Department of Computer Science
University of Bristol

COMS30030 - Image Processing and Computer Vision

Lecture 06

Object Detection

Majid Mirmehdi | majid@cs.bris.ac.uk

What is ‘Object Detection’?

- Object detection aims at bridging the ‘semantic gap‘ between...

  - given pixel values, *and*
  - meaningful objects (grouping of pixels + classification of groups)

Image regions need to be found and assigned with **semantic labels** from a space of object classes

What is ‘Object Detection’?

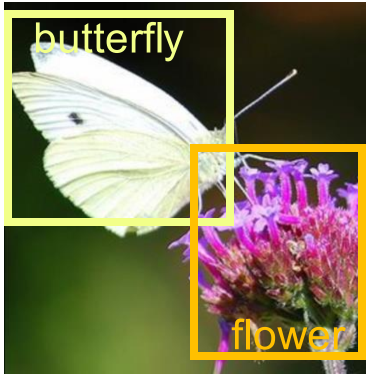
Why do classical shape detection and segmentation on their own rarely work for real-world object detection?

- high intra-class variance
- low inter-class variance
- classes are rarely well defined

- change of illumination, scale, pose, deformation, occlusion...

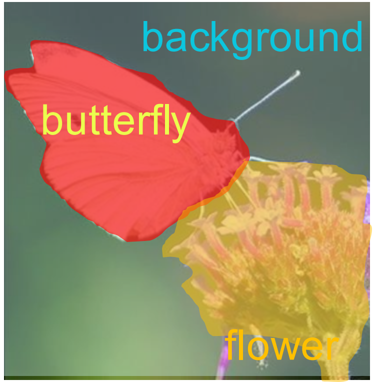
Terminology

Classification → butterfly

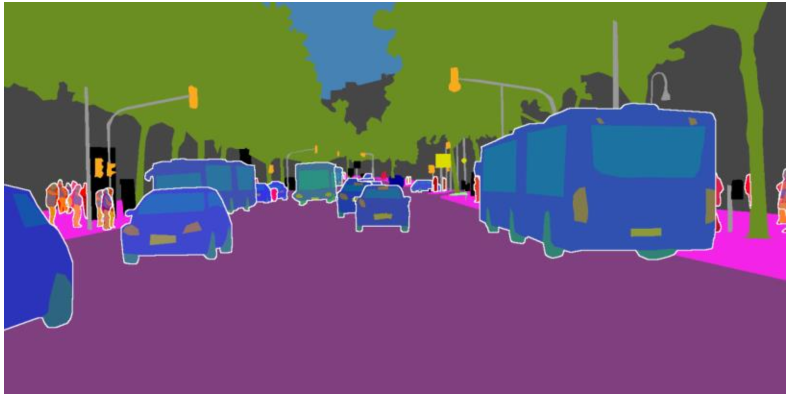


Multiple object detection

object detection =
Classification + localisation



Semantic Segmentation



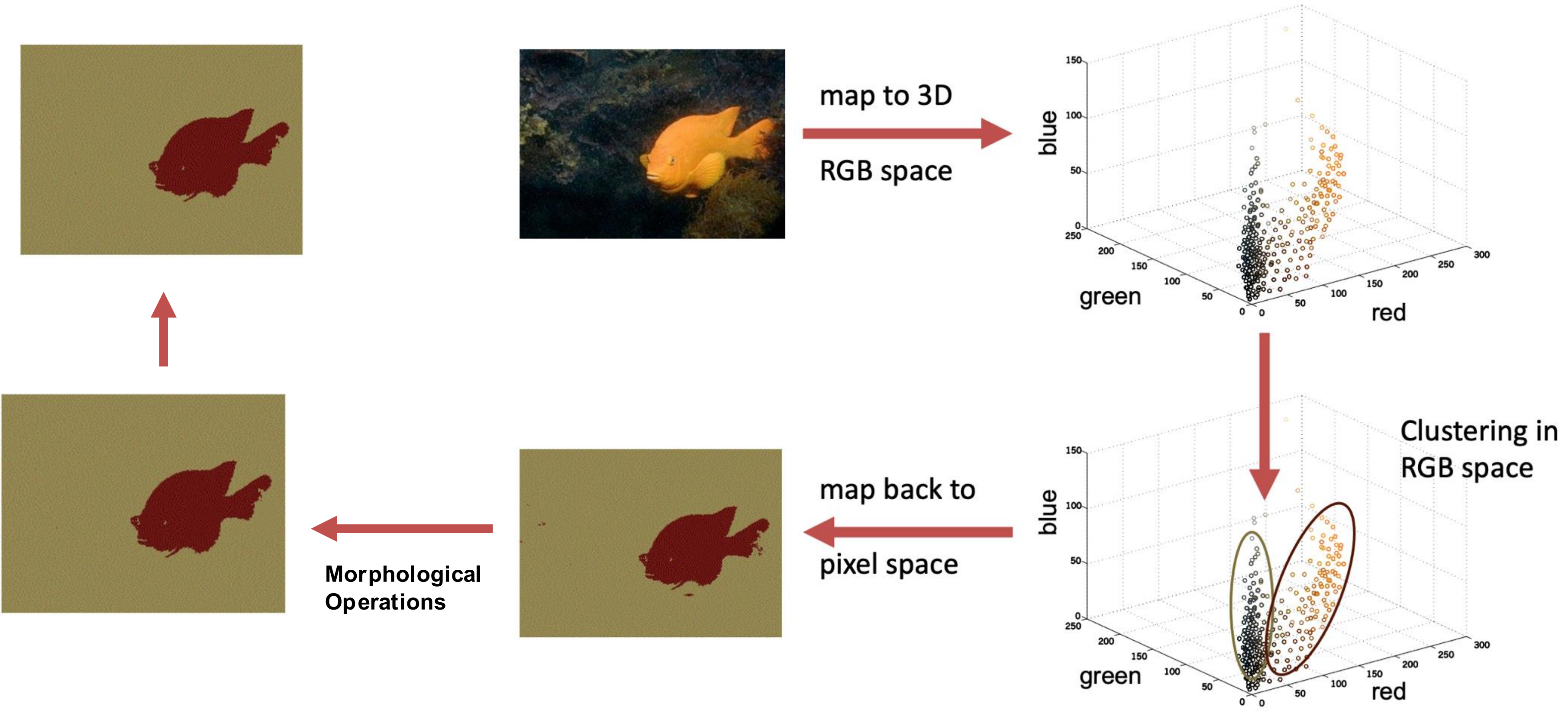
Panoptic Segmentation

Object Detection Techniques

- **Line and circle detection**: Techniques like the Hough Transform can be used to detect lines and circles in an image, which can indirectly help locate objects with specific geometric shapes.
- **Colour-based detection**: In some cases, objects can be detected based on their colour properties. This is especially useful when objects have distinct and consistent colors.
- **Template matching**: Using sliding a template over the input image and finding regions where the template best matches the local image content.
- **Classifiers with sliding window detectors**: Applying image classification on overlapped patches in the image.
- **Deep learning-based object detectors**: Object detector automatically learns image features required for detection tasks, and instance segmentation.

(out of scope in this unit)

Colour-based Detection



Morphological operations

What are they used for?

- Binary images (although version for greylevel images also exists)
- Can be used for **post-processing** segmentation results, e.g. noise filtering, enhancing object structure, ...
- Segmentation
- Quantitative description of objects (área, perimeter, etc.)

<u>Core techniques</u>

Erosion

Dilation

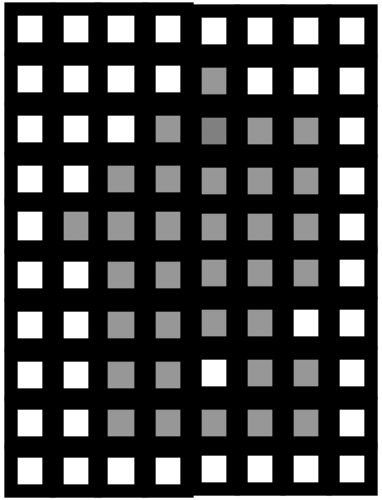
Opening

Closing

Morphological operations

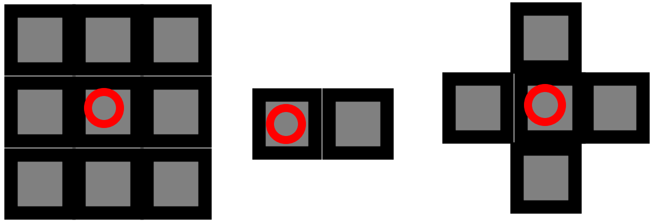
Two *sets*:

- Image
- Morphological ***kernel*** *(or structuring element)*



- Dilation (D)
  - Union of the **kernel** with the **image** set.
  - Increases resulting area.
- Erosion (E)
  - Intersection of the **kernel** with the **image** set.
  - Decreases resulting area.

Example ***kernels***



Next few slides inspired from Punam K Saha https://user.engineering.uiowa.edu/~aip/lectures/

Dilation

Morphological dilation '⊕' combines two sets using vector of set elements

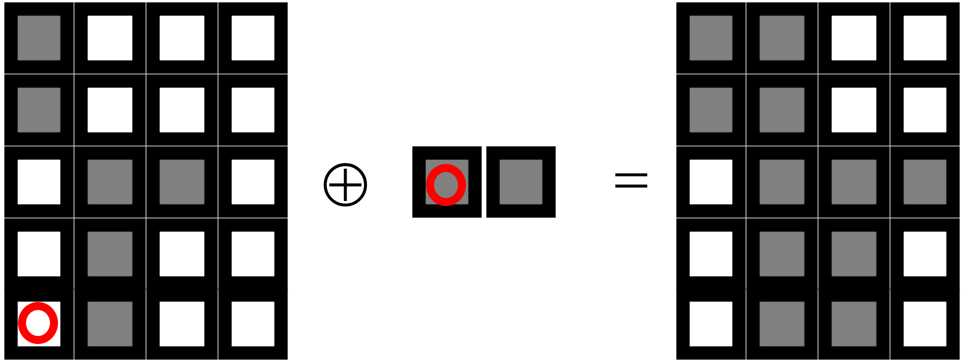
$$X \oplus B = \{p \in Z^2 \mid p = x + b, \quad x \in X, b \in B\}$$

Commutative: $X \oplus B = B \oplus X$

Associative: $X \oplus (B \oplus D) = (X \oplus B) \oplus D$

Invariant of translation: $X_h \oplus B = (X \oplus B)_h$

Is an increasing transformation: If $X \subseteq Y$ then $X \oplus B \subseteq Y \oplus B$



Erosion

Morphological erosion '⊖' combines two sets using vector subtraction of set elements and is a dual operator of dilation

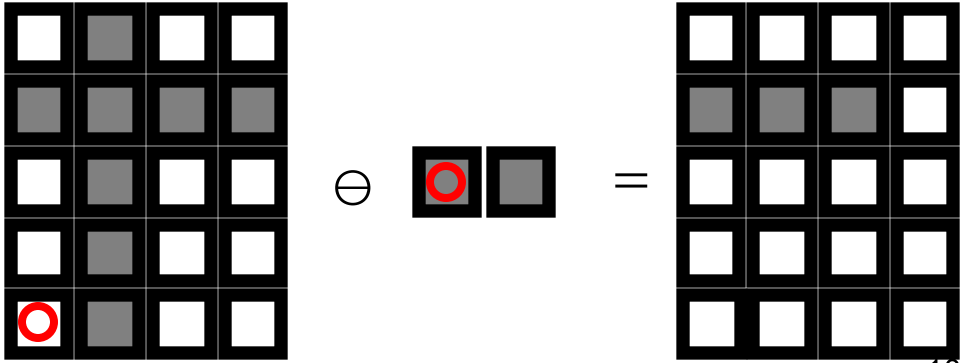
$$X \ominus B = \{p \in Z^2 \mid \forall\, b \in B, \quad p + b \in X\}$$

Not Commutative: $X \ominus B \neq B \ominus X$

Not associative: $X \ominus (B \ominus D) \neq (X \ominus B) \ominus D$

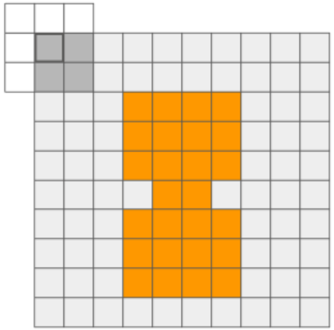
Invariant to translation: $X_h \ominus B = (X \ominus B)_h$ and $X \ominus B_h = (X \ominus B)_{-h}$

Is an increasing transformation: If $X \subseteq Y$ then $X \ominus B \subseteq Y \ominus B$

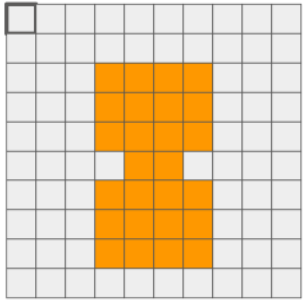


Dilation and Erosion examples

Dilation

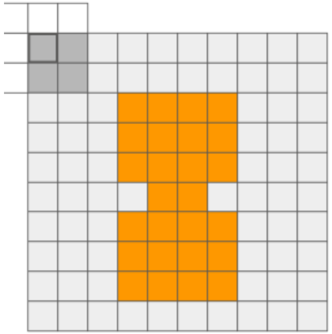


A=binary image

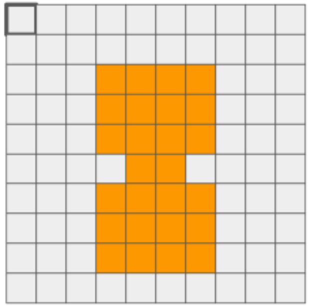


B =

1	1	1
1	1	1
1	1	1



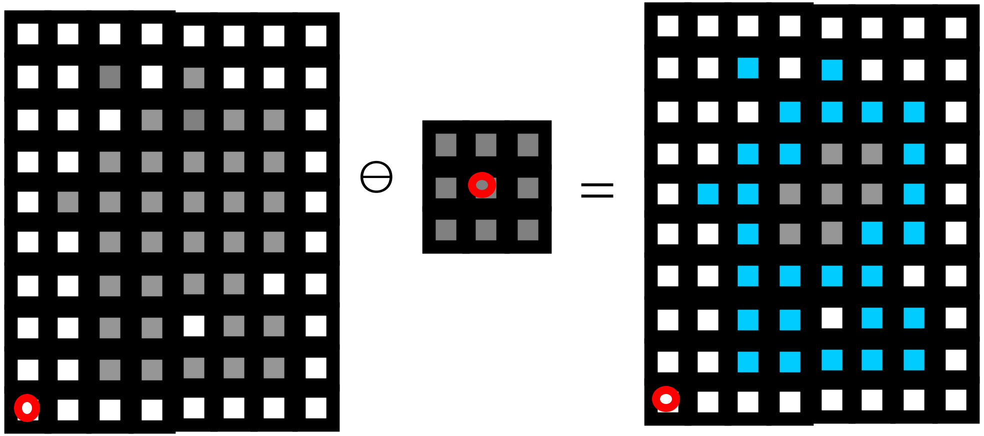
A=binary image

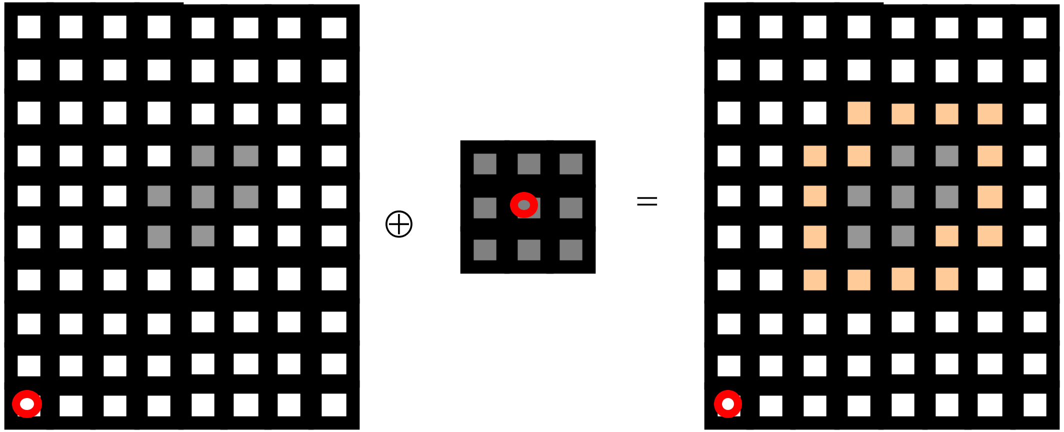


Erosion

Opening: Erosion followed by Dilation

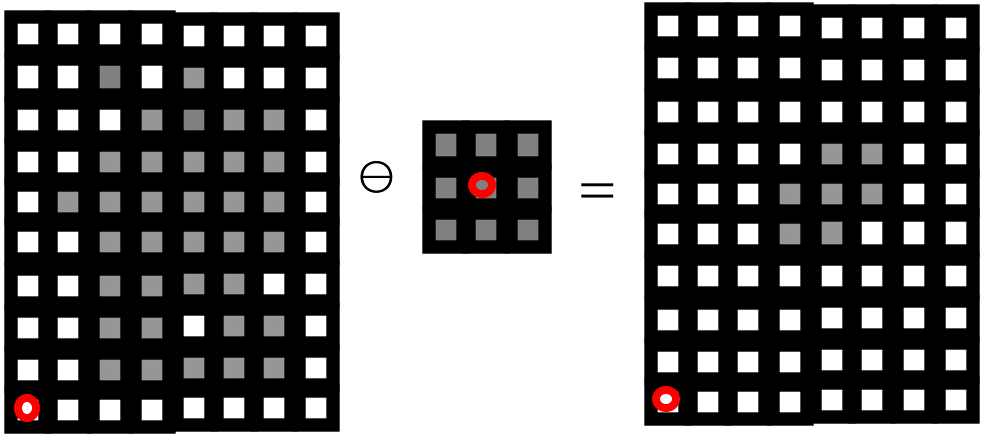
$$X \circ B = (X \ominus B) \oplus B$$

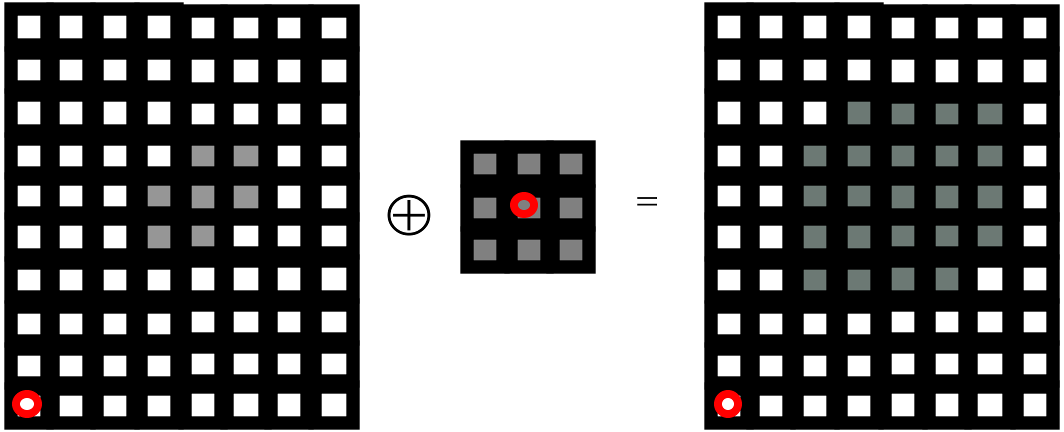




Opening: Erosion followed by Dilation

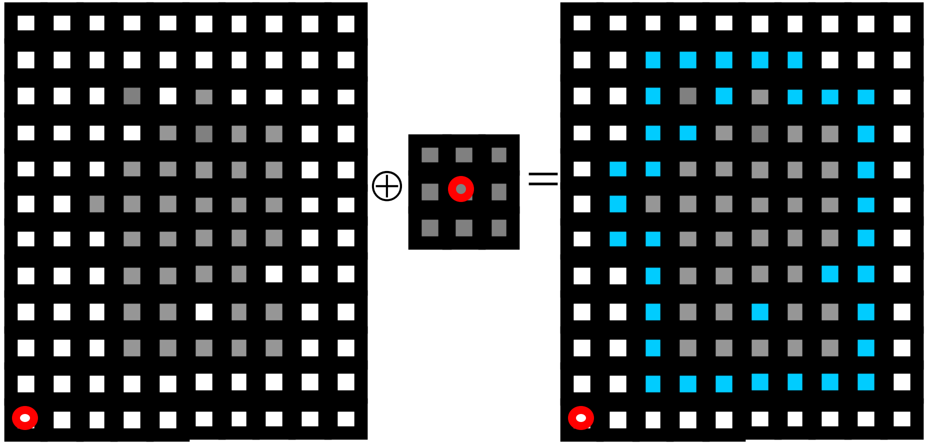
$$X \circ B = (X \ominus B) \oplus B$$

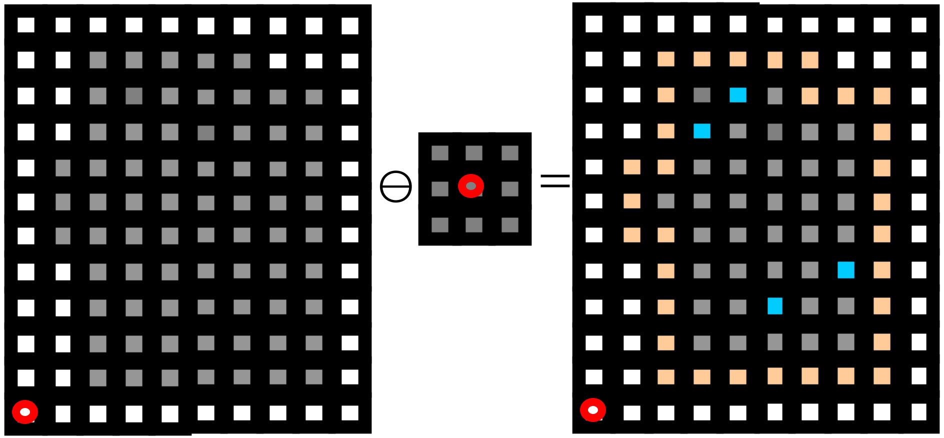




Closing: Dilation followed by Erosion

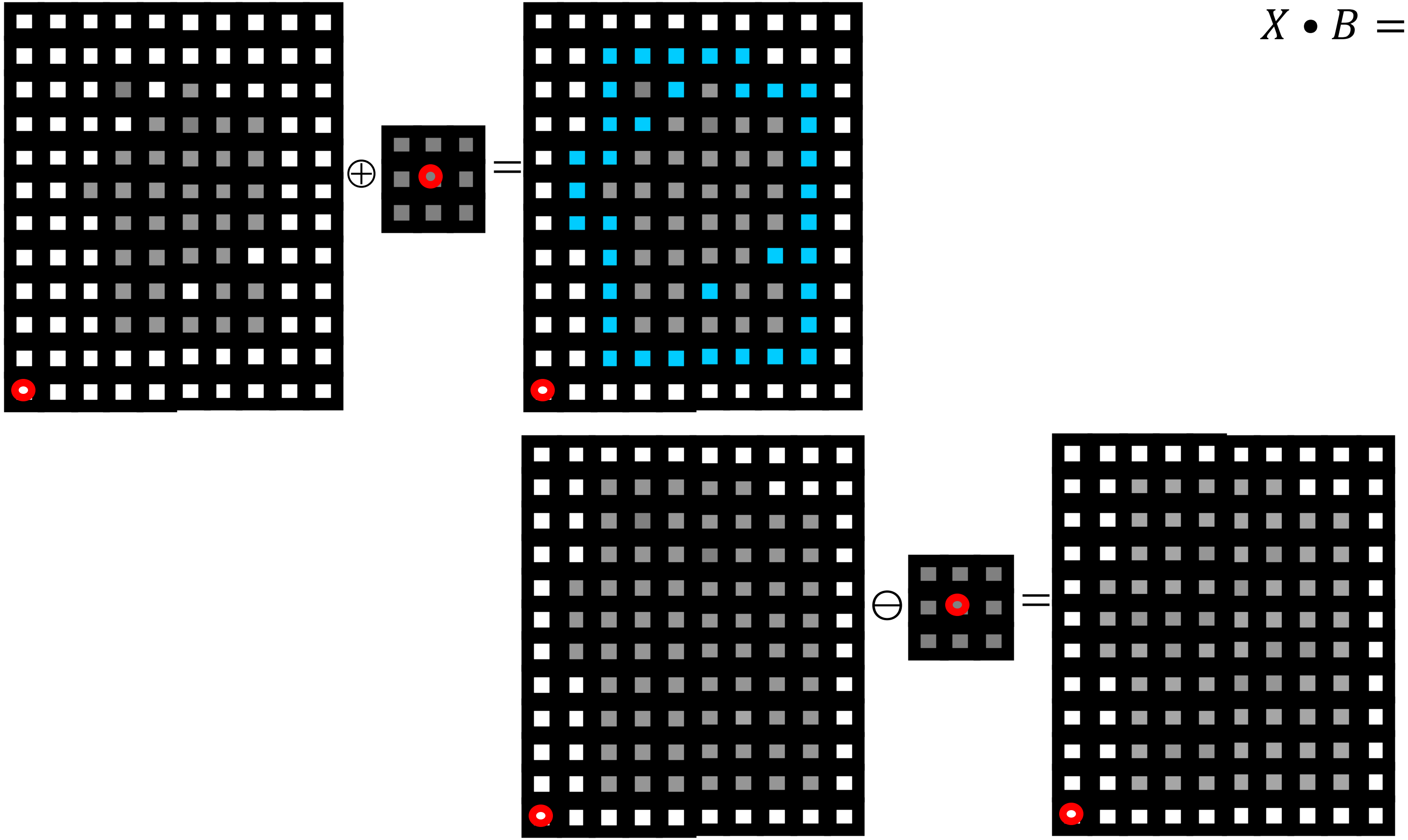
$$X \bullet B = (X \oplus B) \ominus B$$





Closing: Dilation followed by Erosion

$$X \bullet B = (X \oplus B) \ominus B$$



Examples

Orignial image

Eroded image

Dilated image

Examples

image

erosion

dilation

image

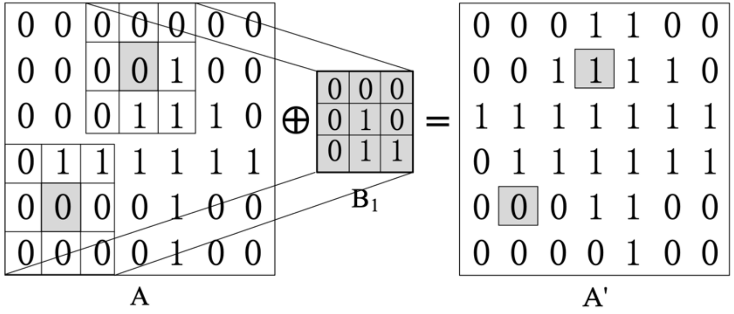
opening

erosion then dilation

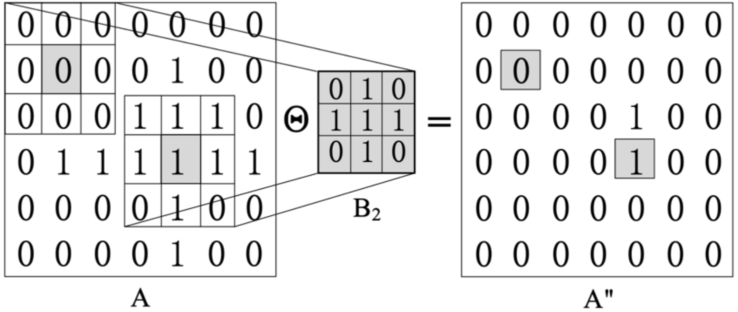
image

closing

dilation then erosion



(a) Dilation operator $A \oplus B_1 = A'$

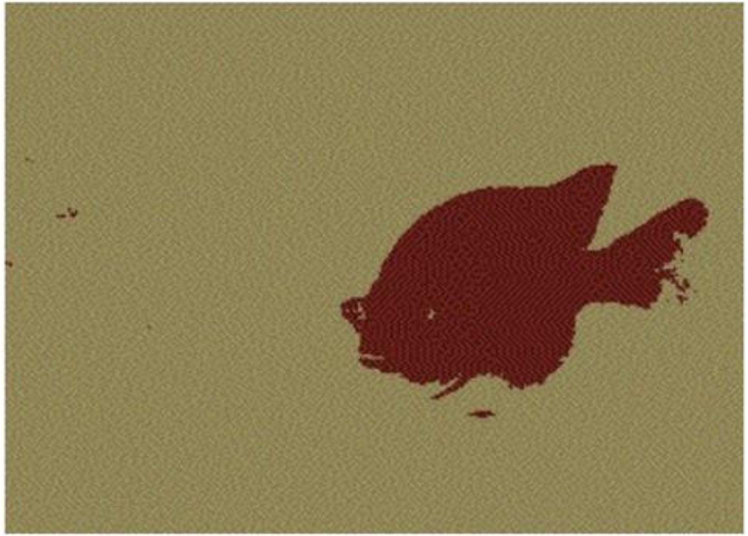


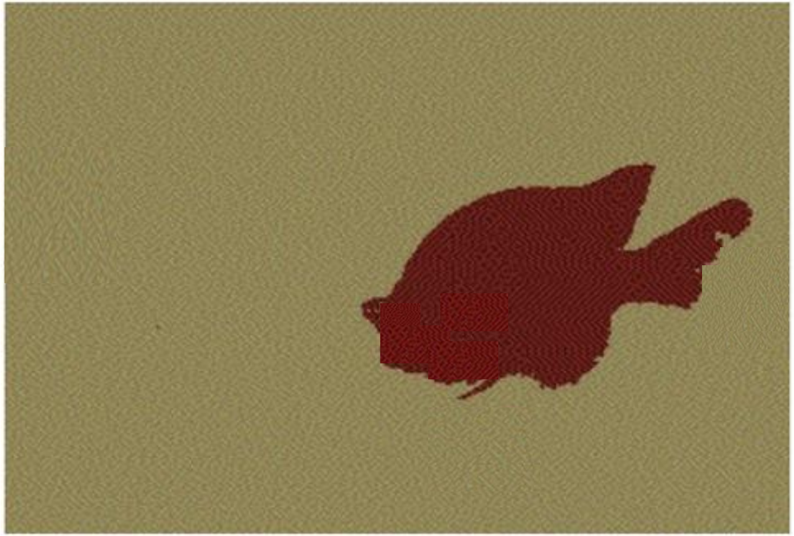
(b) Erosion operator $A \Theta B_2 = A''$

https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

https://doi.org/10.3390/ijgi9040186

Example of Opening

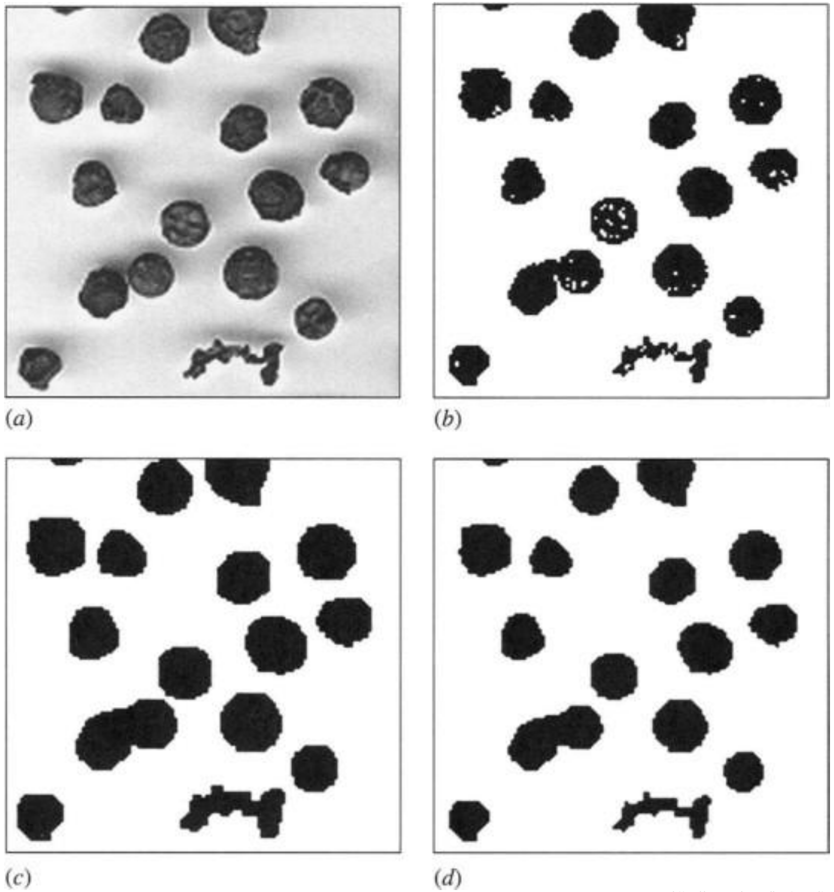




Example of Closing

(a) Image of peppercorns

(b) Thresholded



(c) 3x3 dilation…

(d) …then 3x3 erosion

https://www.sciencedirect.com/topics/engineering/morphological-operations

Example of Edge Detection!

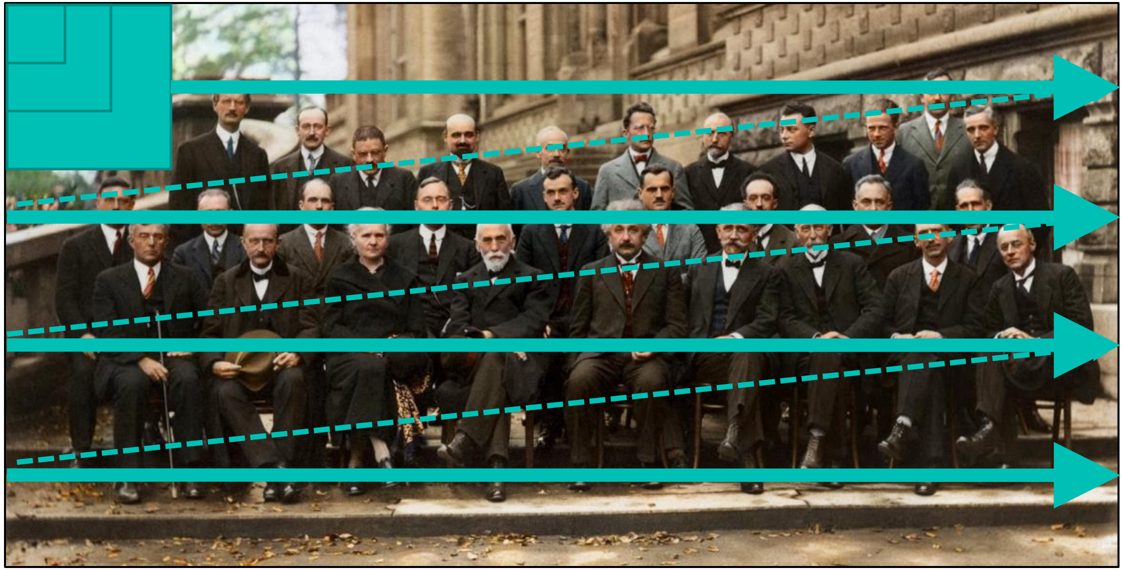
Erosion as isotopic shrink.

Contours obtained by subtraction of an eroded image from the original.

https://www.sciencedirect.com/topics/engineering/morphological-operations

Sliding Window Detectors

- Image is tested for object presence window-by-window
- The window is `slided' and `scaled' throughout the image



- Each resulting window is judged w.r.t. an object model giving a response indicating object presence or absence

Template Matching

- Find the best similarity (or the lowest difference) or within the defined threshold

pixel i in box y in the image, y has the same size as $\hat{y}$

pixel i in template $\hat{y}$

- Maximum

  correlation: $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\mu_y}{\sigma_y}\right)\left(\frac{\hat{y}_i-\mu_{\hat{y}}}{\sigma_{\hat{y}}}\right)$

mean

std

- Minimum

  mean absolute error: $\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$

  mean square error : $\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

total number of pixels in the template box

https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html

Template Matching

- Find the best similarity (or the lowest difference) or within the defined threshold

- correlation: $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\mu_y}{\sigma_y}\right)\left(\frac{\hat{y}_i-\mu_{\hat{y}}}{\sigma_{\hat{y}}}\right)$

Similarity map

- mean absolute error: $\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$
- mean square error : $\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

error map

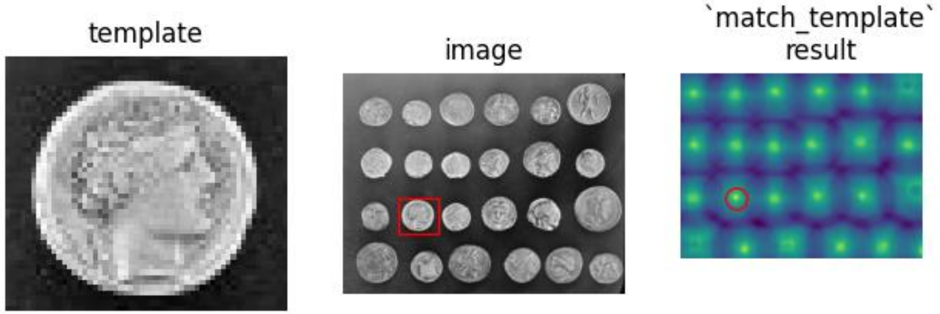
https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html

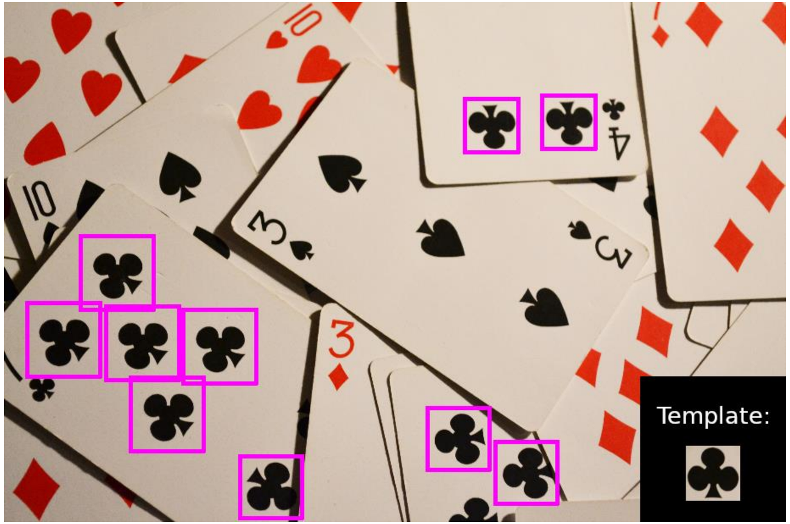
Template Matching can be expensive…

- Template image size: 53 x 48
- Source image size: 177 x 236
- Assumption: template image is inside the source image.
- Correlation (search) matrix size: 124 x 188

- Computation count: 124 x 188 x 53 x 48 = 59,305,728

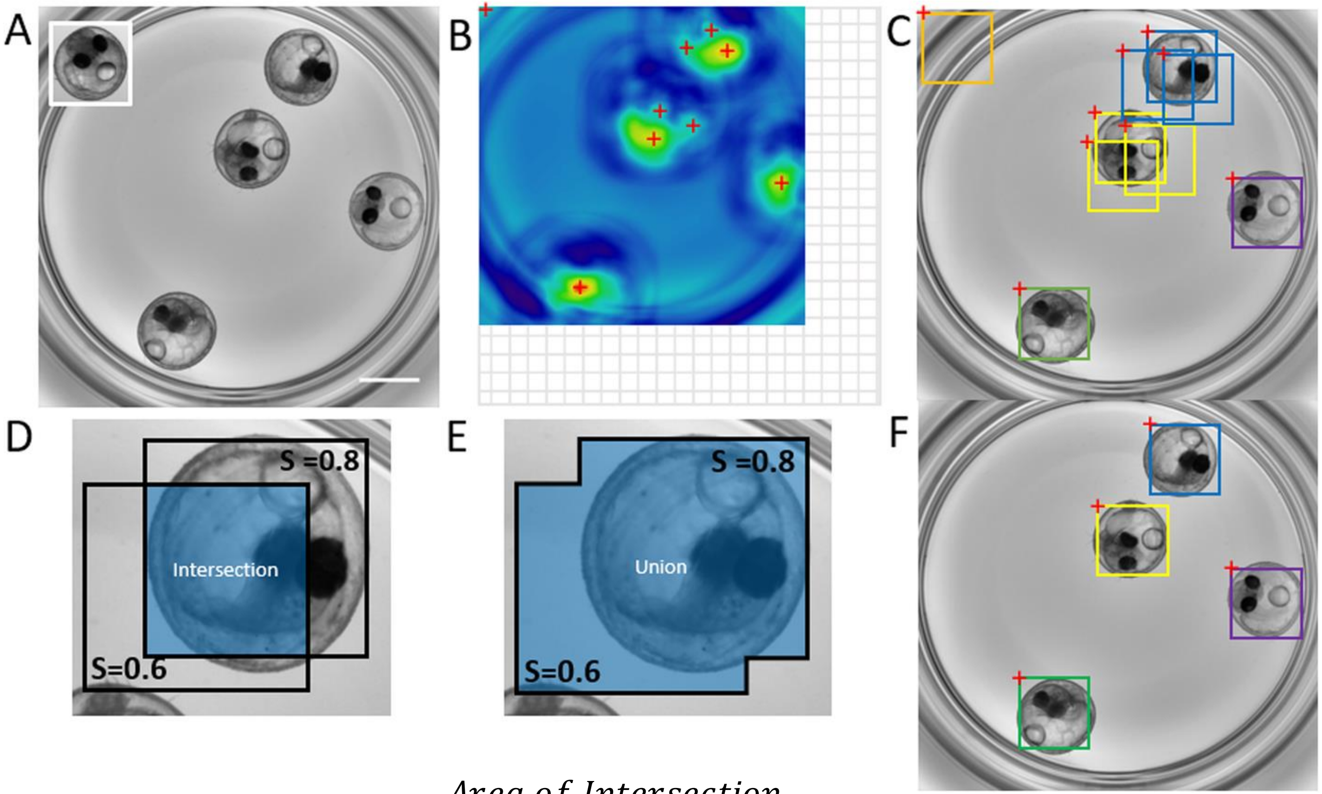
Template Matching examples





https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-3363-7

Template Matching examples



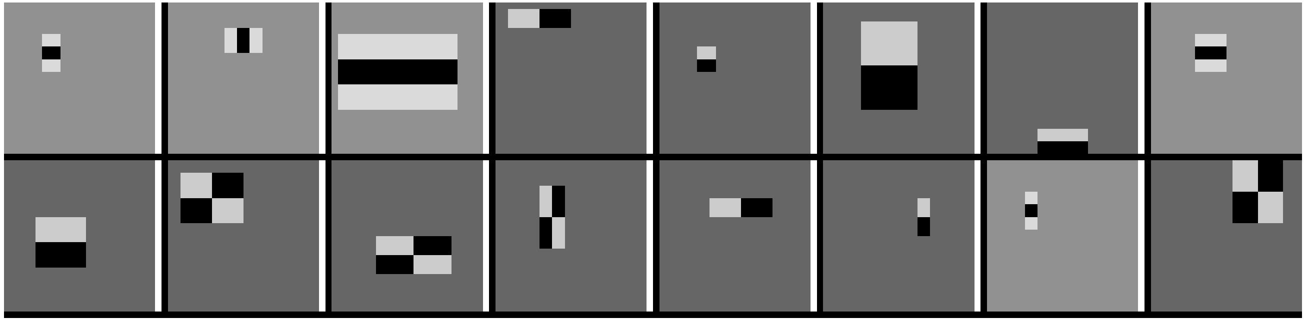
$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

Same object (IoU closer to 1) or distinct objects that are close to each other (IoU closer to 0).

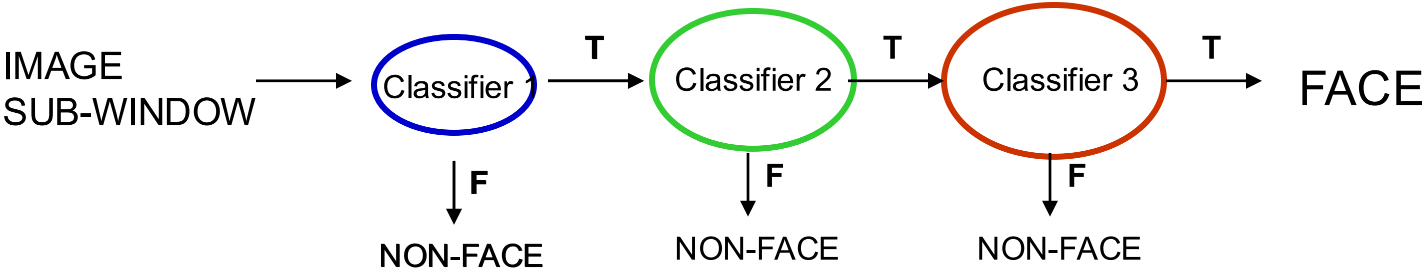
https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-3363-7

Viola-Jones: Another Sliding Window Approach

Hand-crafted weak features, but computationally efficient, calculated in sliding windows…

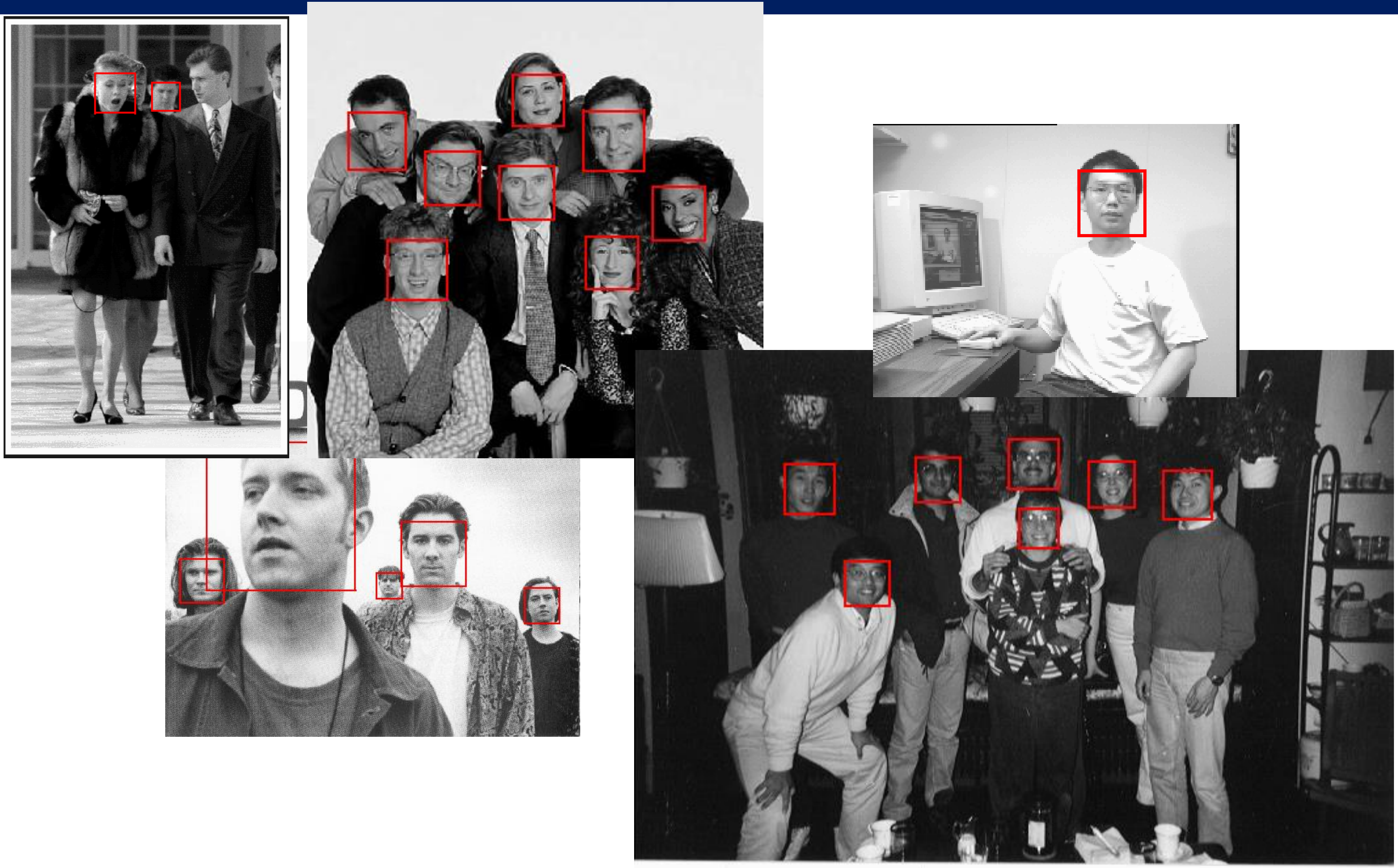


Construct a cascade of classifiers, which can reject most of the negative examples at early stages of processing, thereby significantly reducing computation time.



https://ieeexplore.ieee.org/document/990517

Viola-Jones: Another Sliding Window Approach



https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

Point Feature Matching

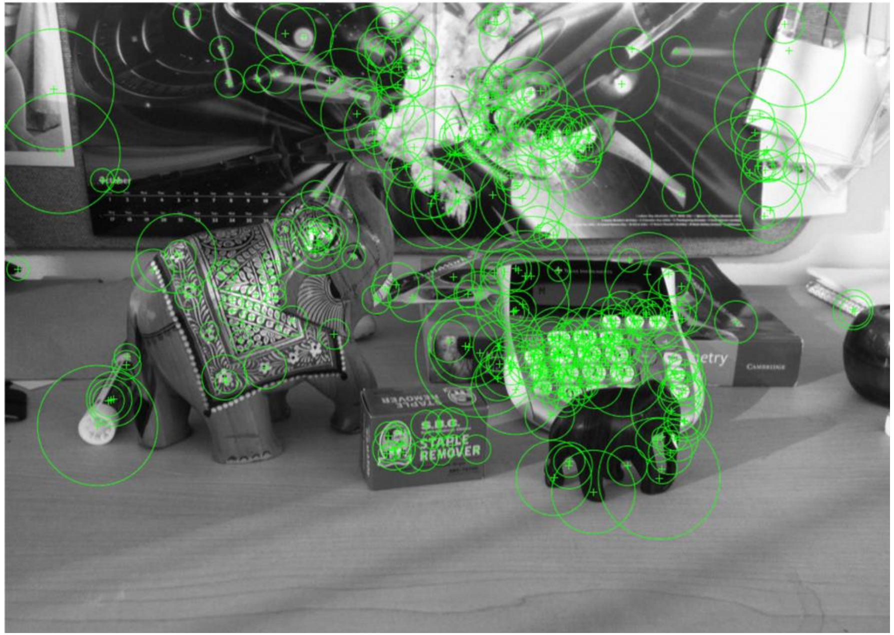
https://uk.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-matching.html

Point Feature Matching

- Harris corner detector
- Scale-Invariant Feature Transform (SIFT)
- Speeded Up Robust Features (SURF)

100 Strongest Point Features from Box Image

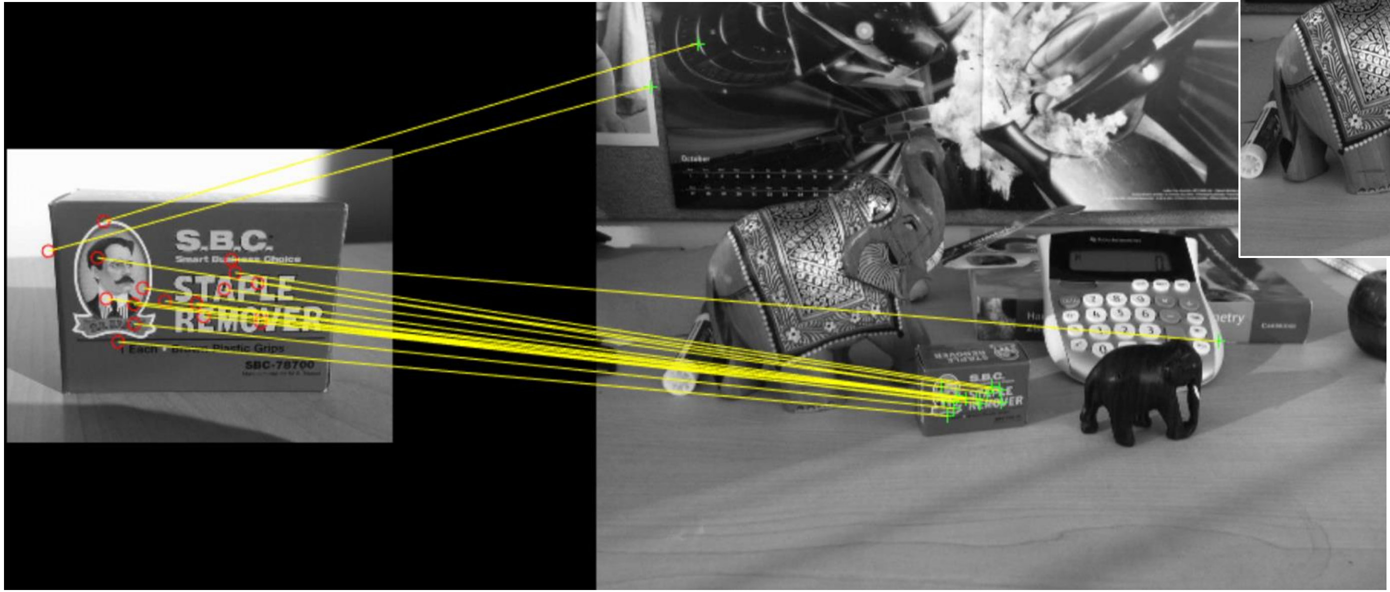
300 Strongest Point Features from Scene Image



https://uk.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-matching.html

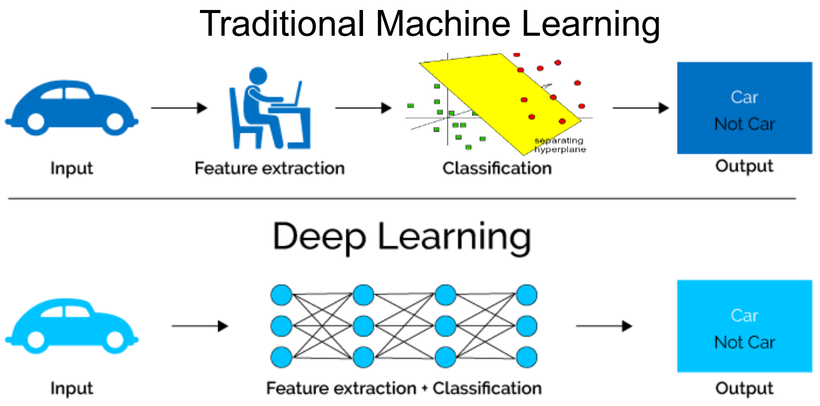
Point Feature Matching

- Rank feature similarities
- Random sample consensus (RANSAC) algorithm

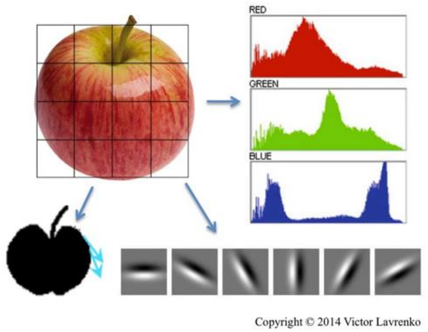


https://uk.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-matching.html

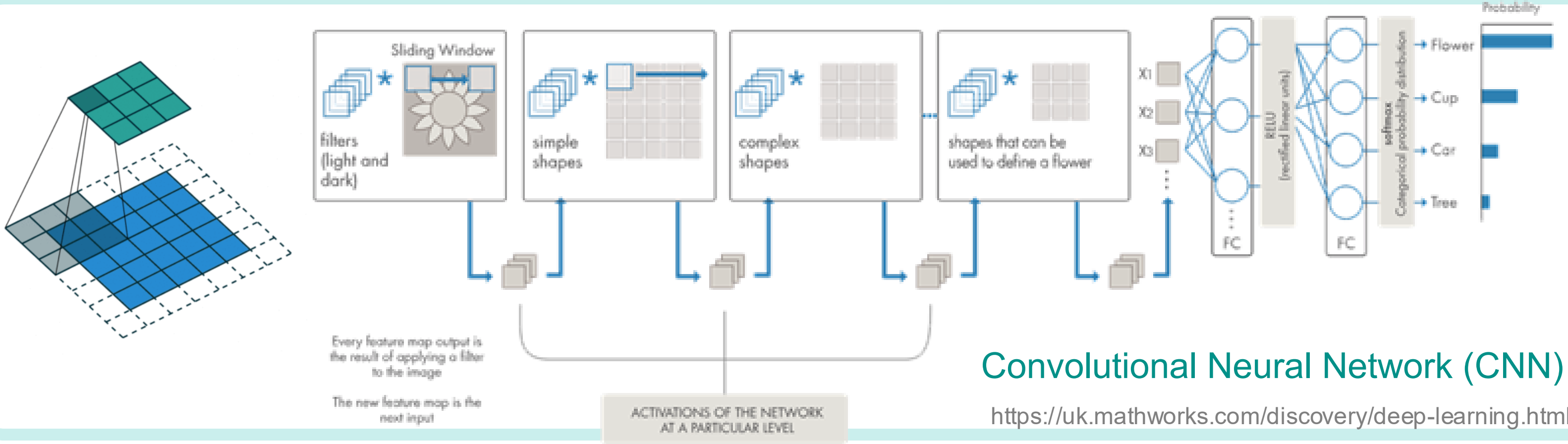
Next year: Deep learning-based object detectors



Feature extraction



X_{apple}= {$mean_{red}$, $variance_{red}$, $mean_{green}$, $variance_{green}$, $mean_{blue}$, $variance_{blue}$, orientation, solidity, texture, … }

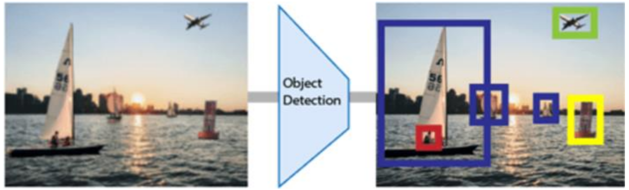


Convolutional Neural Network (CNN)

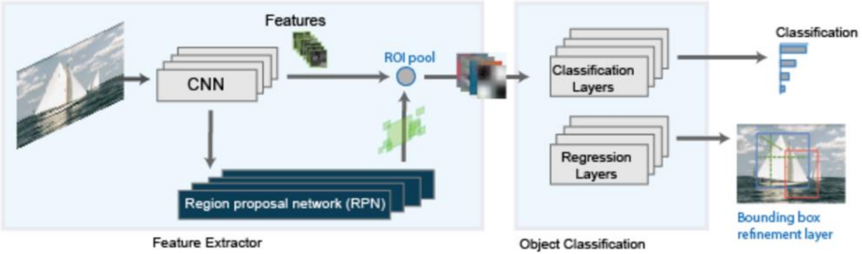
https://uk.mathworks.com/discovery/deep-learning.html

Next year: Deep learning-based object detectors

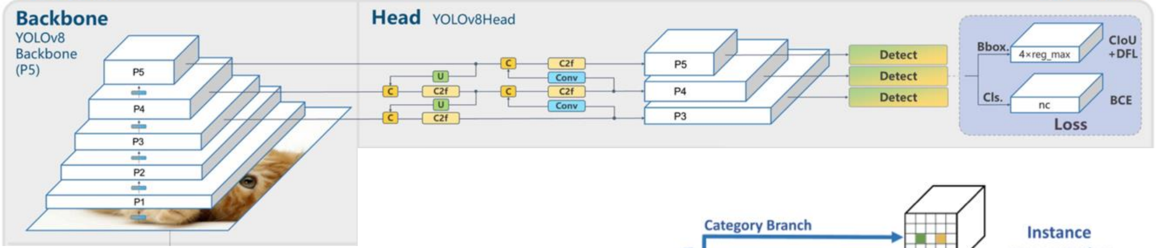
Objective

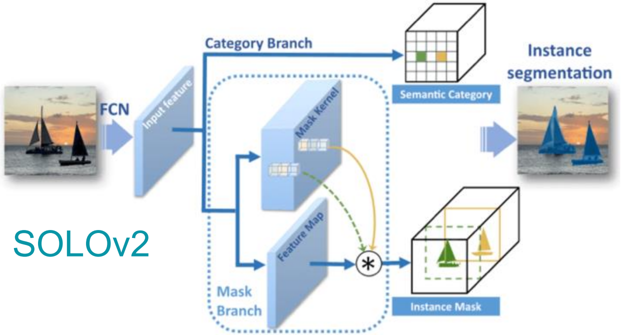


Faster-RCNN

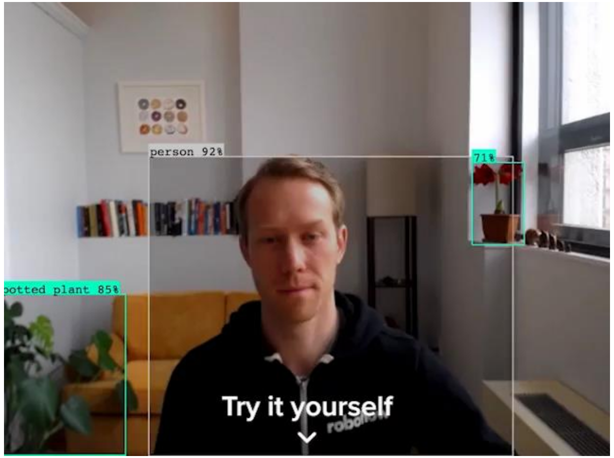


YOLOv8 [YOLOv5](), [YOLOv6](), [YOLOX]()





SOLOv2



https://yolov8.com/

Next Lecture

Object Detection:

Viola-Jones Detector