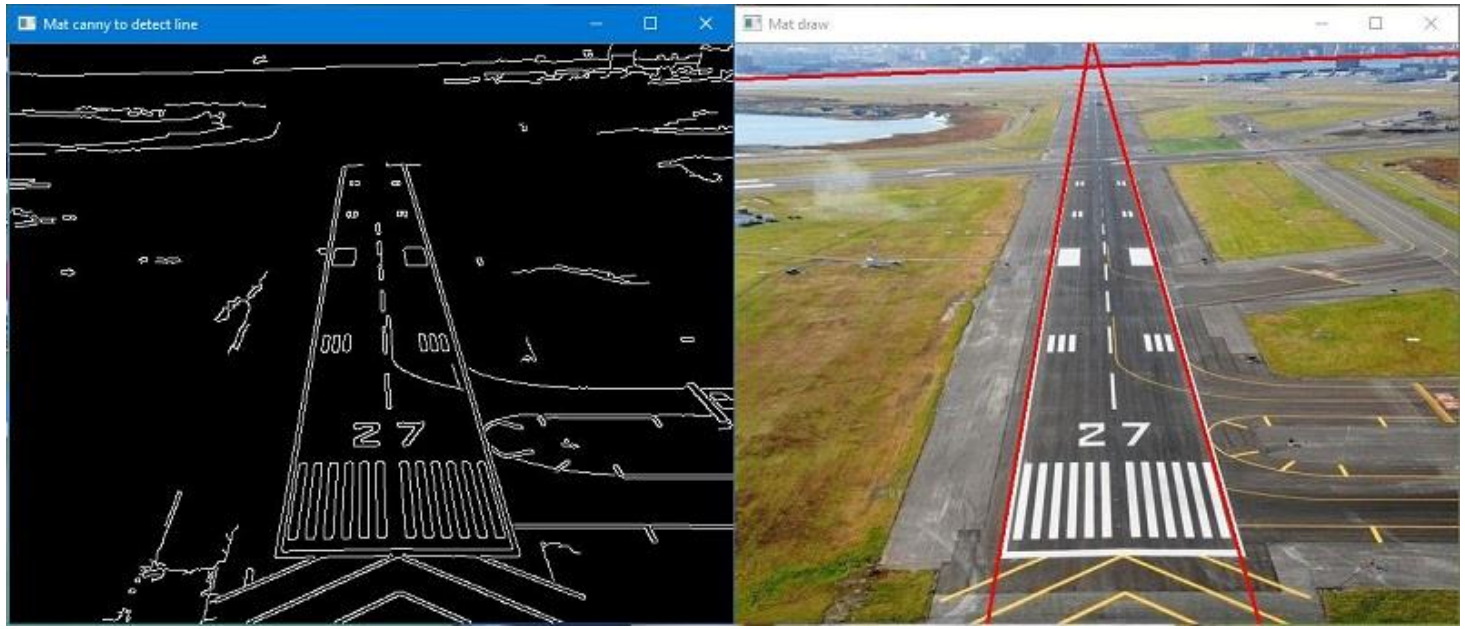
- Greyscale and Gaussian filter to remove noise
- Kernel to find intensity gradients
- Low threshold to eliminate extra edge detections
- Double the threshold to determine strong edge pixels
- Extend potential edges from strong pixels



1) https://en.wikipedia.org/wiki/Canny_edge_detector
2) https://thigiacmaytinh.com/tim-duong-thang-bang-ham-houghlines/

Hough Transform

- Used to calculate *lines* and *circles* in an image
  - Needs binary image



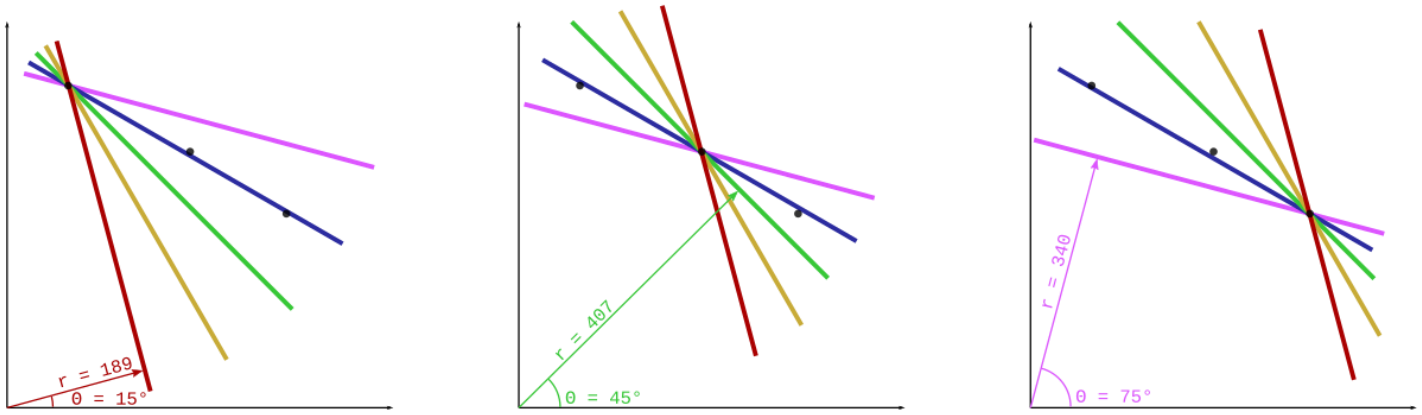
1) Source: https://thigiacmaytinh.com/tim-duong-thang-bang-ham-houghlines/

Hough Transform - Algorithm

- Pick metaparameters, (e.g., number of rotation degrees to test)
- For each **non-zero pixel**, try **all possible lines** (rotations) through the point
  - Track (accumulate) how many times a line appears
- Lines with most appearances are most likely

Example: three non-zero pixels

Metaparameters:
θ = 15 degrees
θmin = 15
θmax = 75



θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4

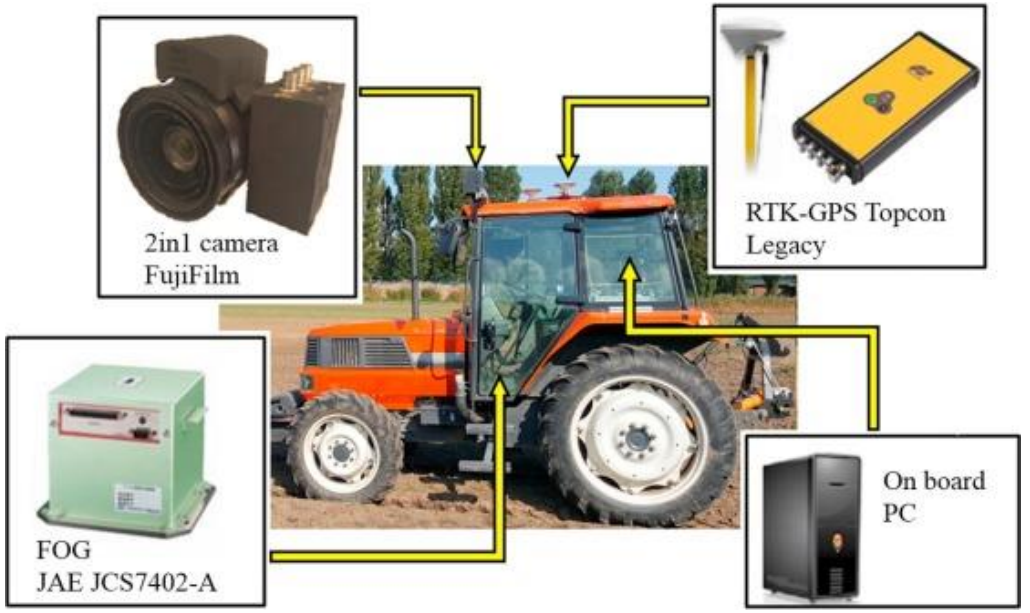
θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3

θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

1) Source: https://en.wikipedia.org/wiki/Hough_transform

Case study! Crop row detection

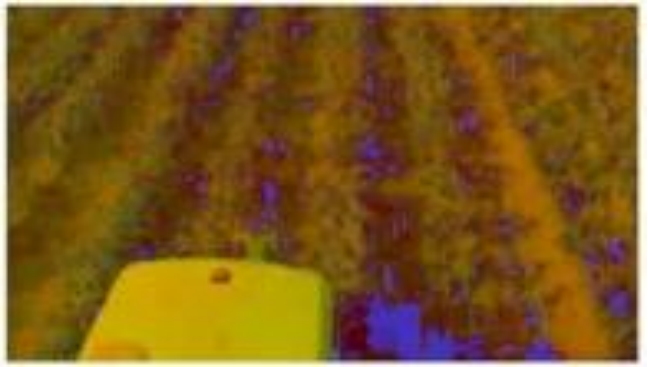
- How to detect crop rows?



1) Ospina R, Noguchi N. **Simultaneous mapping and crop row detection by fusing data from wide angle and telephoto images**. Computers and Electronics in Agriculture. 2019 Jul 1;162:602-12.

Case study! Crop row detection

- Proposed algorithm
  - Step 1: Convert to HSV

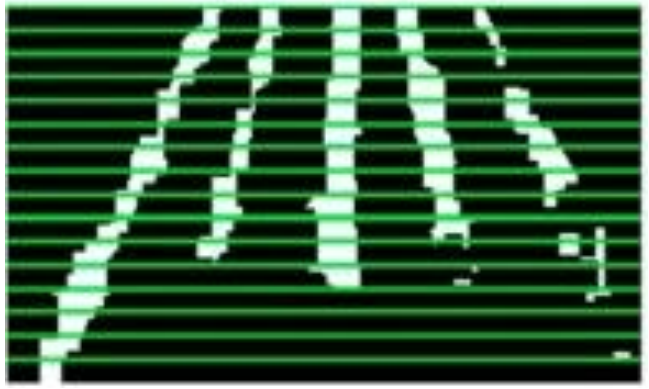


Crop row detection

- Proposed algorithm
  - Step 1: Convert to HSV
  - Step 2: Threshold into binary

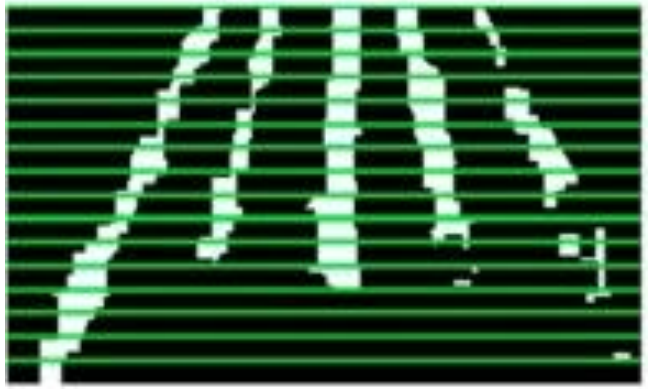
Crop row detection

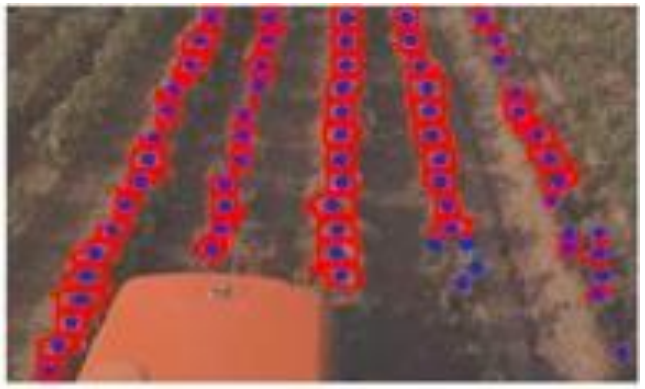
- Proposed algorithm
  - Step 1: Convert to HSV
  - Step 2: Threshold into binary
  - Step 3: Erode / dilute to group
    - Draw horizontal lines



Crop row detection

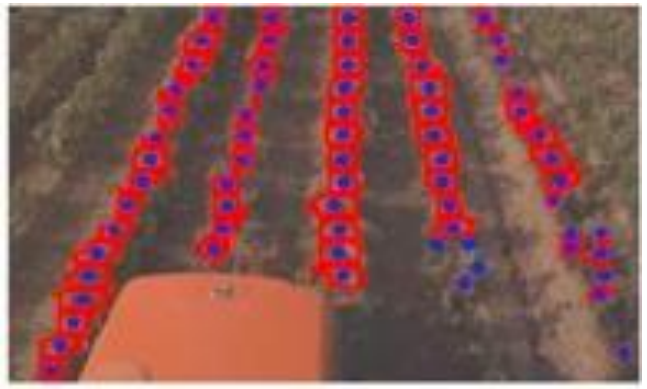
- Proposed algorithm
  - Step 1: Convert to HSV
  - Step 2: Threshold into binary
  - Step 3: Erode / dilute to group
    - Draw horizontal lines
  - Step 4: Detect intersections

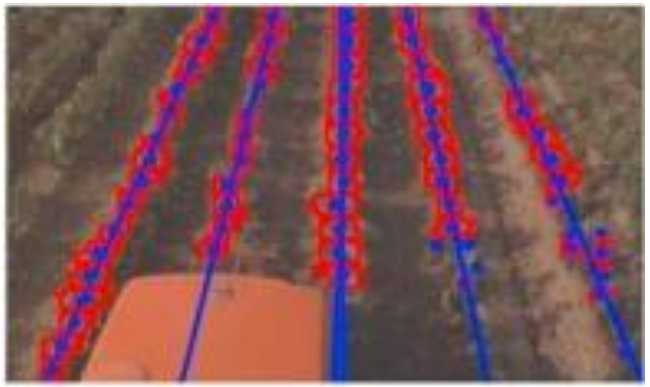




Crop row detection

- Proposed algorithm
  - Step 1: Convert to HSV
  - Step 2: Threshold into binary
  - Step 3: Erode / dilute to group
    - Draw horizontal lines
  - Step 4: Detect intersections
  - Step 5: Hough transform to detect rows





Crop row detection

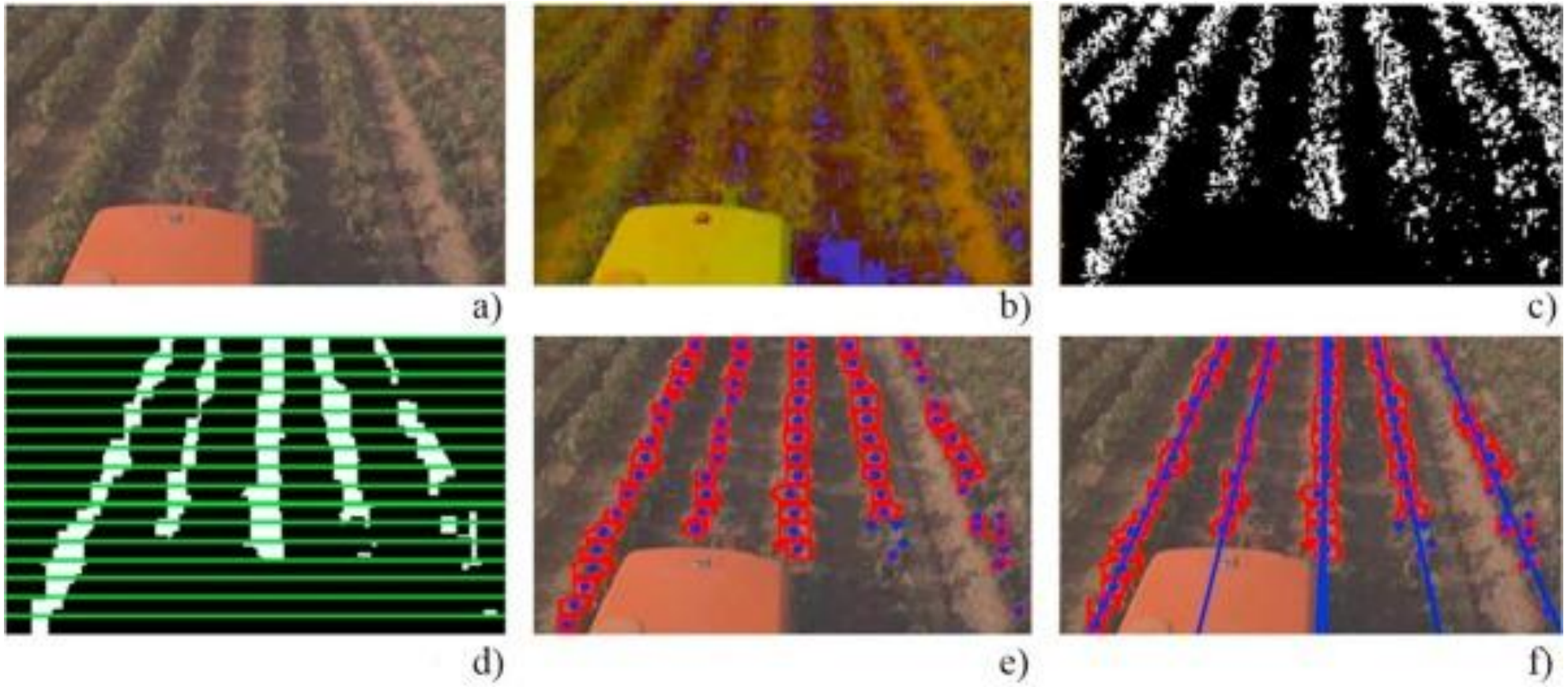
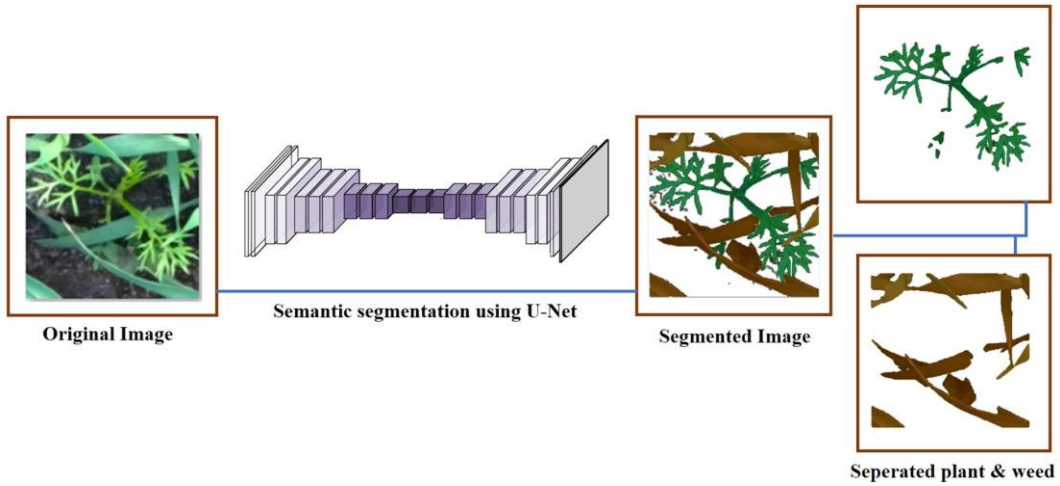


Image applications – Deep Learning

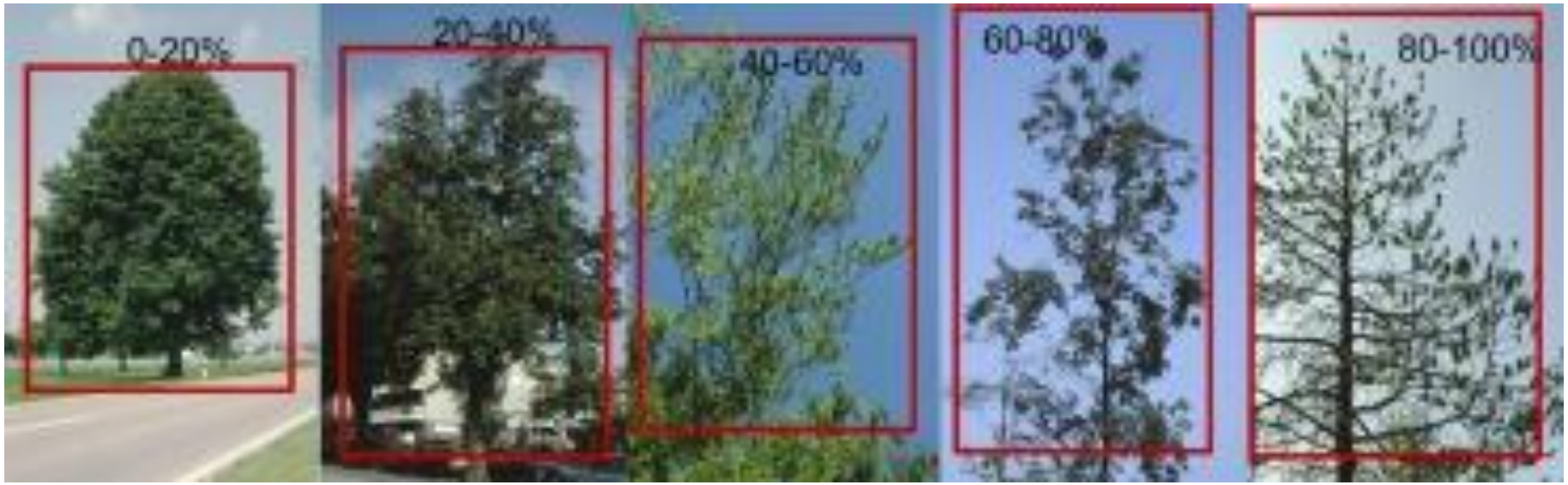
- Deep learning
  - New state-of-the-art in image processing
- Requires good data to train
  - Difficult to acquire
  - Often requires pre-processing
- Example:
  - Crown loss estimation



1) Source: https://www.sciencedirect.com/science/article/pii/S2214317321000731

Case study! Crown loss estimation

- How to detect crown loss in trees?

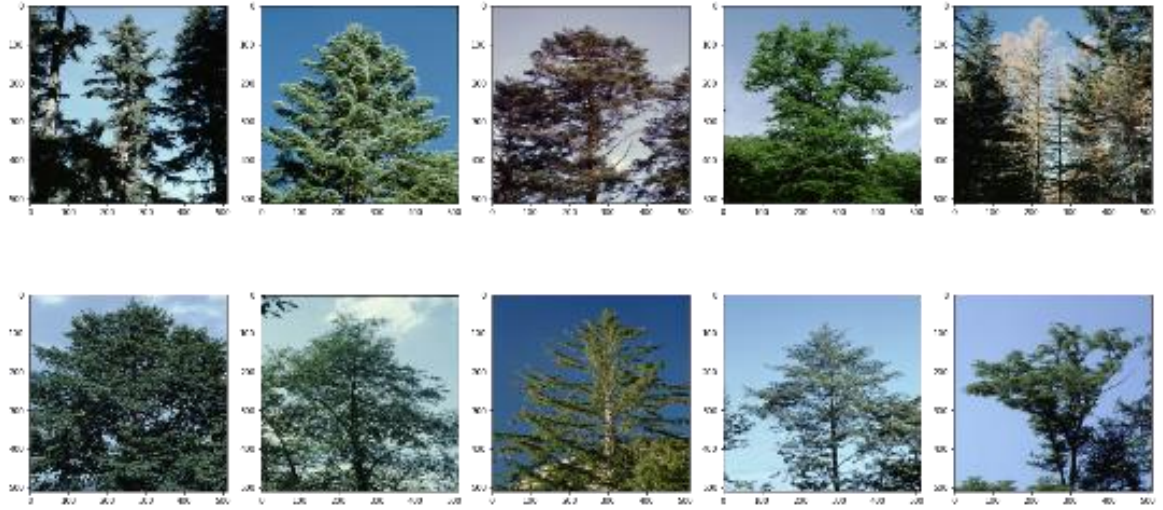


Ho B, Kocer BB, Kovac M. **Vision based crown loss estimation for individual trees with remote aerial robots**. ISPRS Journal of Photogrammetry and Remote Sensing. 2022 Jun 1;188:75-88.

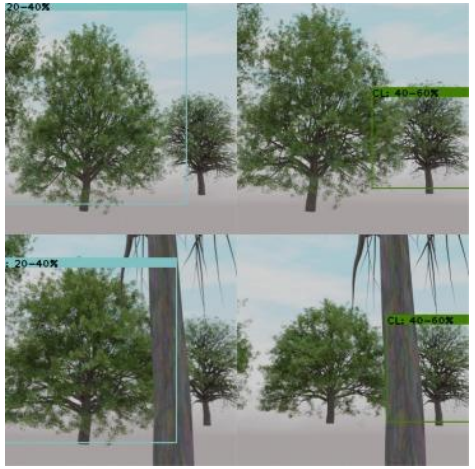
Case study! Crown loss estimation

- Not enough data in datasets
- Supplement training data with simulated trees from Blender

WSL tree dataset



Virtual trees



Obstructed environments

Case study! Crown loss estimation

Learning goals revisited

- Learn different colour models and how they are applied
  - RGB, CMYK, HSV, CIELAB
- Identify the underlying principles for basic image manipulation techniques
  - Normalisation, histograms, kernels
- Understand the core concept of edge and line detection
  - Canny edge detection, Hough transform
- Discuss applications of computer images on agriculture
  - Citrus color index, Crop row detection, crown loss estimation