Object Transformation and Segmentation Applications

Dr. Tran Anh Tuan,
Faculty of Mathematics and Computer Science,
University of Science, HCMC

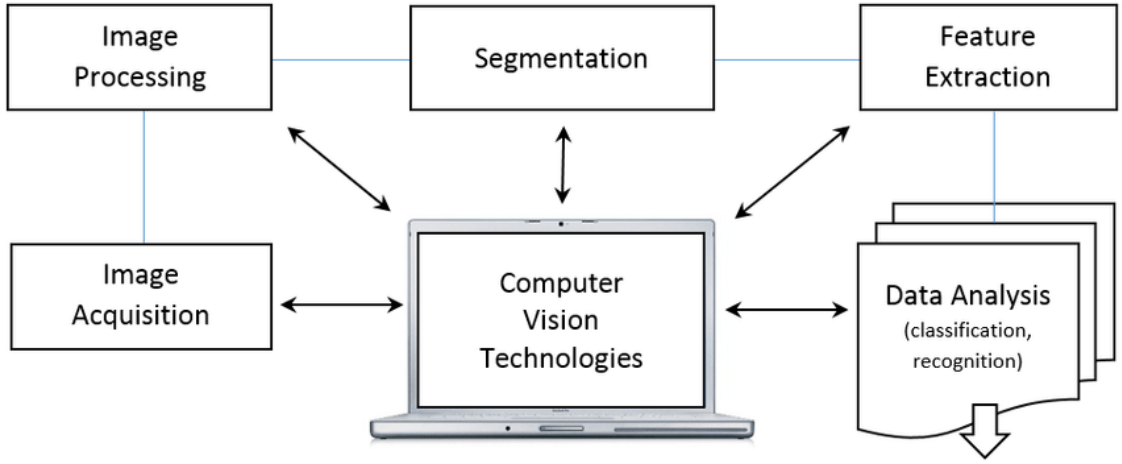
History of Computer Vision

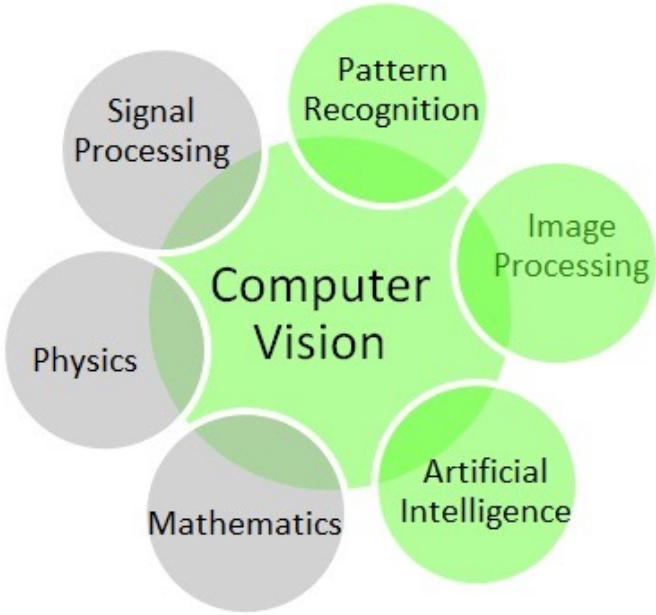
Marvin Minsky, MIT
Turing award, 1969

"In 1966, Minsky hired a first-year undergraduate student and assigned him a problem to solve over the summer:

connect a camera to a computer and get the machine to describe what it sees."

Crevier 1993, pg. 88





Syllabus

Week 1	Introduction to Object Segmentation
	- Overview of image segmentation - Importance and applications of object segmentation - Types of segmentation techniques
Week 2	Traditional Segmentation Techniques
	- Thresholding and region-based segmentation - Edge-based segmentation (e.g., Canny edge detection) Watershed segmentation - Contour-based segmentation
Week 3	Feature Extraction for Segmentation (English & Vietnamese)
	- Introduction to feature extraction - Color, texture, and shape features - Preprocessing techniques for feature extraction
Week 4	Evaluation Metrics for Segmentation
	- Metrics for assessing segmentation accuracy - Precision, recall, F1-score, and intersection over union (IoU) - Ground truth and benchmark datasets
Week 5	Introduction to Deep Learning (English & Vietnamese)
	- Overview of deep learning - Neural networks and deep learning frameworks - Training and inference processes

Syllabus

Week 6	Convolutional Neural Networks (CNNs)
	- Understanding CNN architecture - Convolutional layers, pooling layers, and fully connected layers - Image classification using CNNs
Week 7	Semantic Segmentation with CNNs
	- Introduction to semantic segmentation - Encoder-decoder architecture - Implementing semantic segmentation with CNNs
Week 8	Instance Segmentation with CNNs
	- Understanding instance segmentation - Mask R-CNN and similar architectures - Implementing instance segmentation with CNNs
Week 9	Transfer Learning and Pretrained Models (English & Vietnamese)
	- Transfer learning concept - Using pretrained models for segmentation - Fine-tuning and adapting pretrained models
Week 10	Data Augmentation and Image Preprocessing
	- Techniques for data augmentation - Image preprocessing for better segmentation results - Data balancing and handling imbalanced datasets

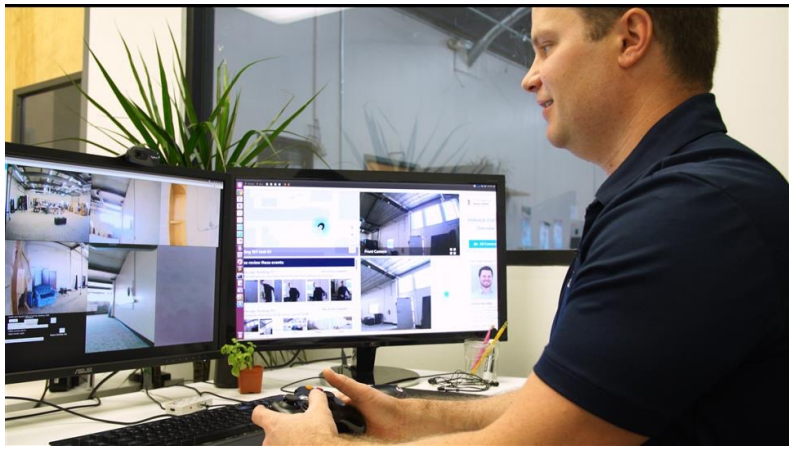
Syllabus

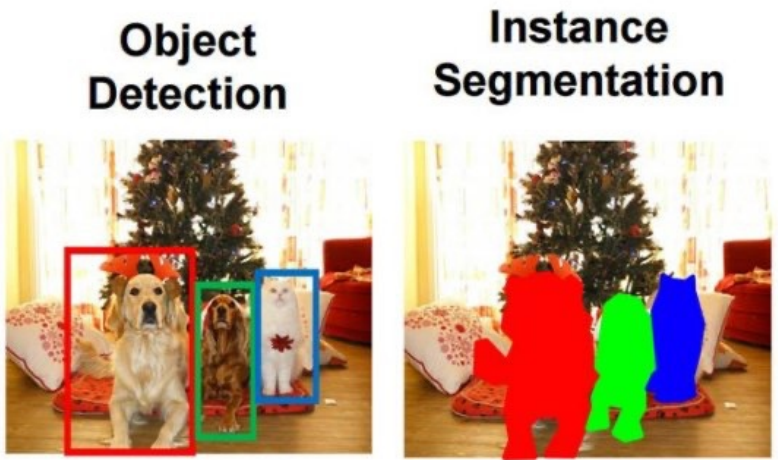
Week 11	Evaluation of Deep Learning Segmentation Models
	- Metrics for evaluating deep learning segmentation models - Comparing traditional and deep learning approaches - Handling challenges and limitations
Week 12	Combining Traditional and Deep Learning Approaches
	- Hybrid approaches to object segmentation - Integrating traditional and deep learning methods - Advantages and use cases
Week 13	Real-world Applications and Case Studies (English & Vietnamese)
	-Object segmentation in medical imaging - Autonomous driving and object detection - Object segmentation for computer vision tasks
Week 14	Future Trends and Research in Object Segmentation
	-Emerging trends in object segmentation - Research directions and challenges - Preparing for advanced studies or research projects

Special Session : Meets Manager and Senior Leader in Computer Vision Company

Introduction

- In computer vision, **image segmentation** is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects).
- The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.
- Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.



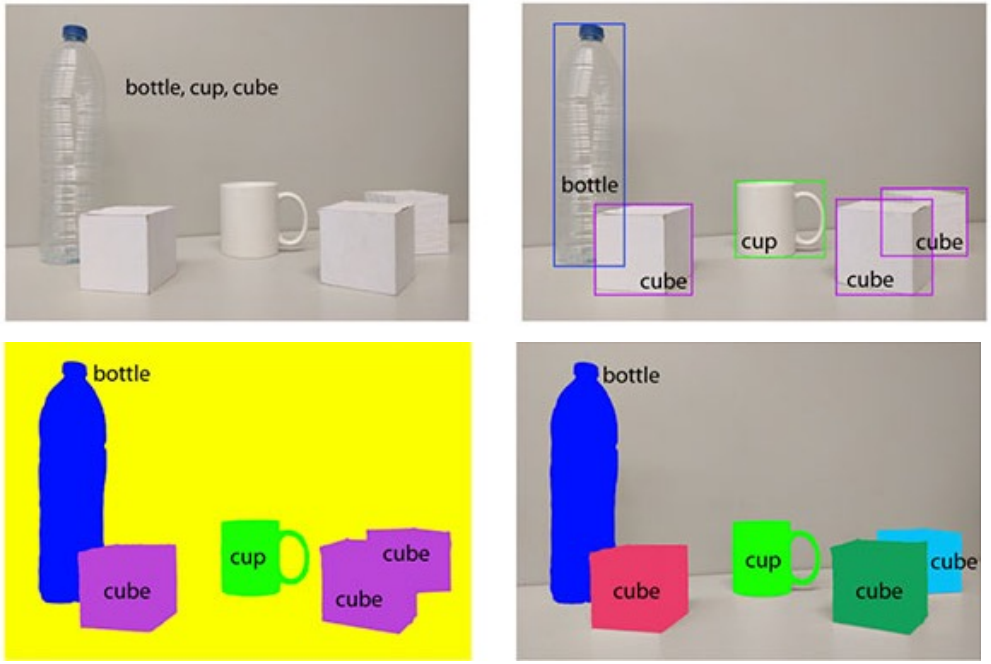


Introduction

In semantic segmentation, all objects of the same type are marked using one class label while in instance segmentation similar objects get their own separate labels.

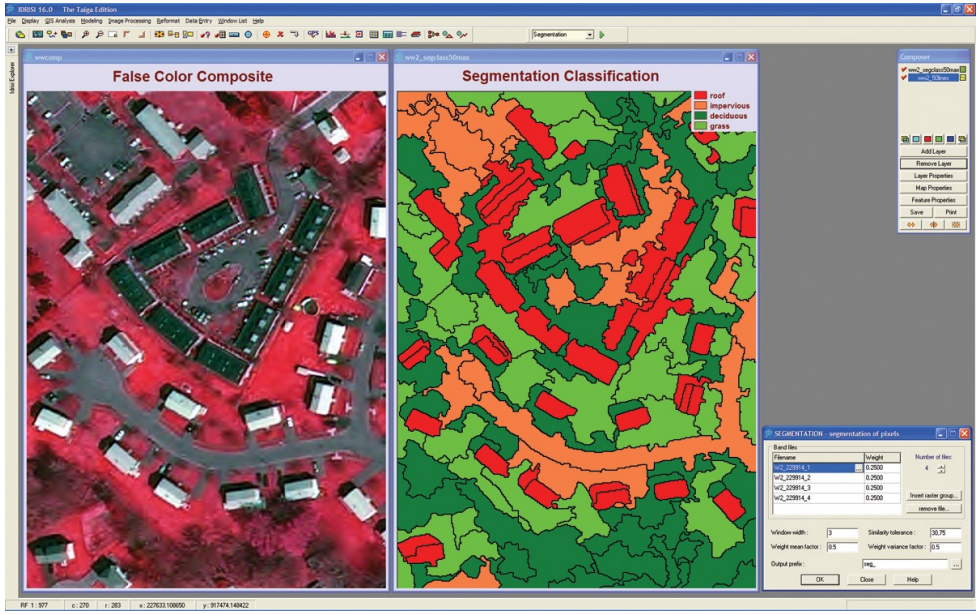
Question and Answer?

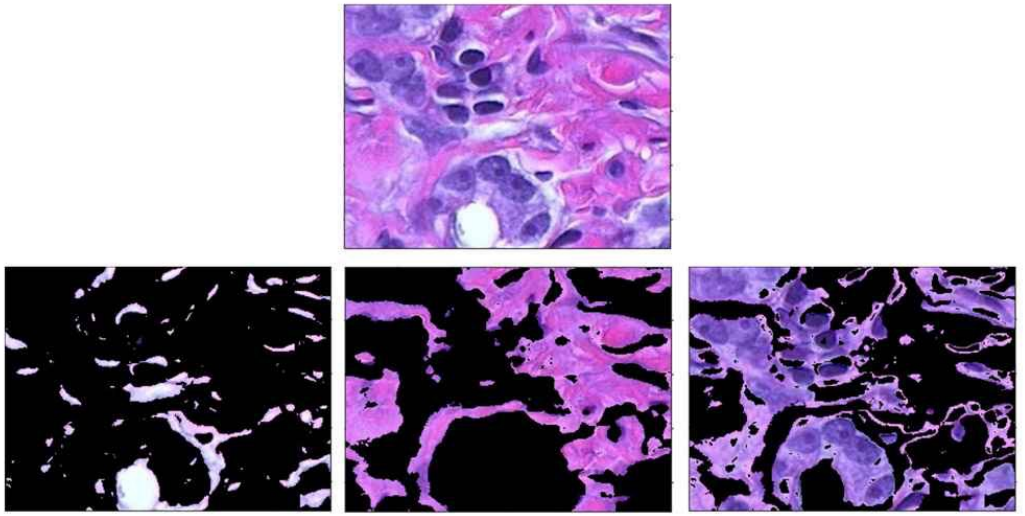
- What is the type of segmentation



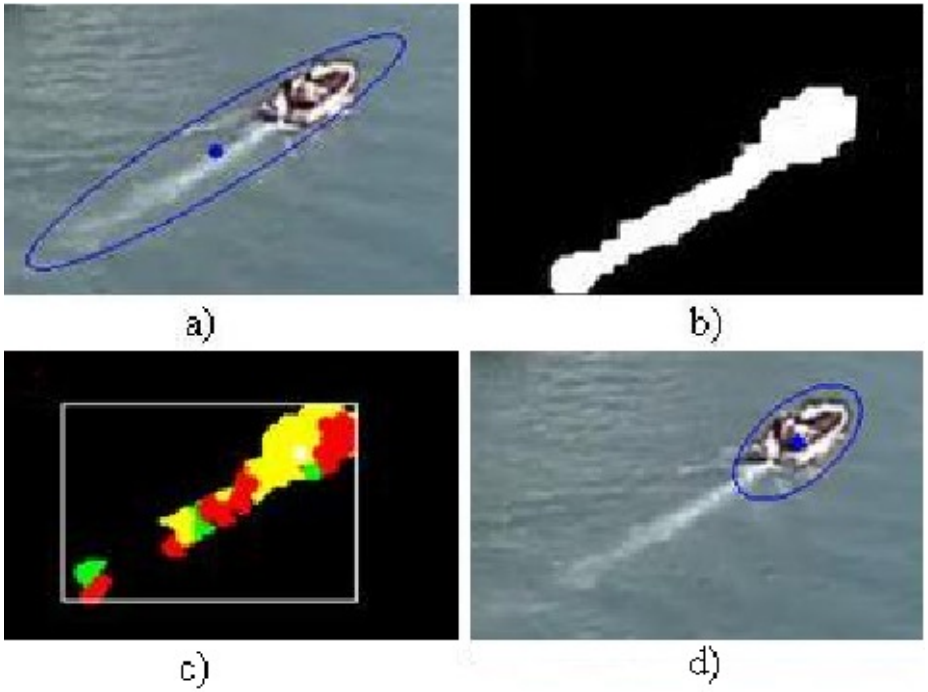
Question and Answer?

- What is the type of segmentation

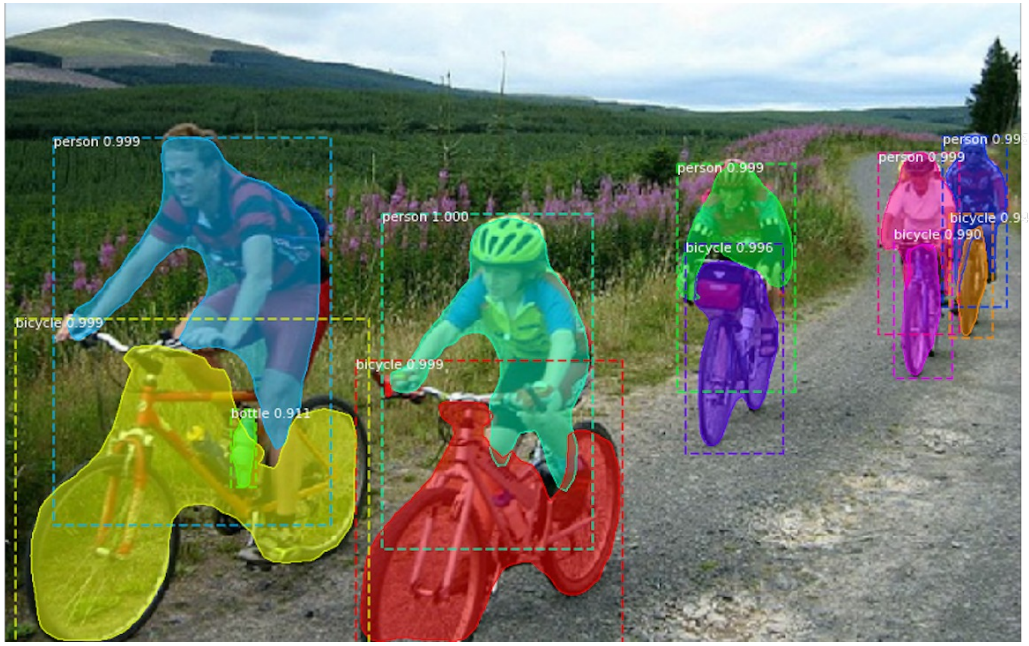




Challenges in Image Segmentation

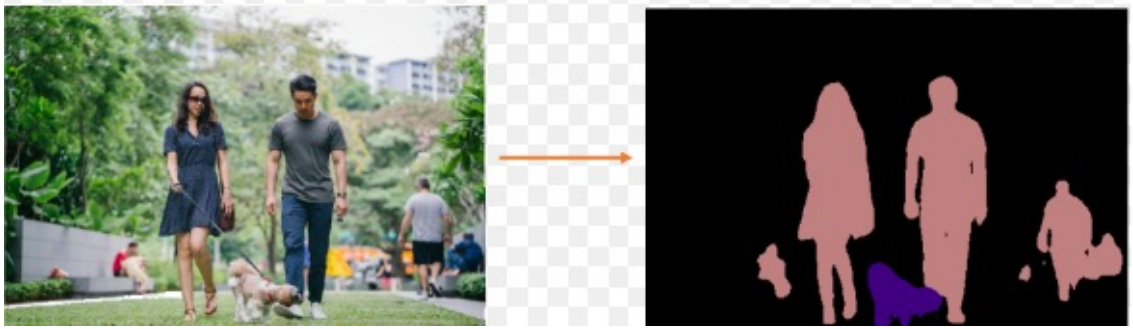


What happens ?

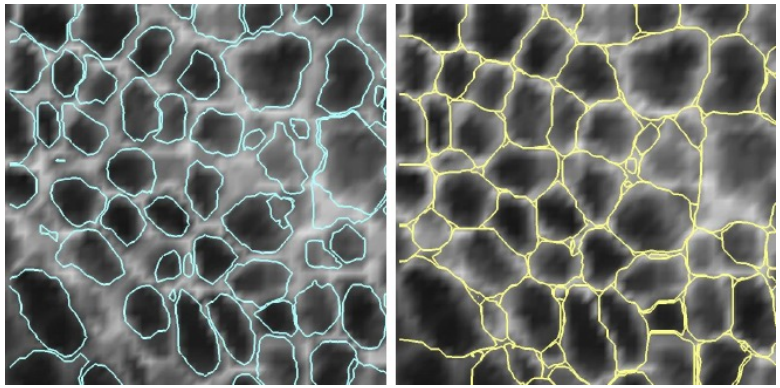


What happens ?

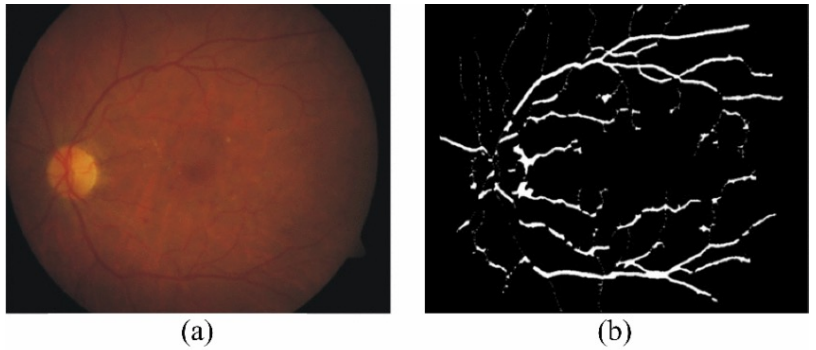
Challenges in Image Segmentation

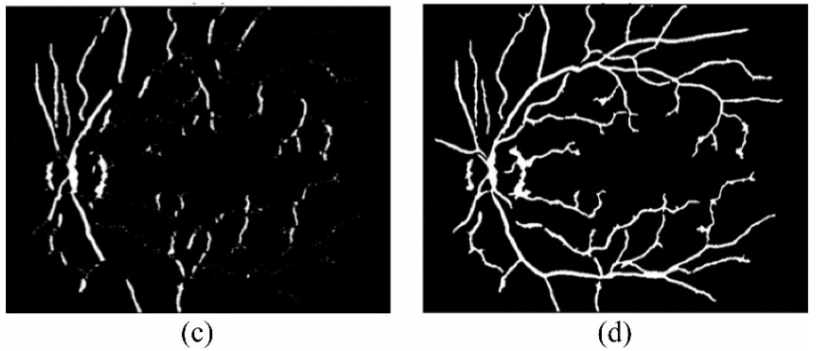


What happens ?



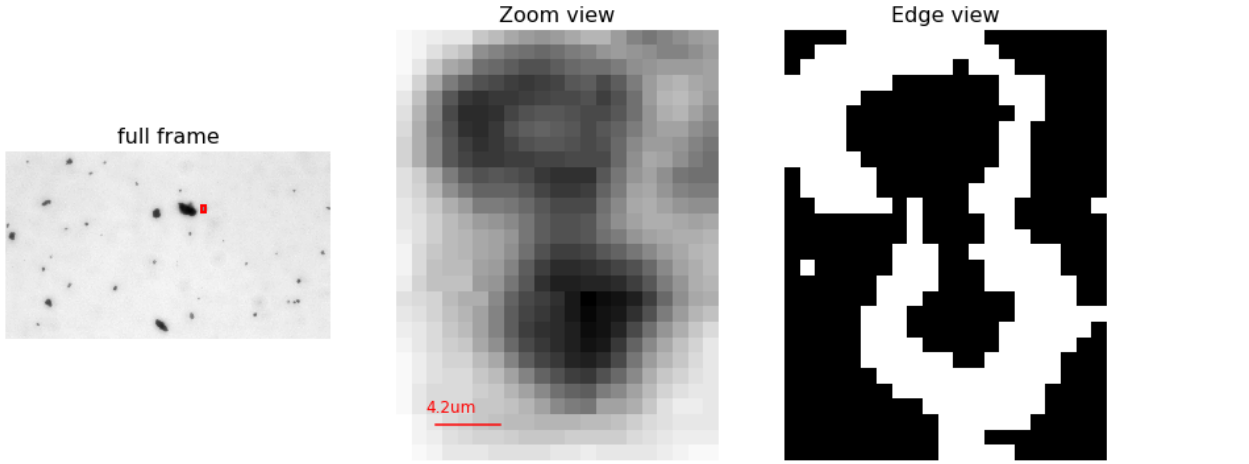
What happens ?



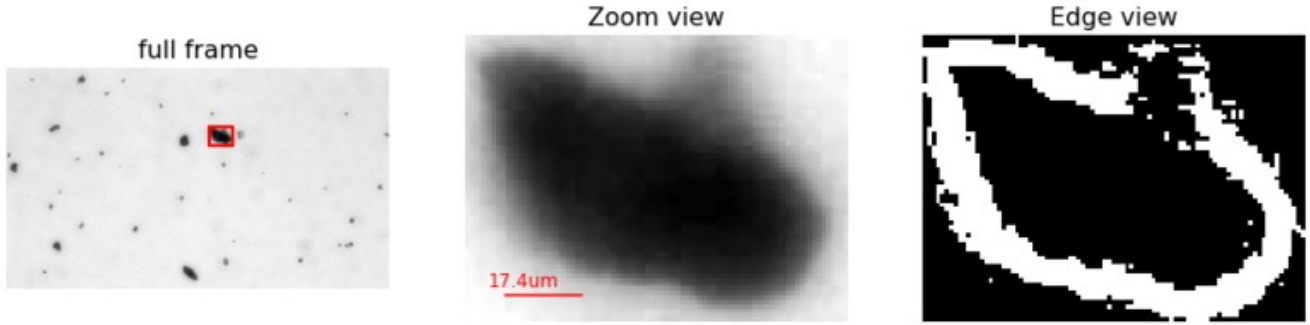


What happens ?

Challenges in Image Segmentation

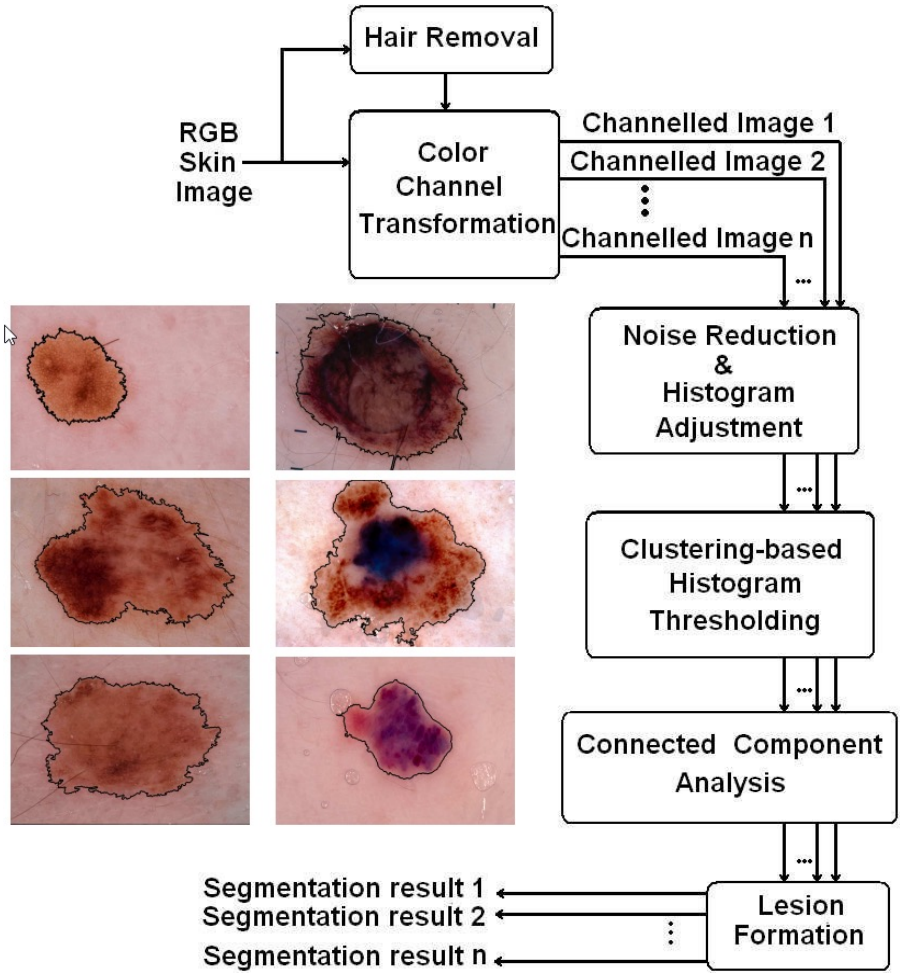


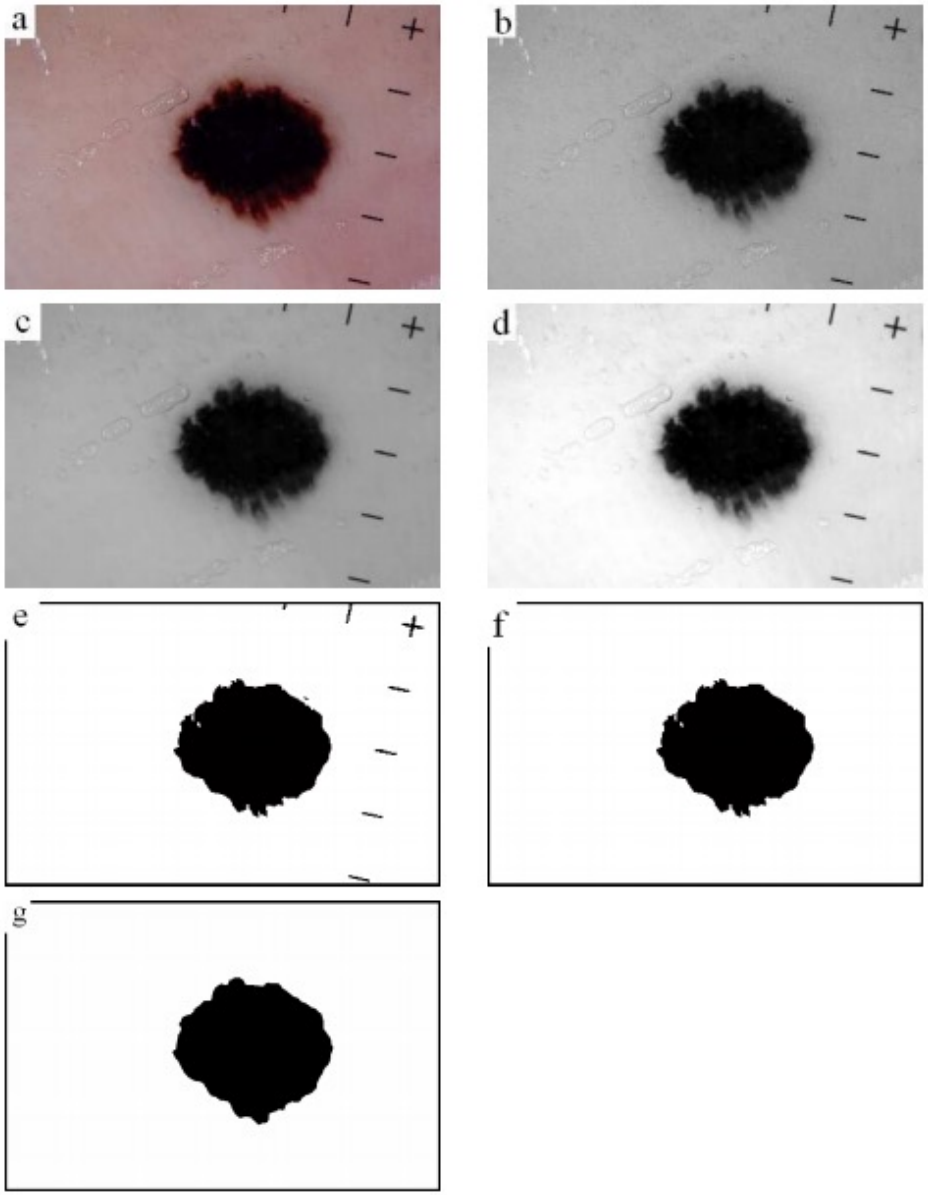
What happens ?



What happens ?

Image Segmentation Process





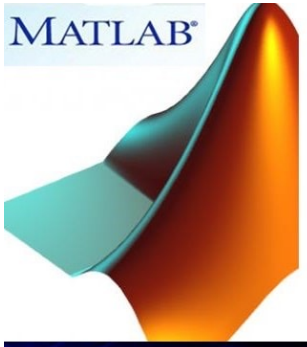
Computer Vision Framework

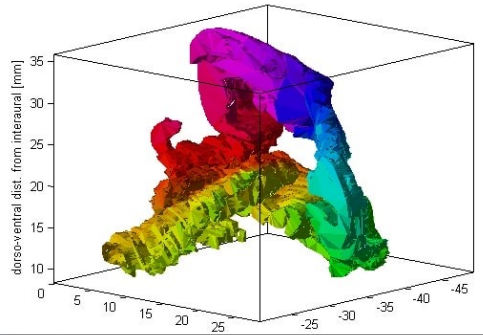
MATLAB

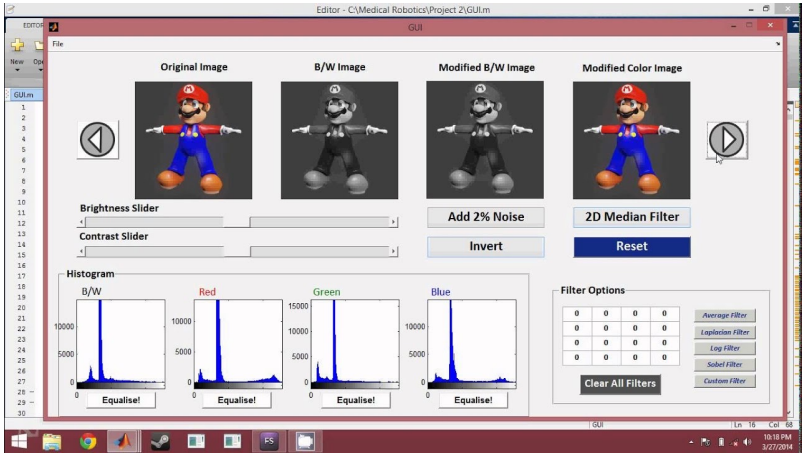
MATLAB, an abbreviation for 'matrix laboratory,' is a platform for solving mathematical and scientific problems. It is a proprietary programming language developed by MathWorks, allowing matrix manipulations, functions and data plotting, algorithm implementation, user interface creation and interfacing with programs written in programming languages like C, C++, Java and so on.

In MATLAB, the IPT is a collection of functions that extends the capability of the MATLAB numeric computing environment. It provides a comprehensive set of reference-standard algorithms and workflow applications for image processing, analysis, visualisation and algorithm development.

It can be used to perform image segmentation, image enhancement, noise reduction, geometric transformations, image registration and 3D image processing operations. Many of the IPT functions support C/C++ code generation for desktop prototyping and embedded vision system deployment.







Dr. Tran Anh Tuan, Faculty of Mathematics and Computer Science, University of Science, HCMC

This section loads some required libraries used in this notebook: **numpy**, **pandas**, **cv2**, **skimage**, **PIL**, **matplotlib**

- Numpy is an array manipulation library, used for linear algebra, Fourier transform, and random number capabilities.
- Pandas is a library for data manipulation and data analysis.
- CV2 is a library for computer vision tasks.
- Skimage is a library which supports image processing applications on python.
- Matplotlib is a library which generates figures and provides graphical user interface toolkit.


```
[ ]  import numpy as np
     import pandas as pd
     import cv2 as cv
     from google.colab.patches import cv2_imshow # for image display
     from skimage import io
     from PIL import Image
     import matplotlib.pylab as plt
```

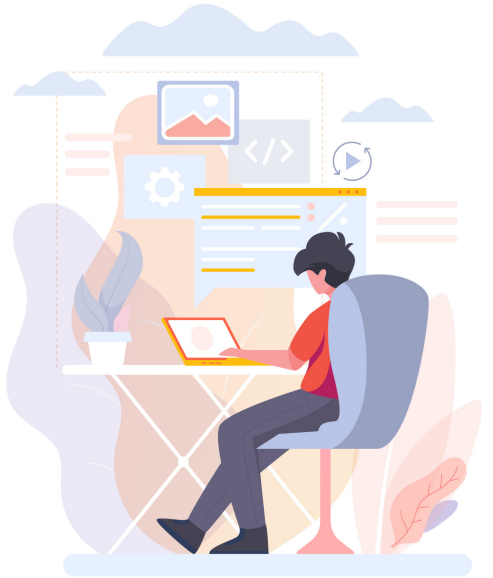

```
[ ]  # Create a list to store the urls of the images
     urls = ["https://iiif.lib.ncsu.edu/iiif/0052574/full/800,/0/default.jpg",
             "https://iiif.lib.ncsu.edu/iiif/0016007/full/800,/0/default.jpg",
            "https://placekitten.com/800/571"]
     # Read and display the image
     # loop over the image URLs, you could store several image urls in the list

     for url in urls:
       image = io.imread(url)
       image_2 = cv.cvtColor(image, cv.COLOR_BGR2RGB)
       final_frame = cv.hconcat((image, image_2))
       cv2_imshow(final_frame)
       print('\n')
```

Python

CV JOB Description

Role & Responsibilities

- Start with one of our core products, such as understanding video ads content, automatic generation of image/video ads, image/video analytics, etc.
- Develop ideas and create algorithms on image/video processing projects.
- Capable of starting a project from idea development to implementation.
- Contribute to the design and development of our products.
- Provide knowledge about technology trends advising about potential benefits and impacts.
- Introduce new techniques, technologies as needed.
- Collaborate well with your fellow teammates.
- Be a core contributor pioneering our new AI Products.



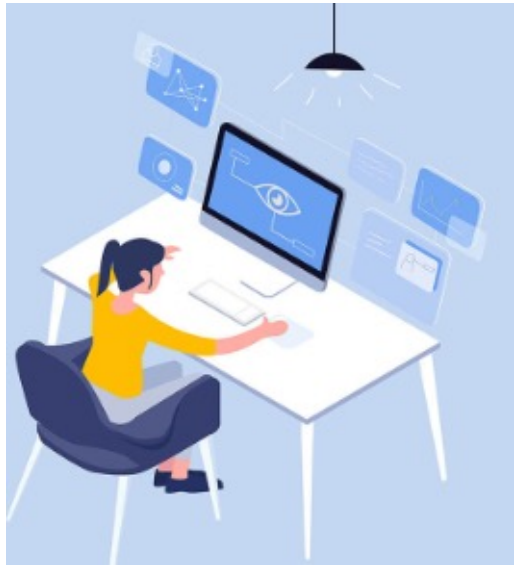
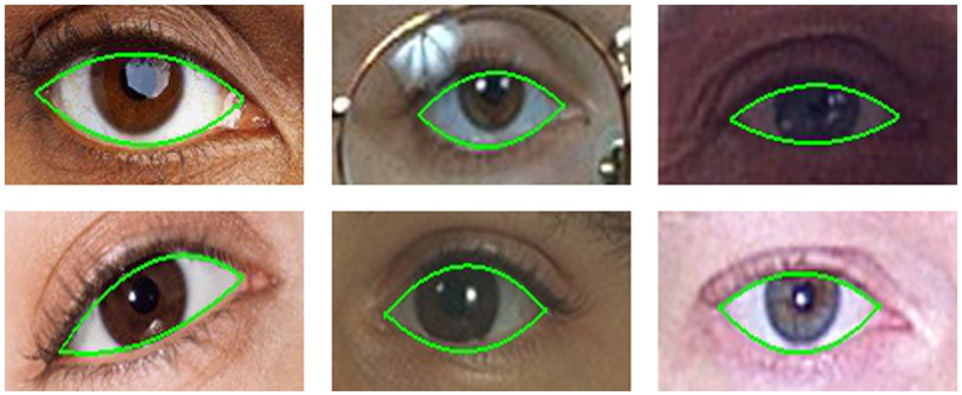
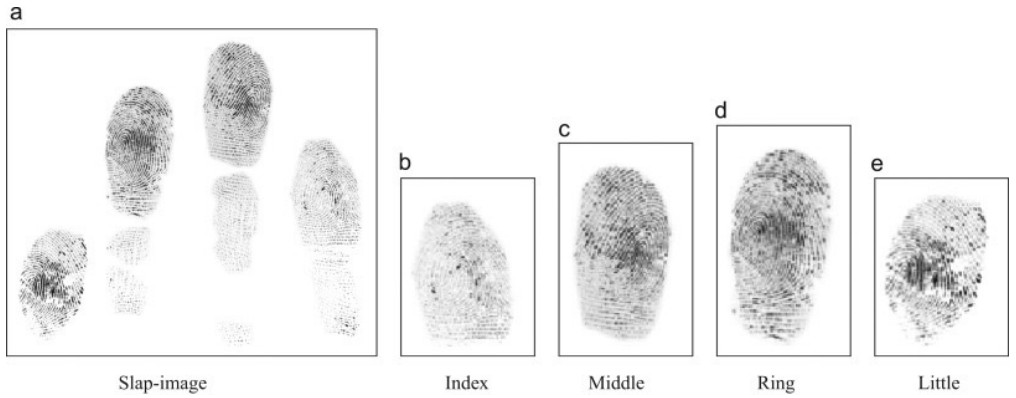


Image Segmentation Applications

- Biomedical Applications



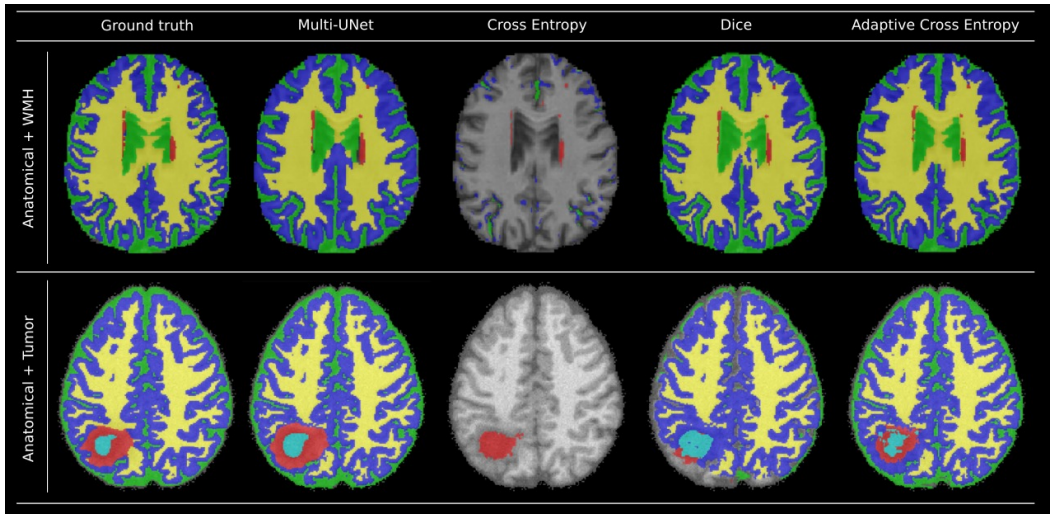
Eye Authentication



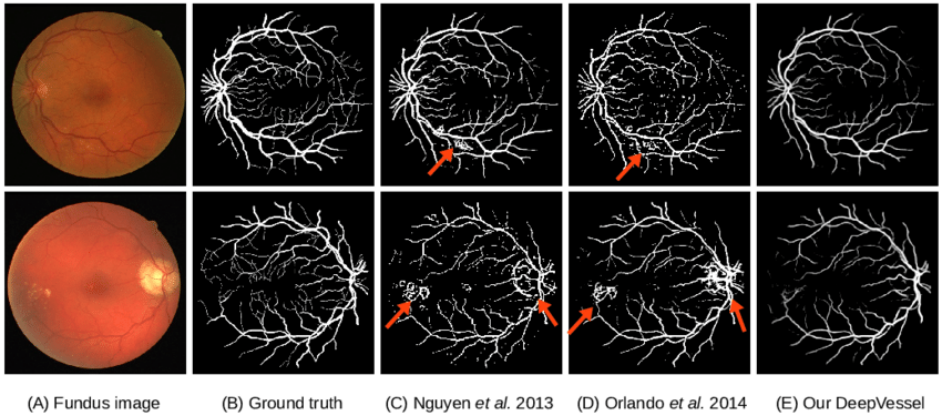
Fingerprint Authentication

Image Segmentation Applications

- Biomedical Applications



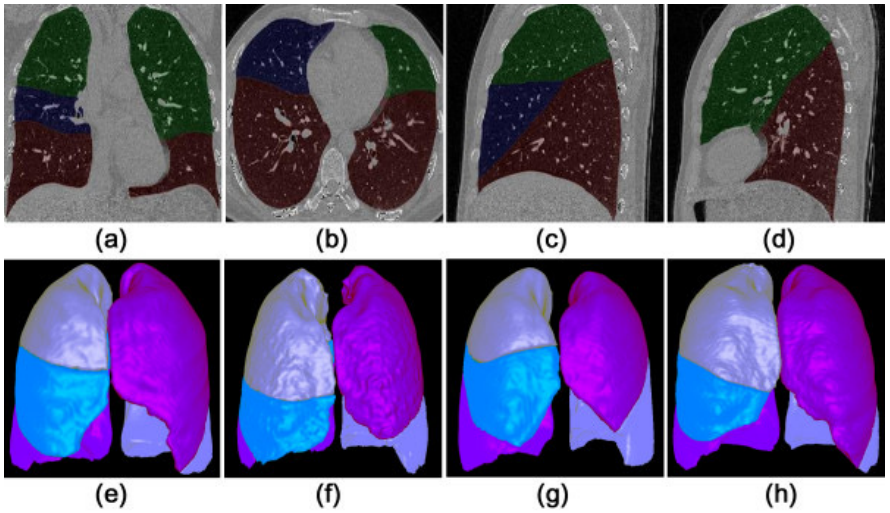
Brain Analysis



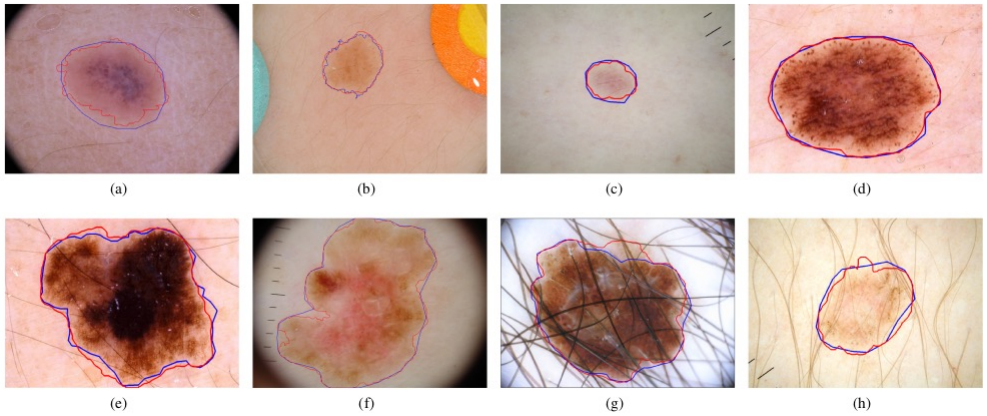
Eye Vessel Analysis

Image Segmentation Applications

- Biomedical Applications



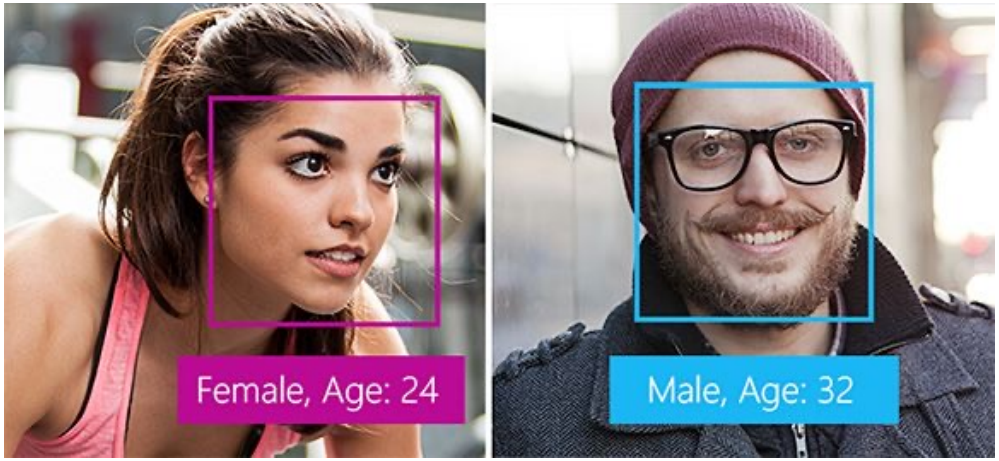
Lung Analysis



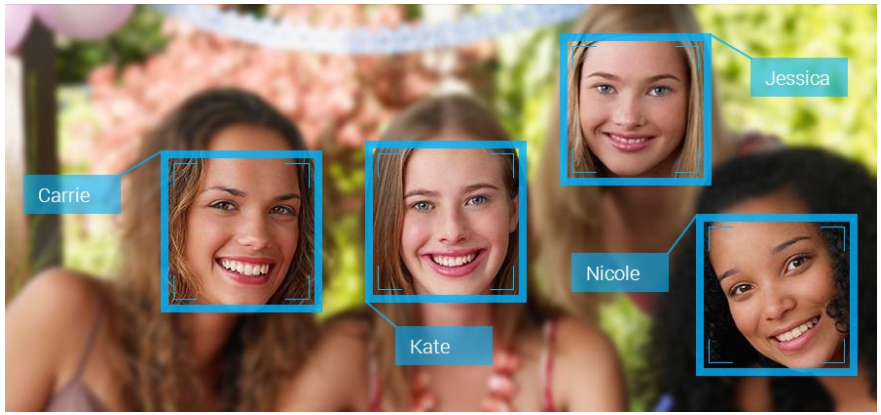
Melanoma Analysis

Image Segmentation Applications

- Face and Gesture Detection



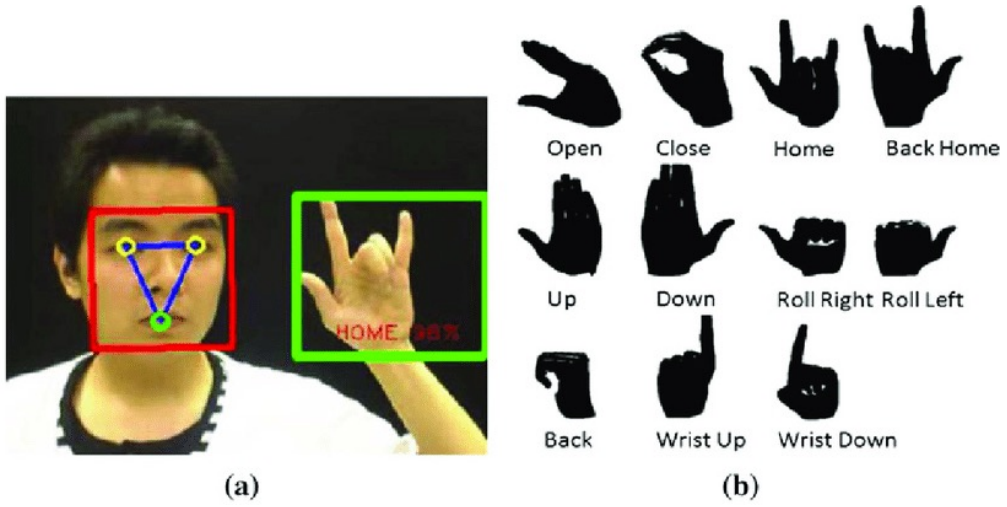
Face Analysis



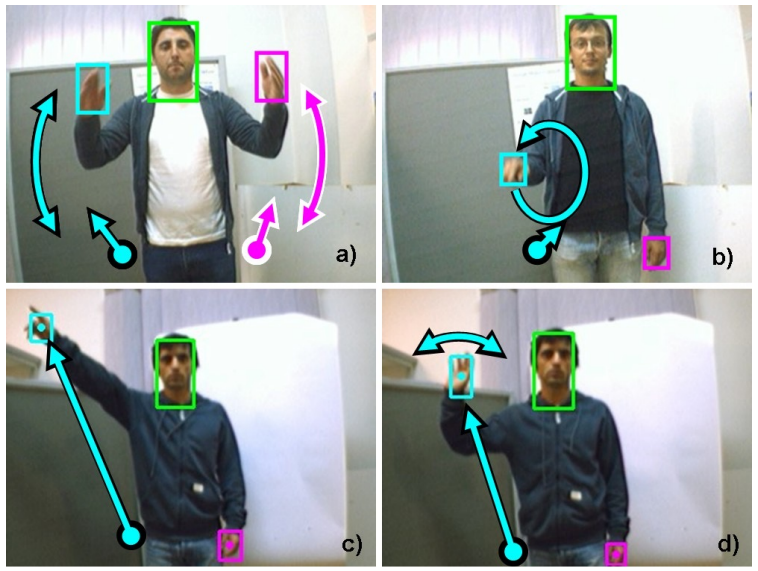
Face Analysis

Image Segmentation Applications

- Face and Gesture Detection



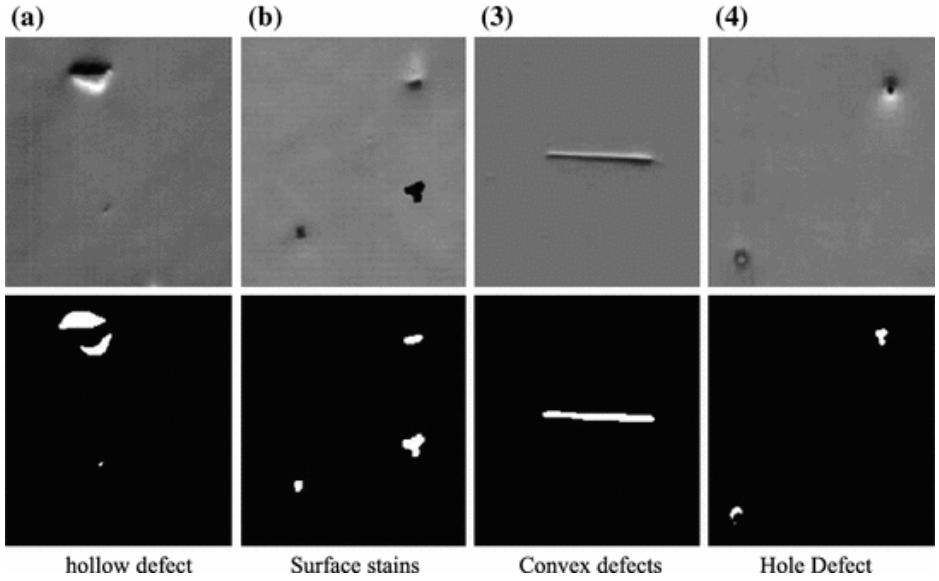
Gesture Analysis



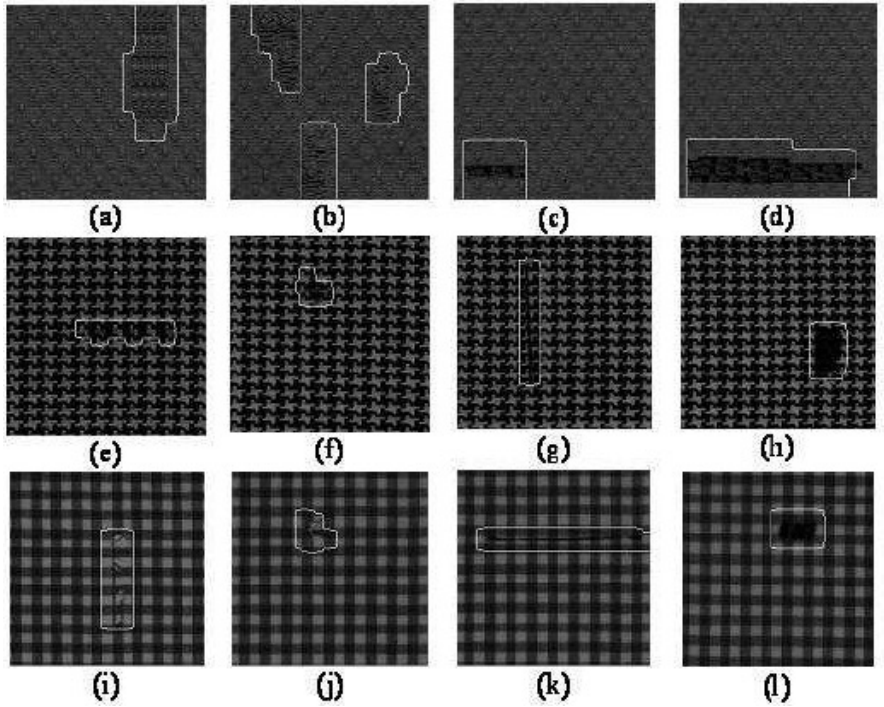
Gesture Analysis

Image Segmentation Applications

- Defect Detection



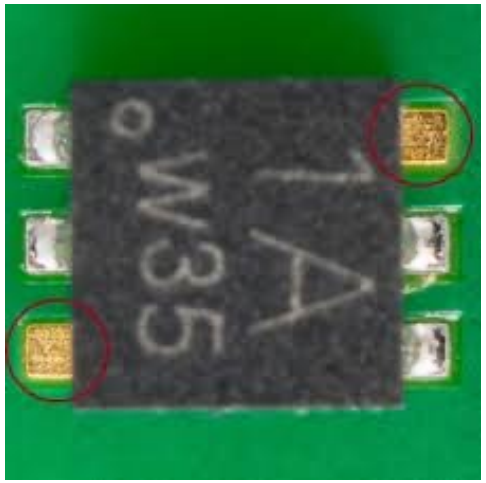
Defect Analysis



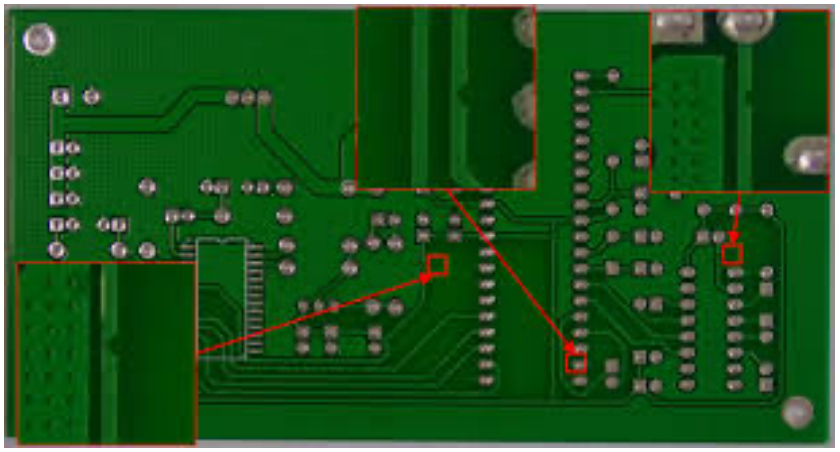
Defect Analysis

Image Segmentation Applications

- Defect Detection



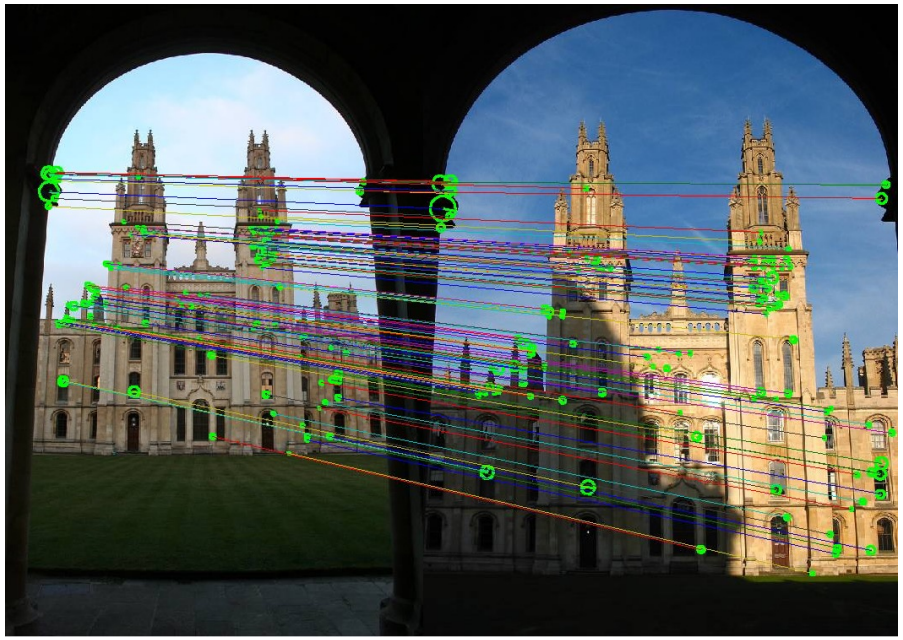
Defect Analysis



Defect Analysis

Image Segmentation Applications

- Content-based Image Retrieval



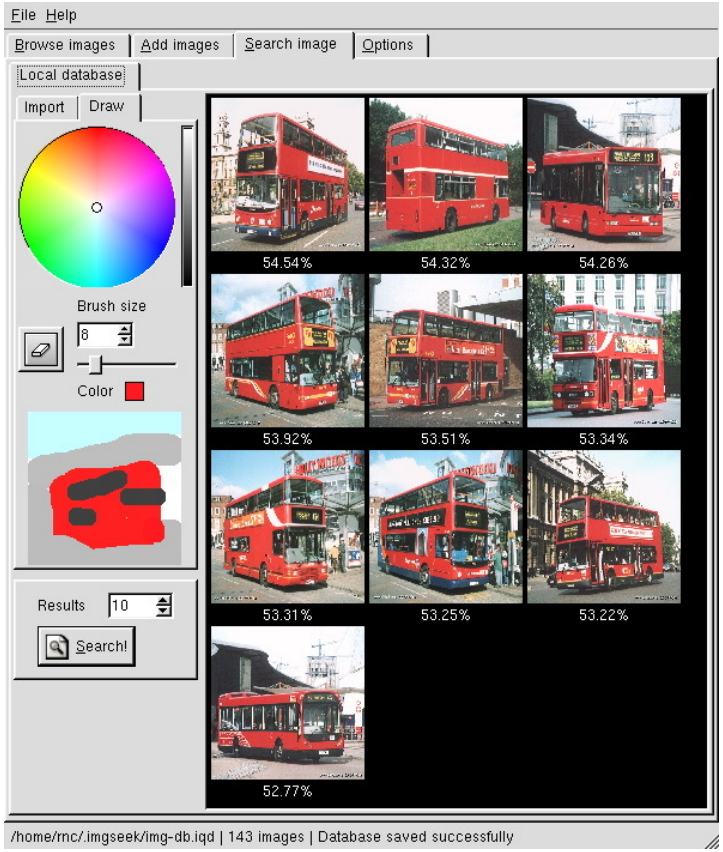
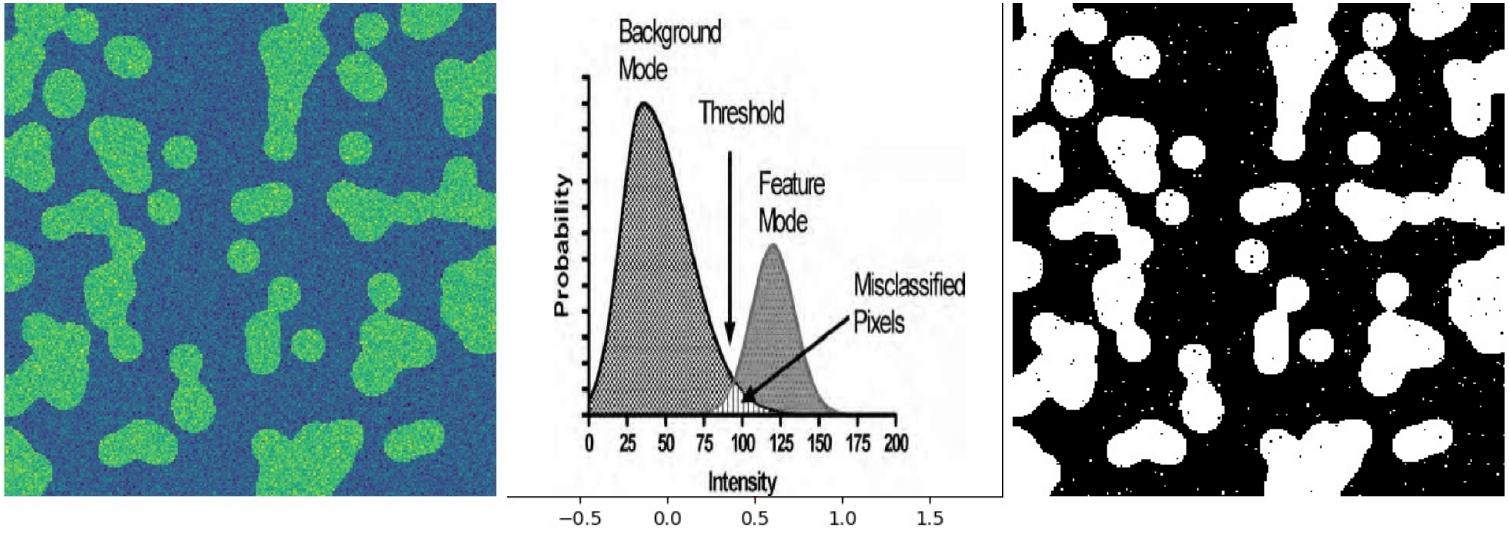


Image Retrieval

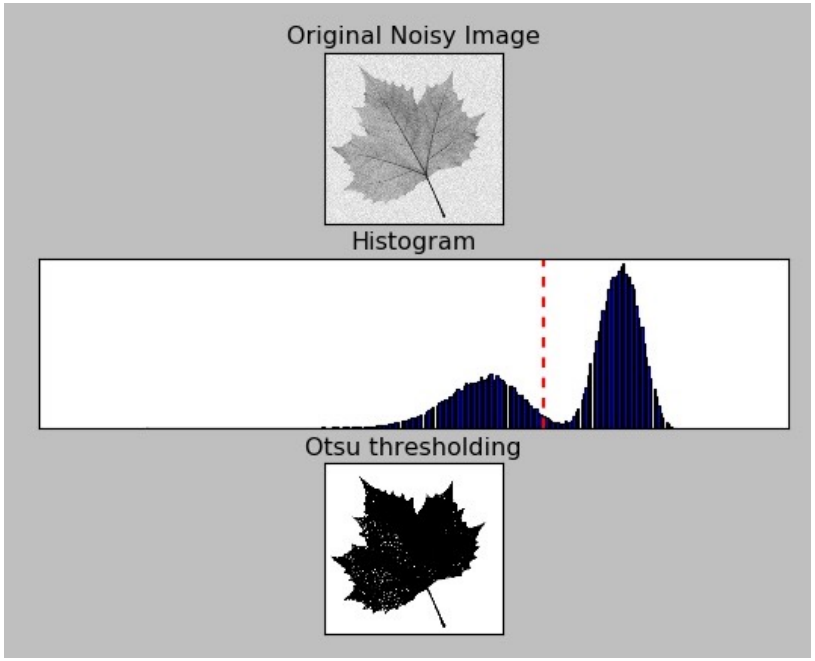
Thresholding-based Segmentation



Histogram Based Segmetation

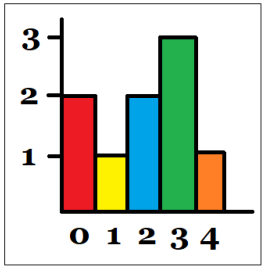
Dr. Tran Anh Tuan, Faculty of Mathematics and Computer Science, University of Science, HCMC

Thresholding-based Segmentation

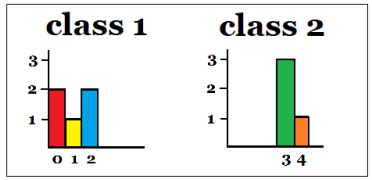


Within class variance

if pixels are classified into N classes (categories),
then the within class variance $(V_w) = \sum_{i=0}^{N} (W_i * \sigma_i^2)$,
where W_i is (# of pixels in class i)/(totoal pixel)



divide into two classes



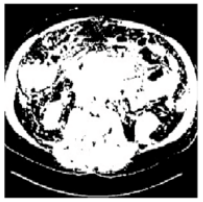
$W_1 = 5/9$ $W_2 = 4/9$

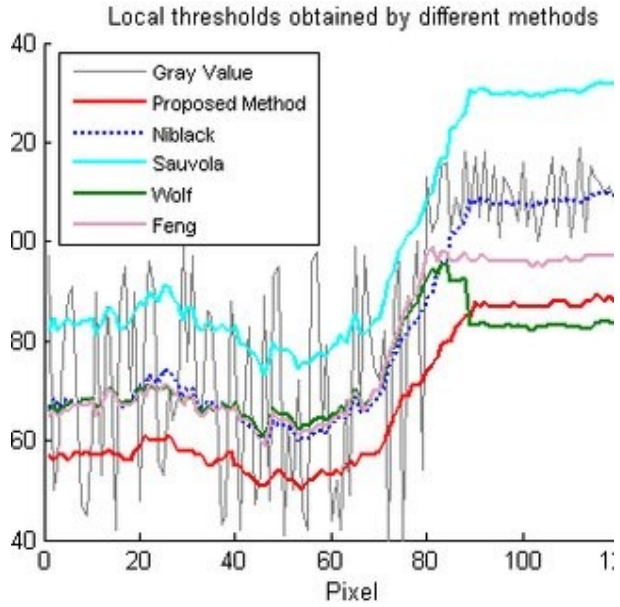
$\sigma_1^2 = 4/5$ $\sigma_2^2 = 3/16$

$V_w = W_1 * \sigma_1^2 + W_2 * \sigma_2^2 = 0.52777$

Otsu's Segmetation

Thresholding-based Segmentation

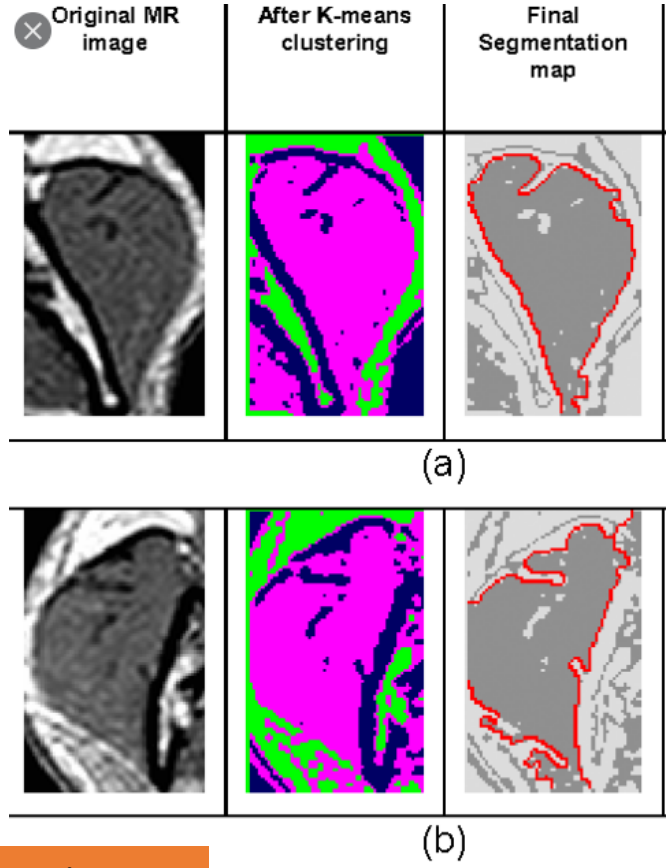
Niblack Method	Sauvola Method	PNSR		Jaccard	
		Niblack	Sauvola	Niblack	Sauvola
		58.9606	52.0983	0.5218	0.2771
		54.2790	52.3042	0.3324	0.2117
		55.0773	52.5917	0.4460	0.2689
		68.6692	54..2790	0.4074	0.2050



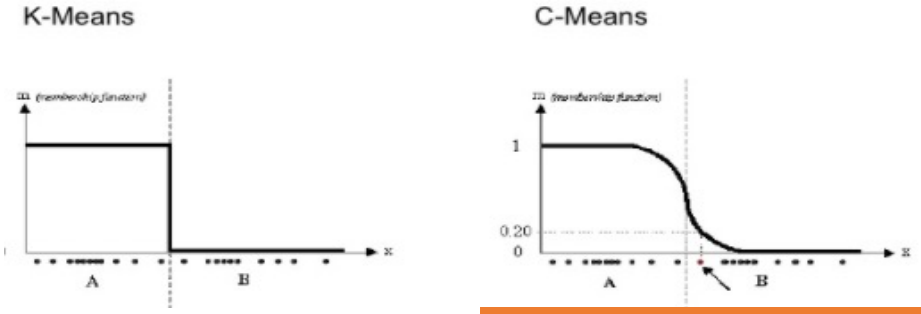
$$T(x,y) = m(x,y) + [1 + k(\frac{s(x,y)}{R} - 1)]$$

Local thresholding (Niblack's Techniques)

Clustering-based Segmentation

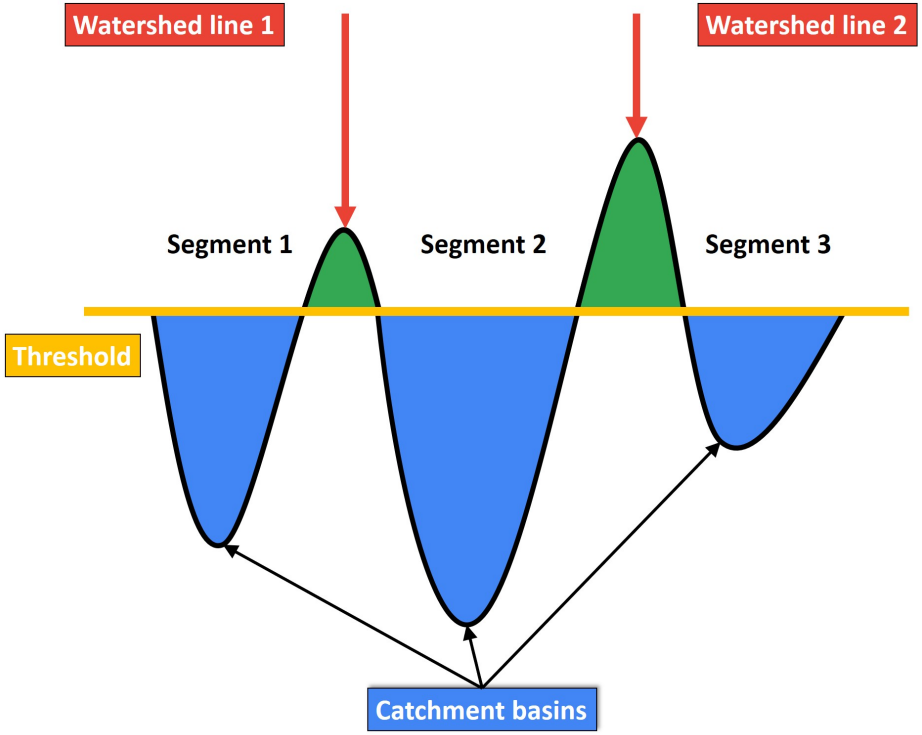


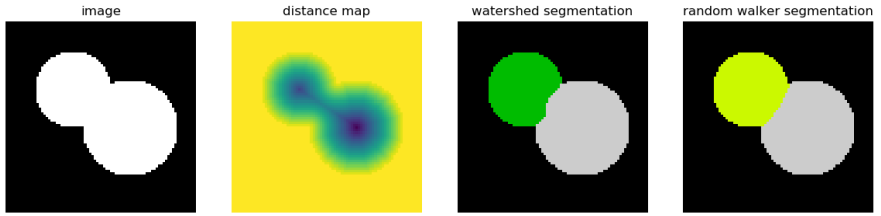
Membership Function

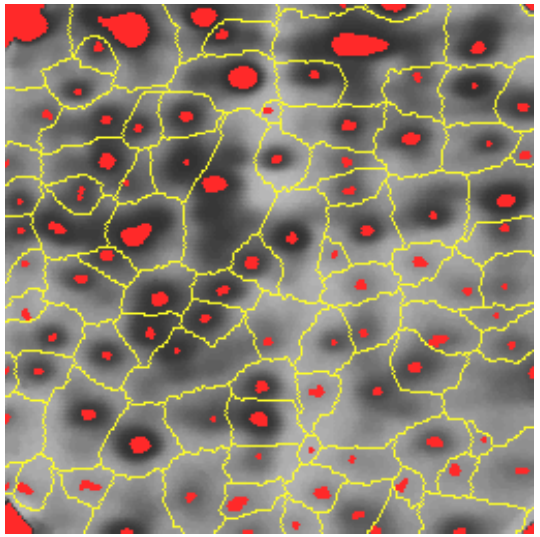


Kmeans and Fuzzy C-Means Clustering

Region-based Segmentation



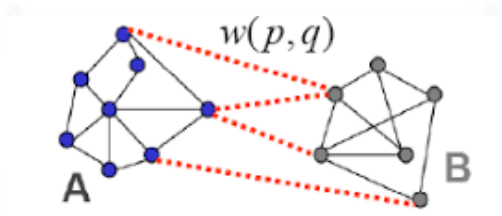




Watershed Segmentation

Dr. Tran Anh Tuan, Faculty of Mathematics and Computer Science, University of Science, HCMC

Graph-based Segmetation



$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

G = {V,E}

V: graph nodes
E: edges connection nodes

Pixels
Pixel similarity

Intensity

$$W(i,j) = e^{\frac{-\|I_{(i)} - I_{(j)}\|_2^2}{\sigma_I^2}}$$

Distance

$$W(i,j) = e^{\frac{-\|X_{(i)} - X_{(j)}\|_2^2}{\sigma_X^2}}$$

Texture

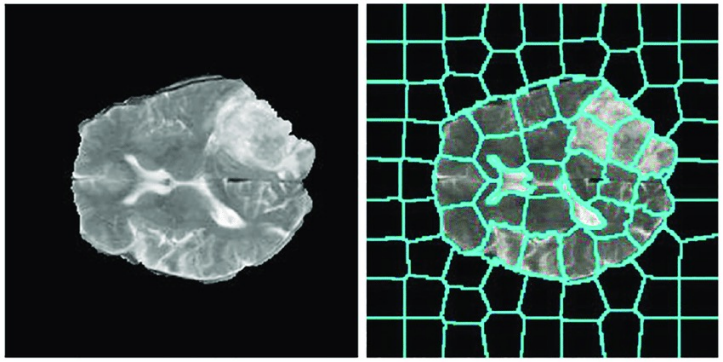
$$W(i,j) = e^{\frac{-\|c_{(i)} - c_{(j)}\|_2^2}{\sigma_c^2}}$$

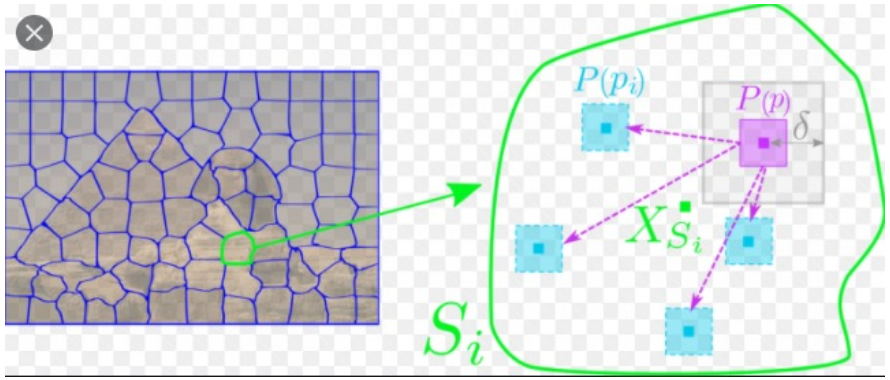
Criterion for partition:

$$\min cut(A,B) = \min_{A,B} \sum_{u \in A, v \in B} w(u,v)$$

Normalized Cut

Graph-based Segmetation

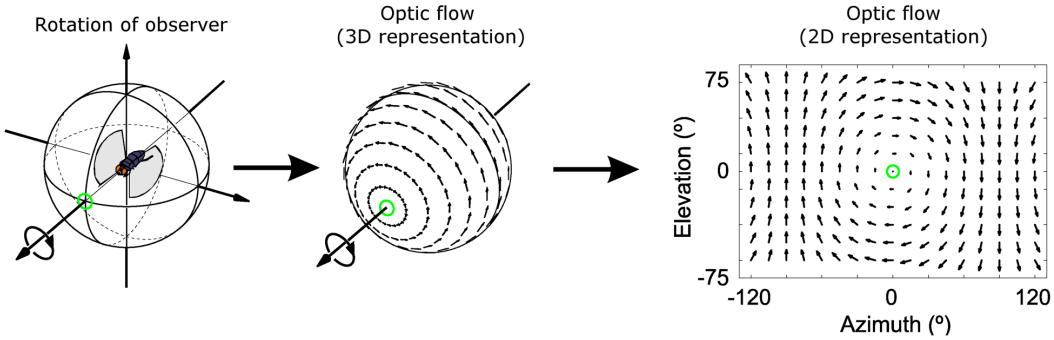




Superpixels Segmentation

Dr. Tran Anh Tuan, Faculty of Mathematics and Computer Science, University of Science, HCMC

Motion Analysis

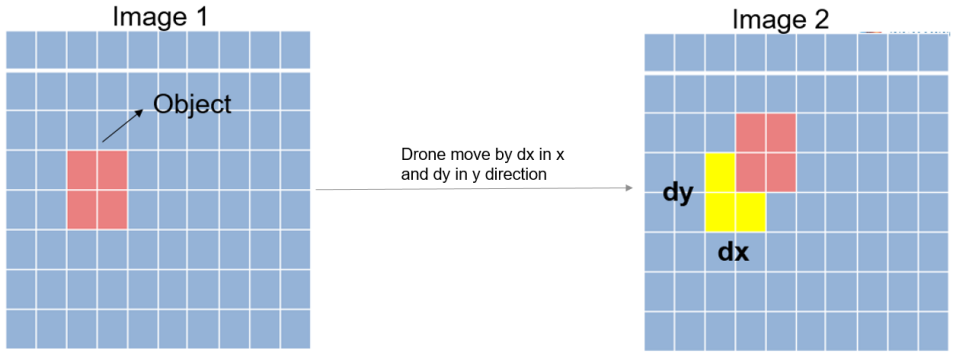


How to estimate pixel motion from image H to image I?

- Solve pixel correspondence problem
  - given a pixel in H, look for nearby pixels of the same colo in I

Key assumptions

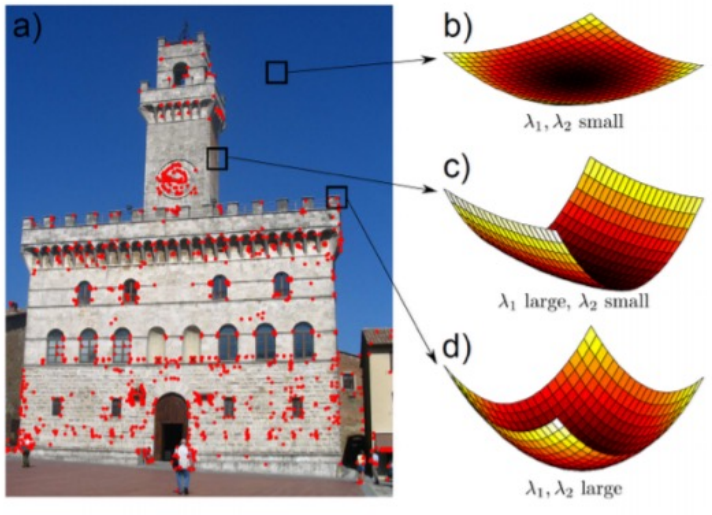
- **color constancy**: a point in H looks the same in I
  - For grayscale images, this is **brightness constancy**
- **small motion**: points do not move very far



Optical Flow

Dr. Tran Anh Tuan, Faculty of Mathematics and Computer Science, University of Science, HCMC

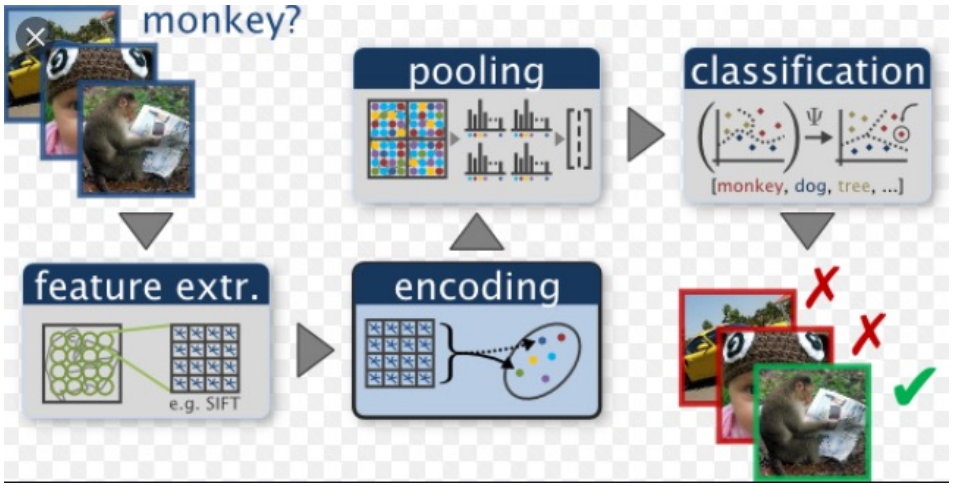
Feature Extraction & Machine Learning



Make decision based on image structure tensor

$$\mathbf{H} = \sum_{i \in \mathcal{N}} \frac{\partial I(\mathbf{x}_i)}{\partial \mathbf{x}} \frac{\partial I(\mathbf{x}_i)}{\partial \mathbf{x}}^T$$

Harris Corner Detectors

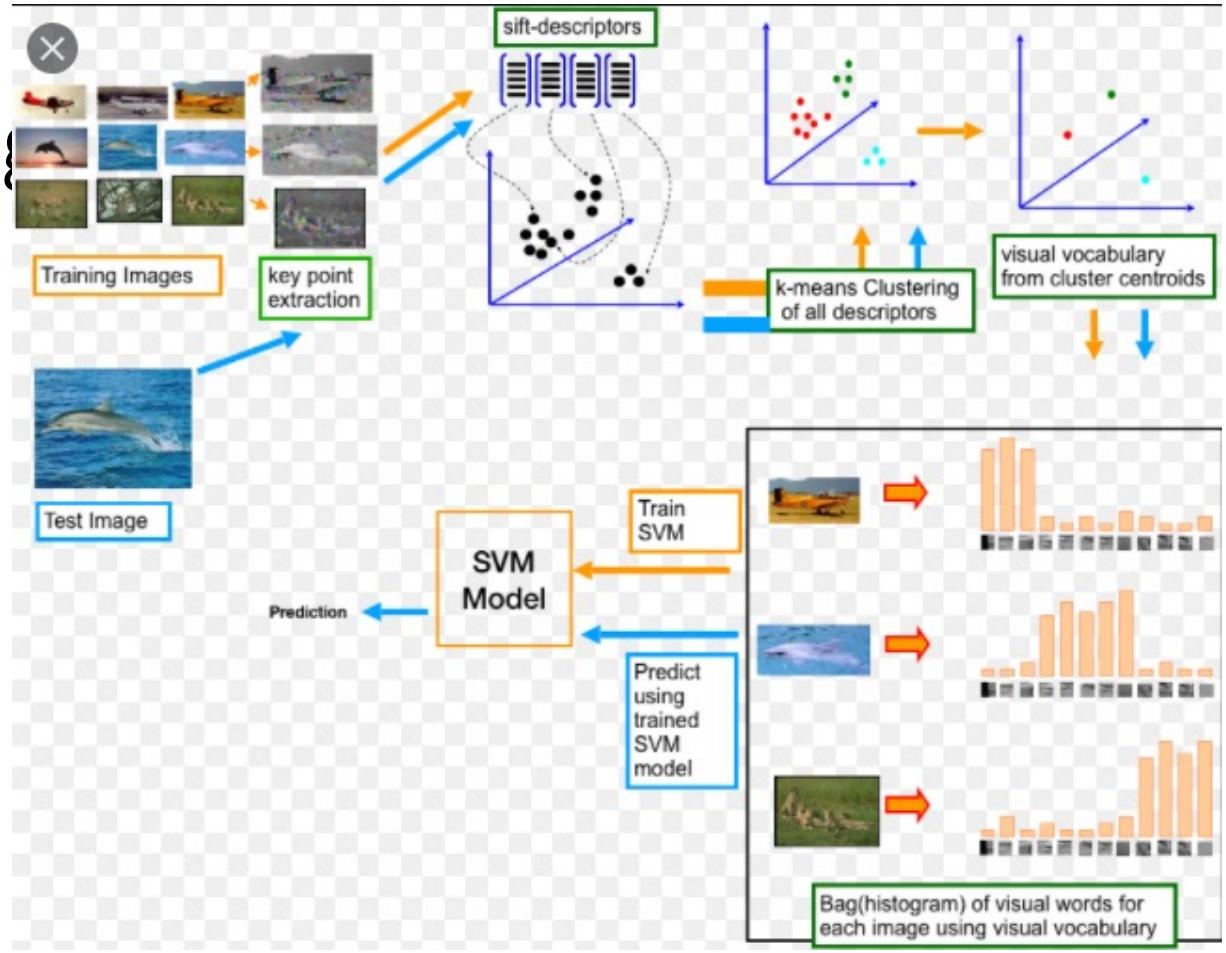


SIFT Descriptors

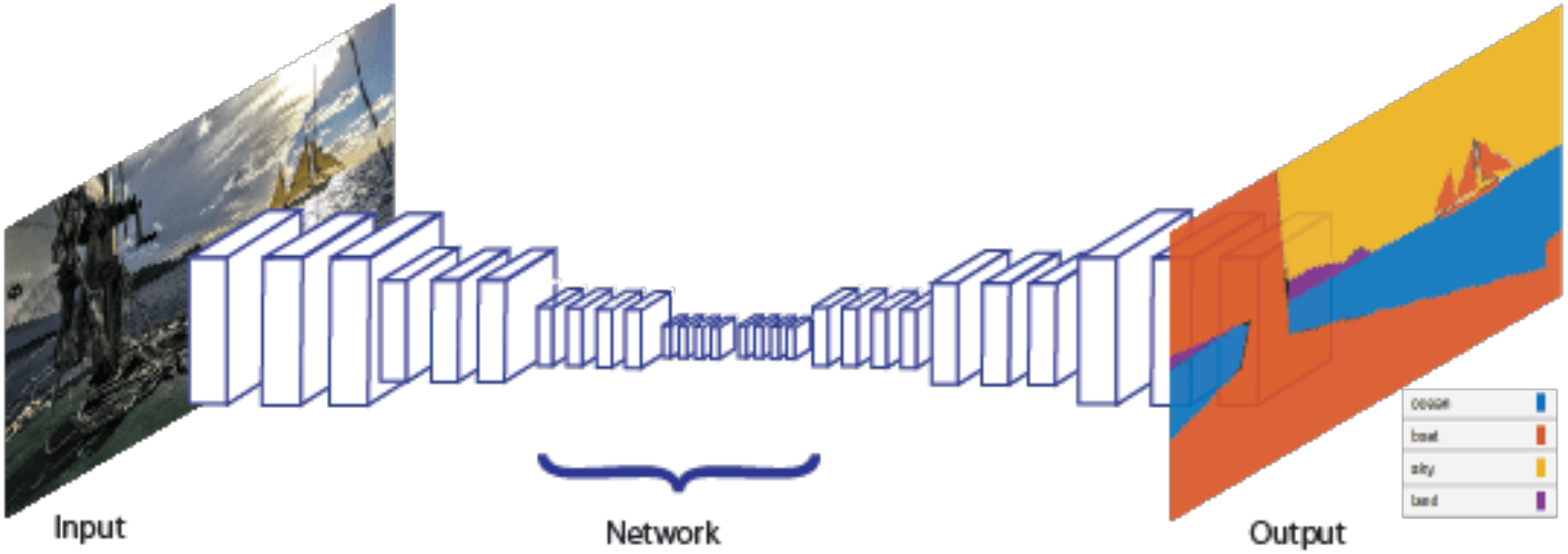
Dr. Tran Anh Tuan, Faculty of Mathematics and Computer Science, University of Science, HCMC

Feature Extraction & Machine Learning

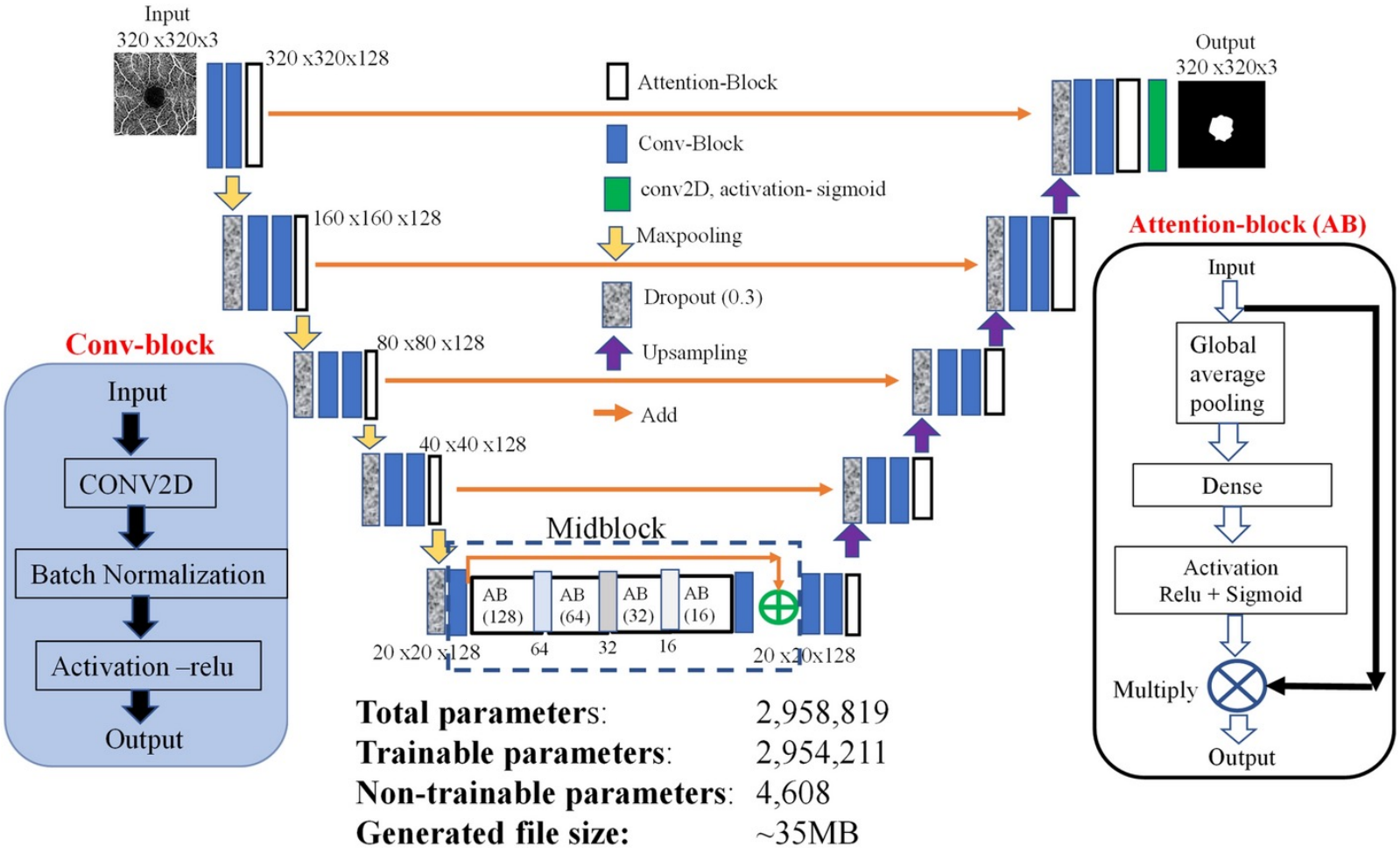
SIFT Descriptors + SVM



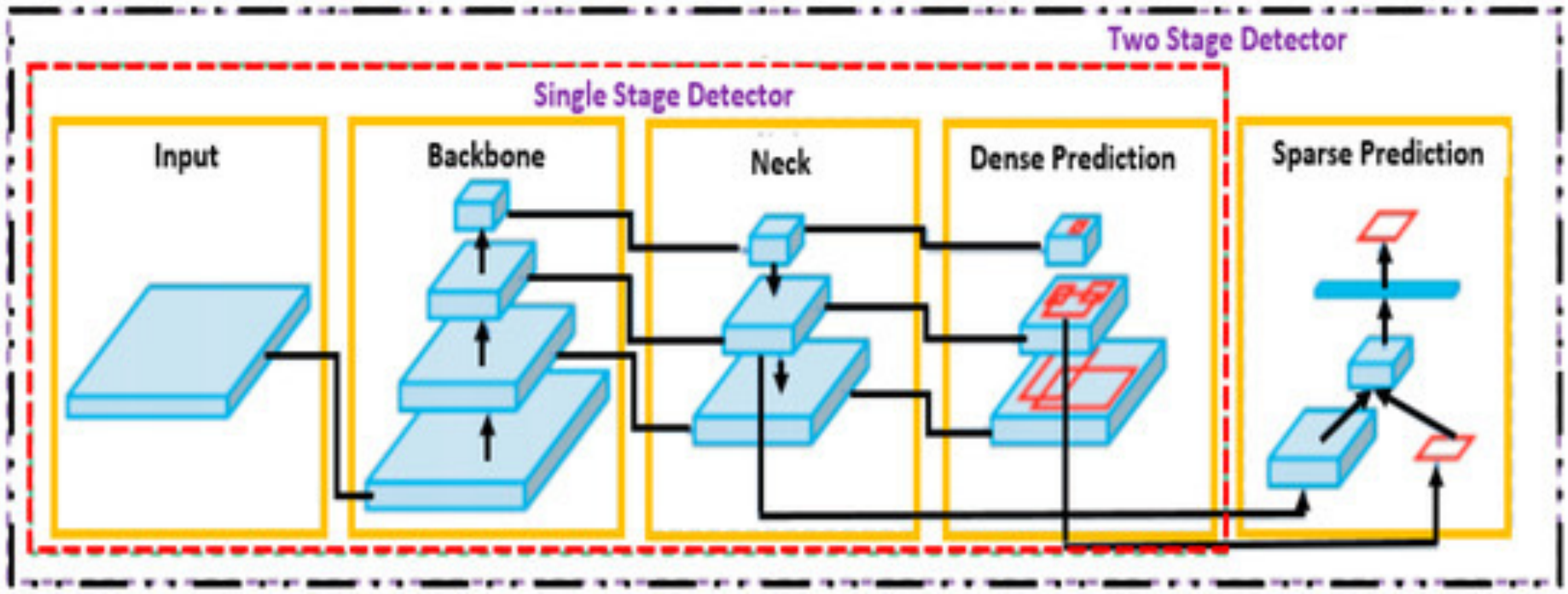
Deep Learning Segmentation

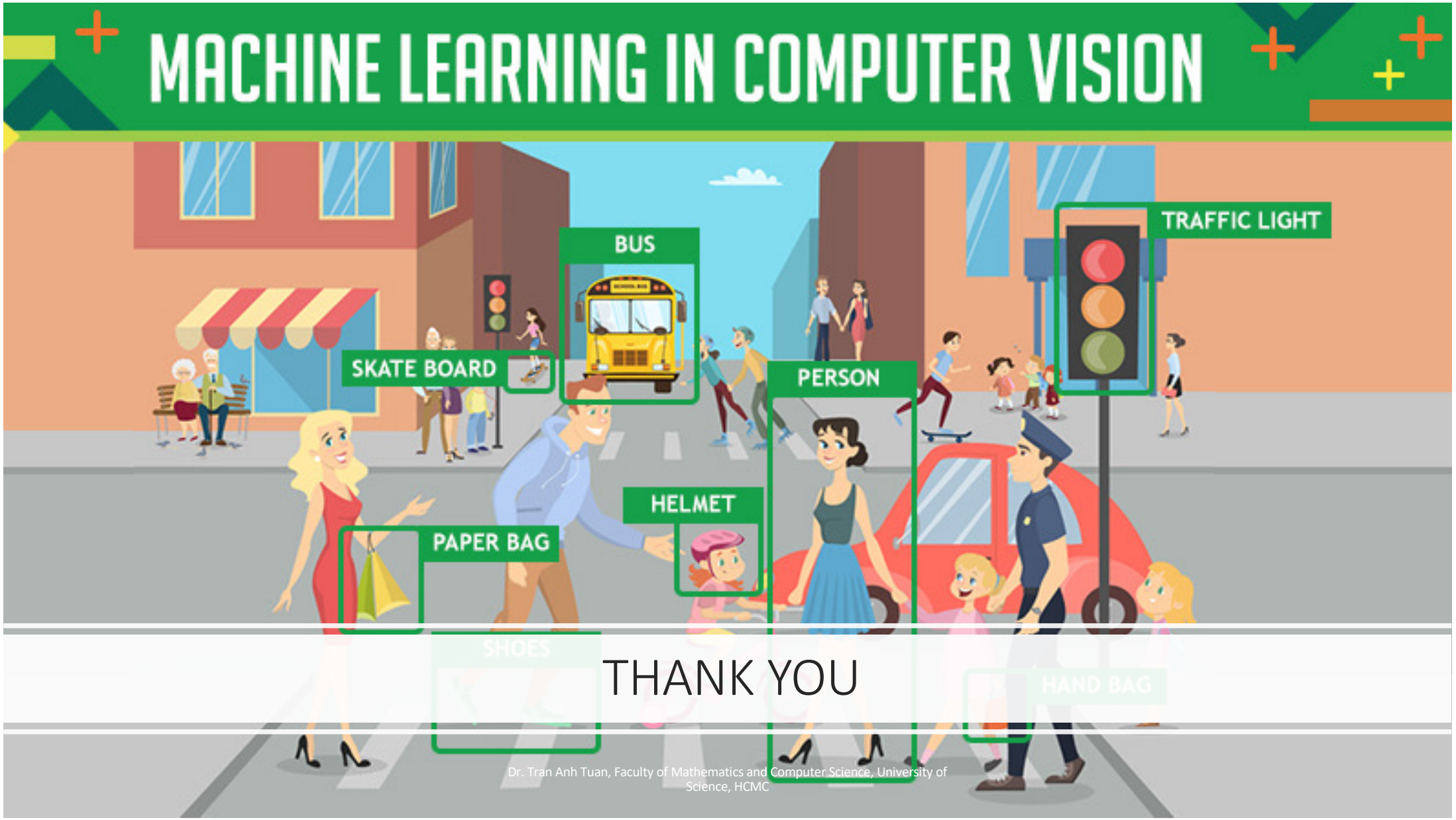


Deep Learning Segmentation



Deep Learning Segmentation





THANK YOU

Dr. Tran Anh Tuan, Faculty of Mathematics and Computer Science, University of Science, HCMC