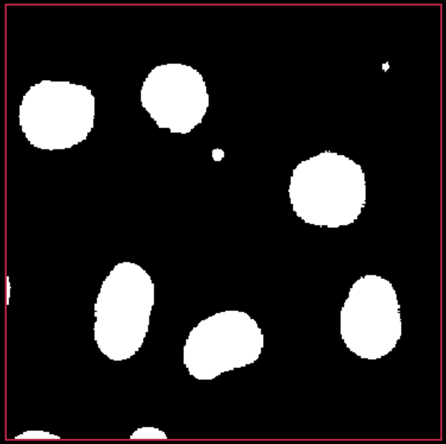
Image Analysis

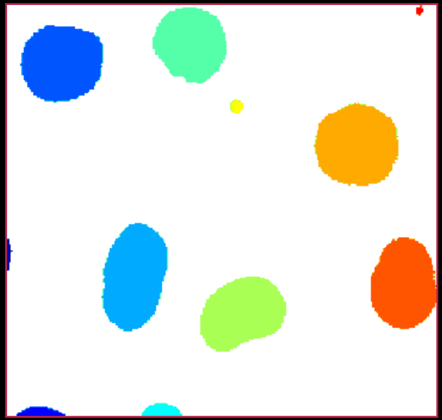
Rasmus R. Paulsen
Tim B. Dyrby
DTU Compute

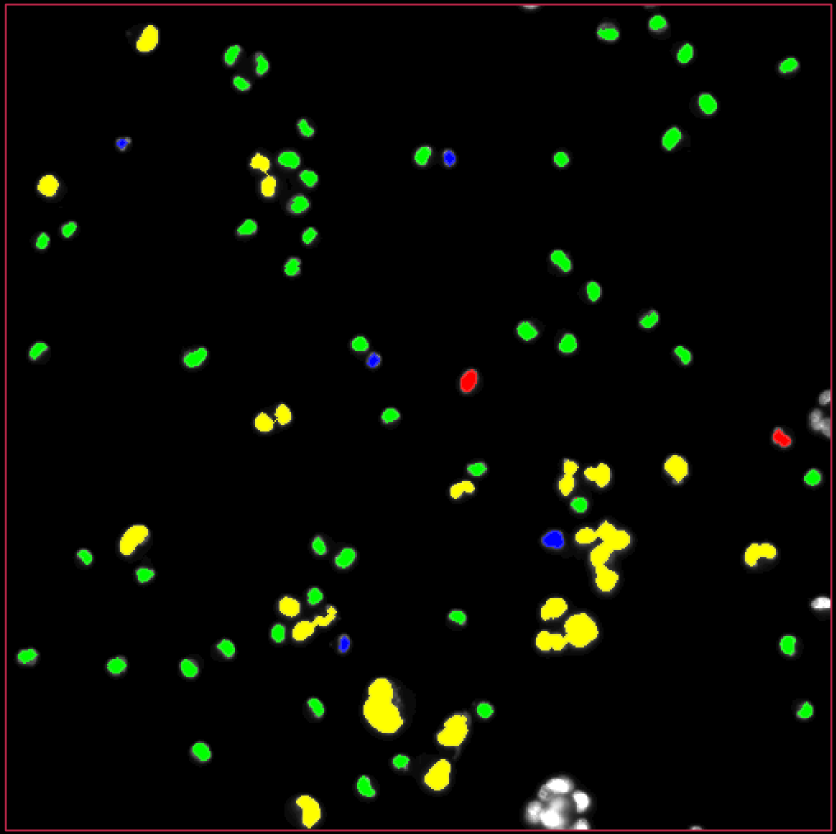
rapa@dtu.dk

http://www.compute.dtu.dk/courses/02502

Lecture 6 – BLOB analysis and feature based classification





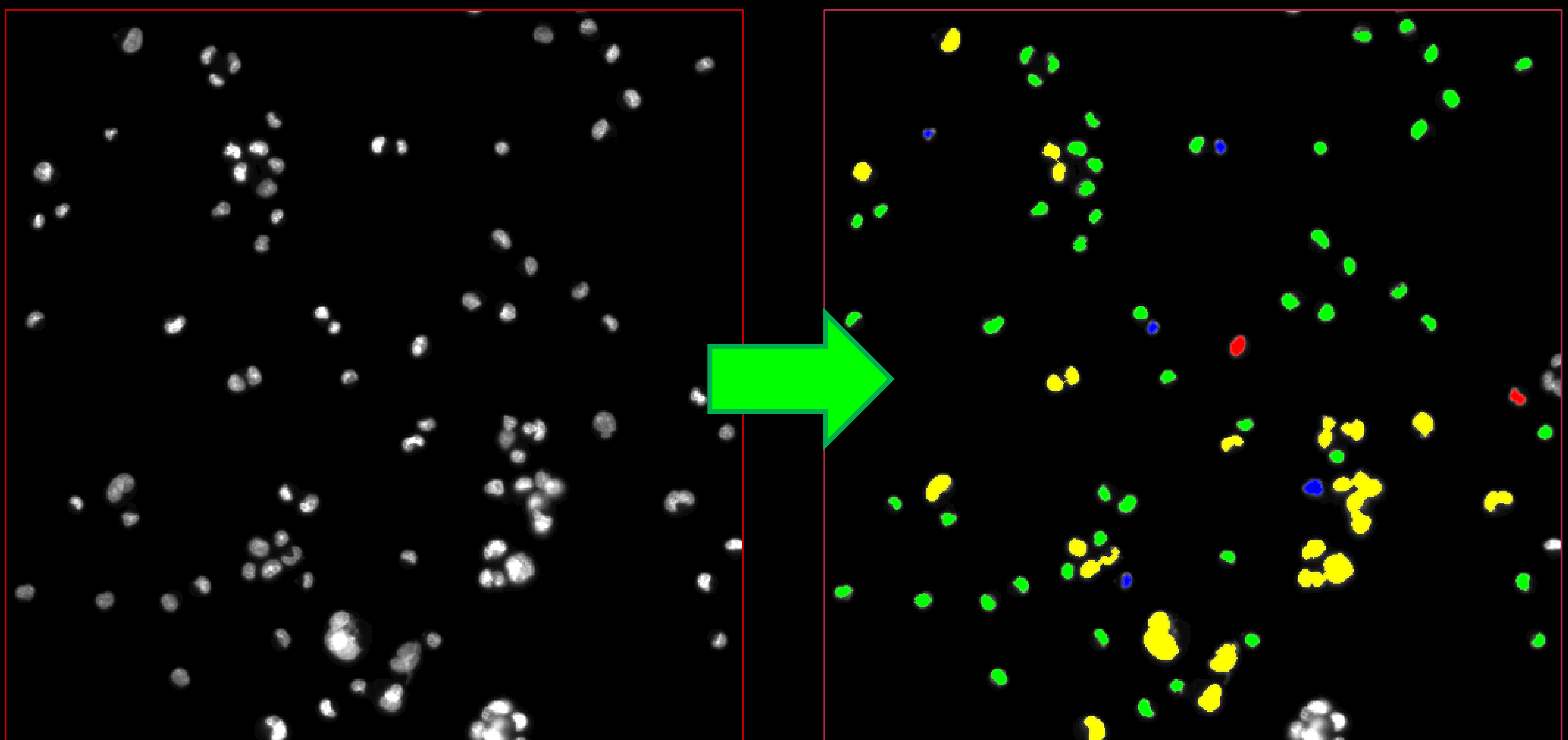


What can you do after today?

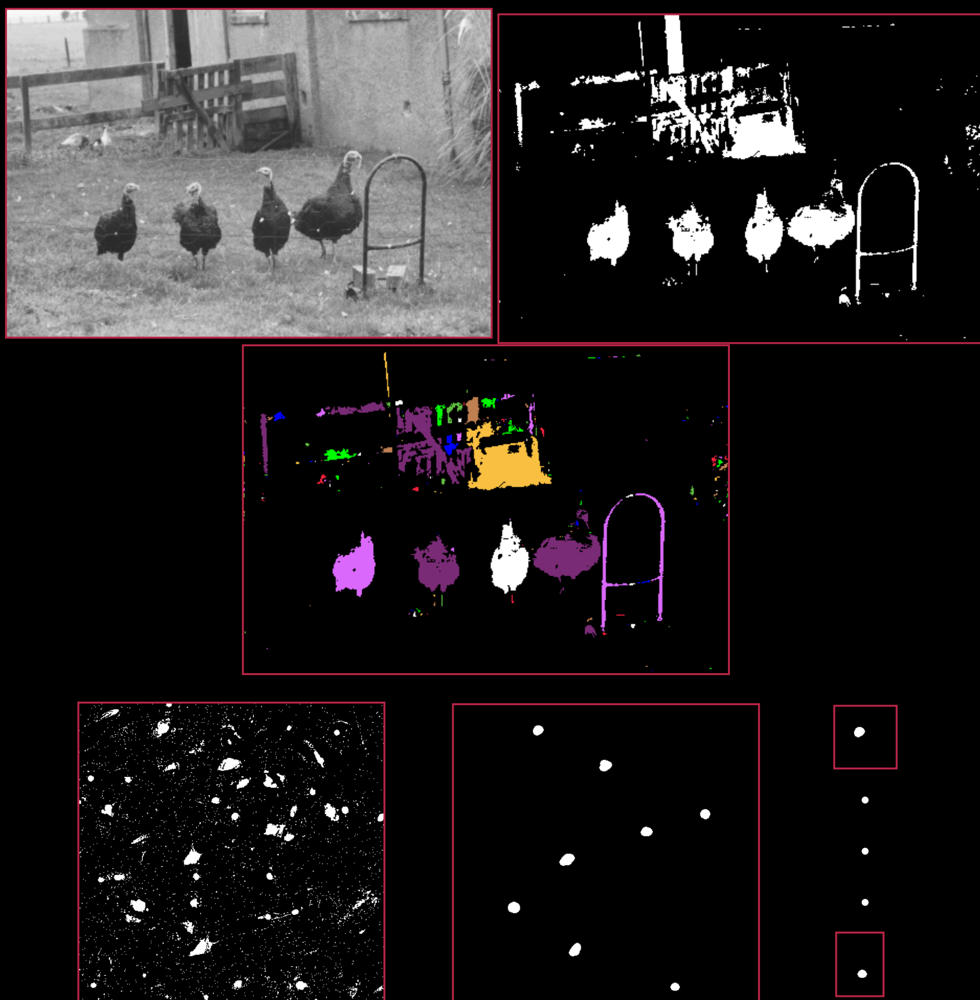
- Calculate the connected components of a binary image. Both using 4-connected and 8-connected neighbours
- Compute BLOB features including area, bounding box ratio, perimeter, center of mass, circularity, and compactness
- Describe a feature space
- Compute blob feature distances in feature space
- Classify binary objects based on their blob features
- Estimate feature value ranges using annotated training data
- Compute a confusion matrix
- Compute rates from a confusion matrix including sensitivity, specificity and accuracy
- Determine and discuss what is the importance of sensitivity and specificity given an image analysis problem

Object recognition

- Recognise objects in images
- Put them into different classes

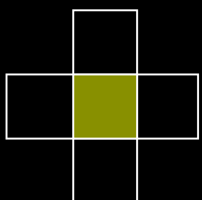


BLOB – what is it?

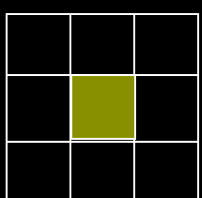


- BLOB = Binary Large Object
  - Group of connected pixels
- BLOB Analysis
  - *Connected component analysis*
  - *Object labelling*

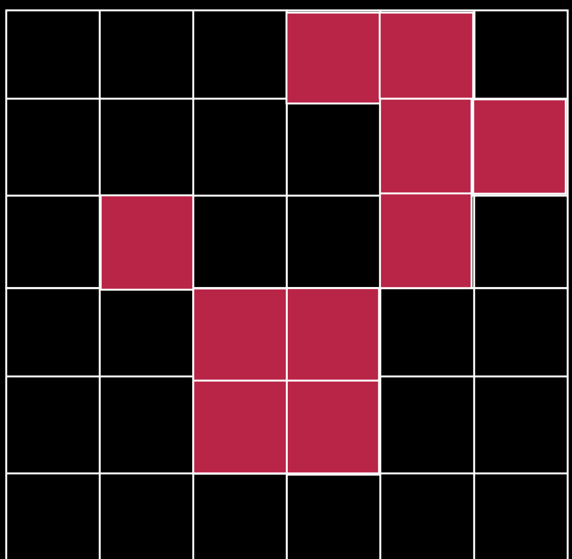
Isolating a BLOB



4-connected



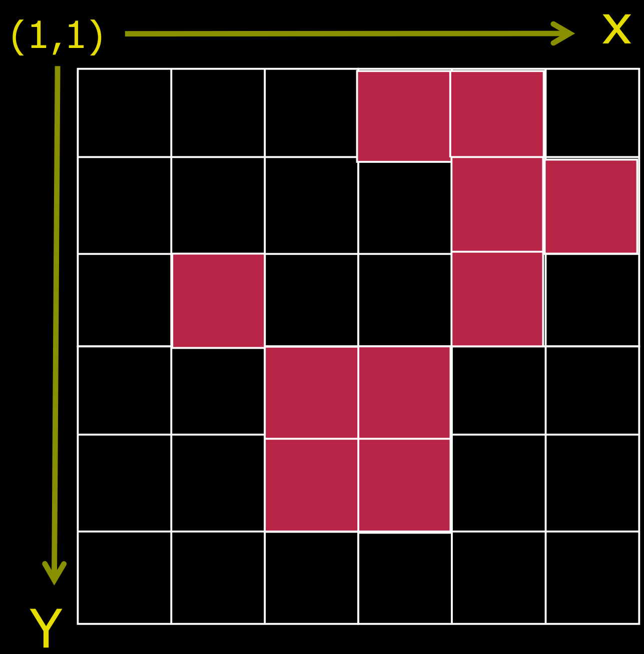
8-connected



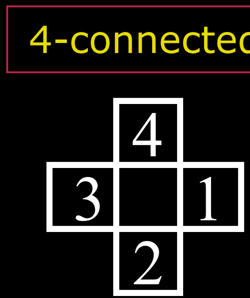
Image

- What we want:
  - For each object in the image, a list with its pixels
- How do we get that?
  - Connected component analysis
- Connectivity
  - Who are my neighbors?
  - 4-connected
  - 8-connected

Connected component analysis



- Binary image
- Seed point: where do we start?
- *Grassfire* concept
  - Delete (burn) the pixels we visit
  - Visit all *connected* (4 or 8) neighbors



BLOBs 4- and 8-neighbours

A BLOB analysis is performed using both 4- and 8-connectivity. How many BLOBS are found using the two different connectivities?

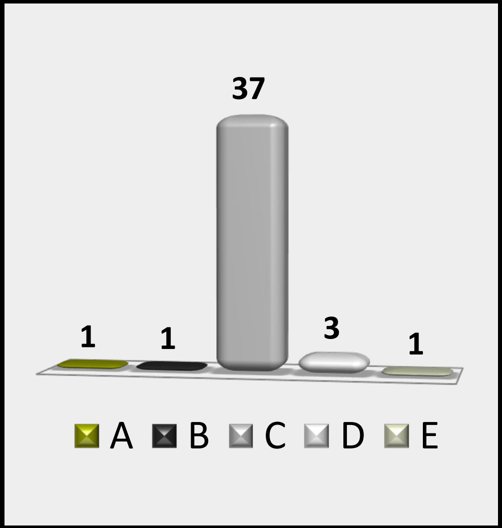
A) 3 and 7

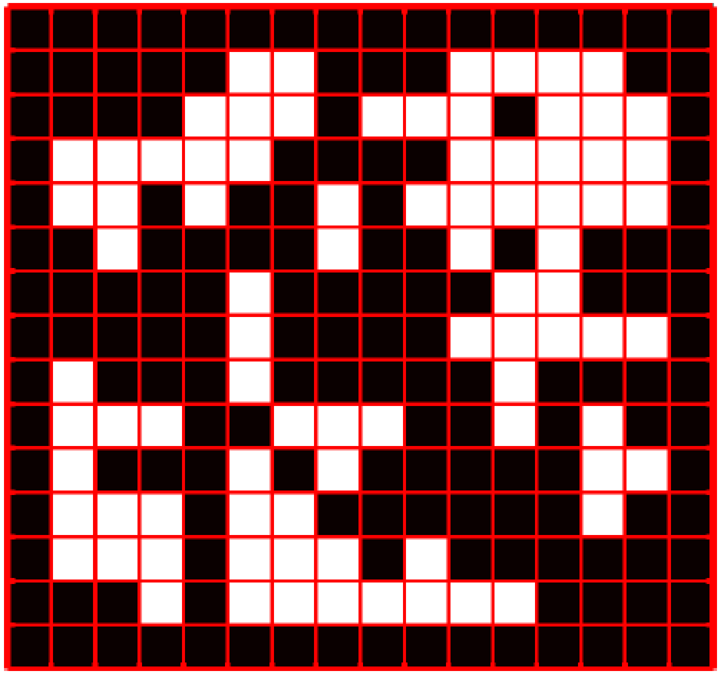
B) 9 and 5

C) 8 and 6

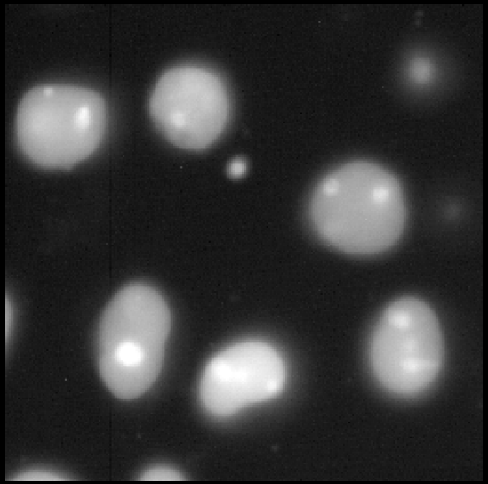
D) 7 and 5

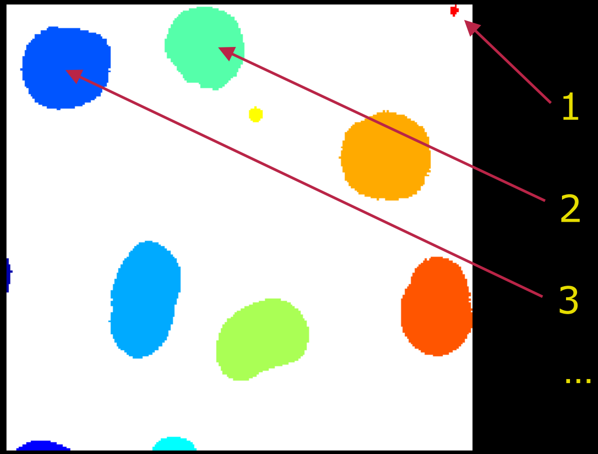
E) 4 and 5





The result of connected component analysis





- An image where each BLOB (component) is labelled
- Each blob now has a unique ID number
- What do we do with these blobs?

Features

- Feature
  - A prominent or distinctive aspect, quality, or characteristic
  - *This radio has many good features*
- Car (Ford-T) features
  - 4 wheels
  - 2 doors
  - 540 kg
  - 20 hp

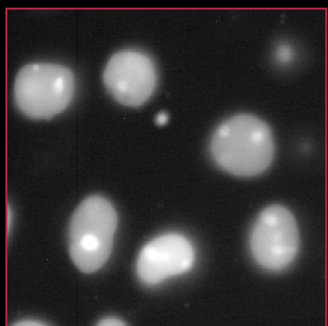
Feature vector

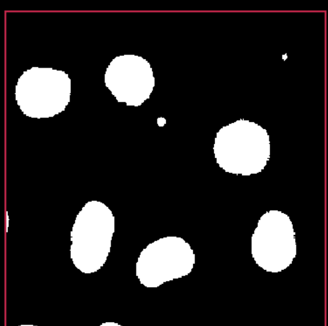
f=[4, 2, 540, 20]

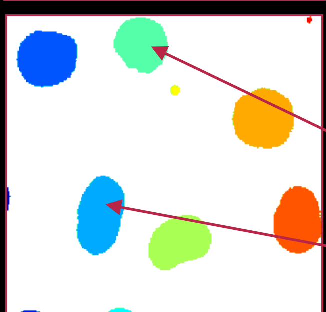
f=[4, 3, 1100, 90]

- Feature vector
  - Vector with all the features for one object
- Ford-T features
  - 4 wheels
  - 2 doors
  - 540 kg
  - 20 hp
- Ford Fiesta features
  - 4 wheels
  - 3 doors
  - 1100 kg
  - 90 hp

Feature extractions





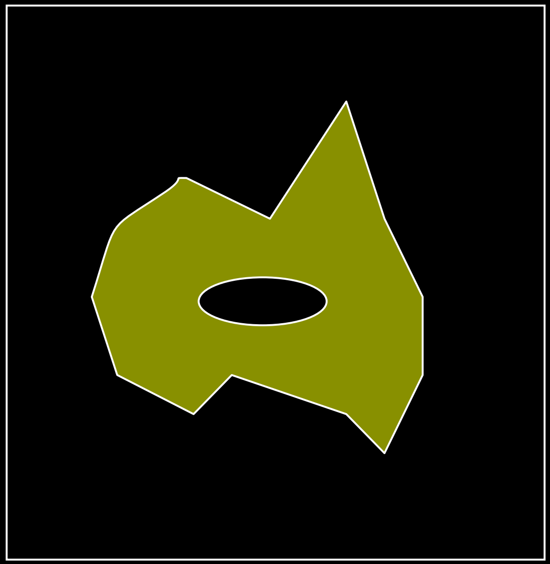


- Compute features for each BLOB that can be used to identify it
  - Size
  - Shape
  - Position
- From image operations to mathematical operations
  - Input: a list of pixel positions
  - Output: Feature vector
- First step: remove invalid BLOBS
  - too small or big- using morphological operations for example
  - border BLOBs

Feature vector = [2,1,…,3]

Feature vector = [4,7,…,0]

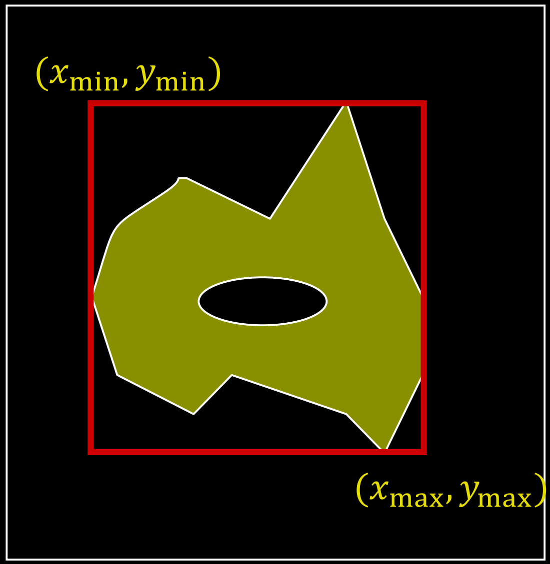
BLOB Features



One BLOB

- Area
  - number of pixels in the BLOB
  - Can be used to remove noise (small BLOBS)

BLOB Features



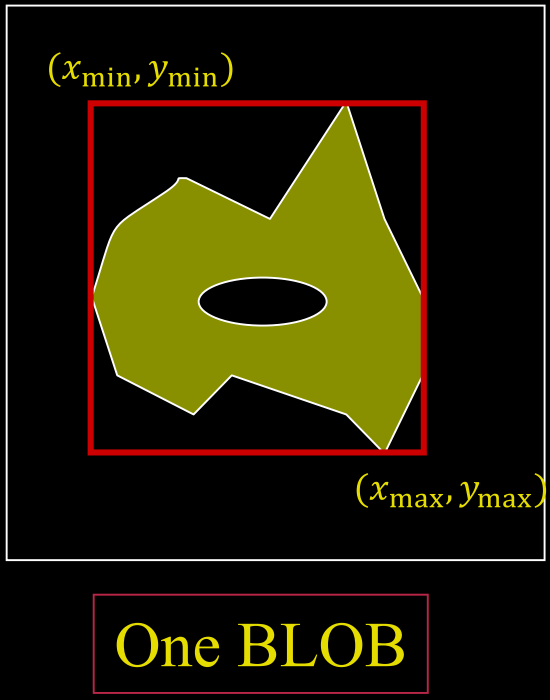
One BLOB

- Bounding box
  - Minimum rectangle that contains the BLOB
  - Height: $y_{\max} - y_{\min}$
  - Width: $x_{\max} - x_{\min}$
  - Bounding box ratio:

$$\frac{y_{\max} - y_{\min}}{x_{\max} - x_{\min}}$$

  - tells if the BLOB is elongated

BLOB Features



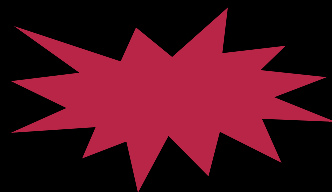
- Bounding box
  - Bounding box area:

$$(y_{\max} - y_{\min}) \cdot (x_{\max} - x_{\min})$$

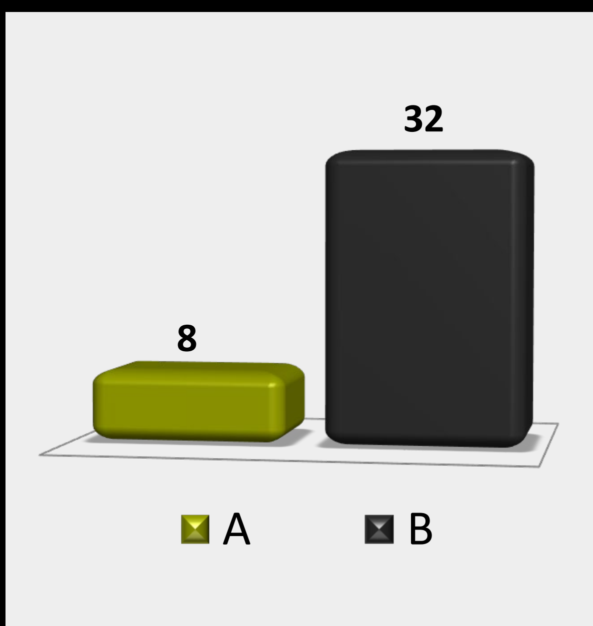
  - Compactness of BLOB

$$\text{Compactness} = \frac{\text{BLOB Area}}{(y_{\max} - y_{\min}) \cdot (x_{\max} - x_{\min})}$$

Compactness – which shape is most compact?

A) Shape 1: 

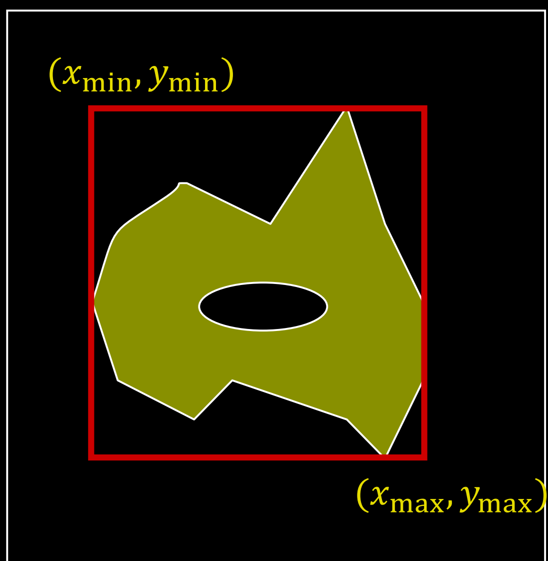
B) Shape 2: 



$$\text{Compactness} = \frac{\text{BLOB Area}}{(y_{\max} - y_{\min}) \cdot (x_{\max} - x_{\min})}$$

BLOB Features

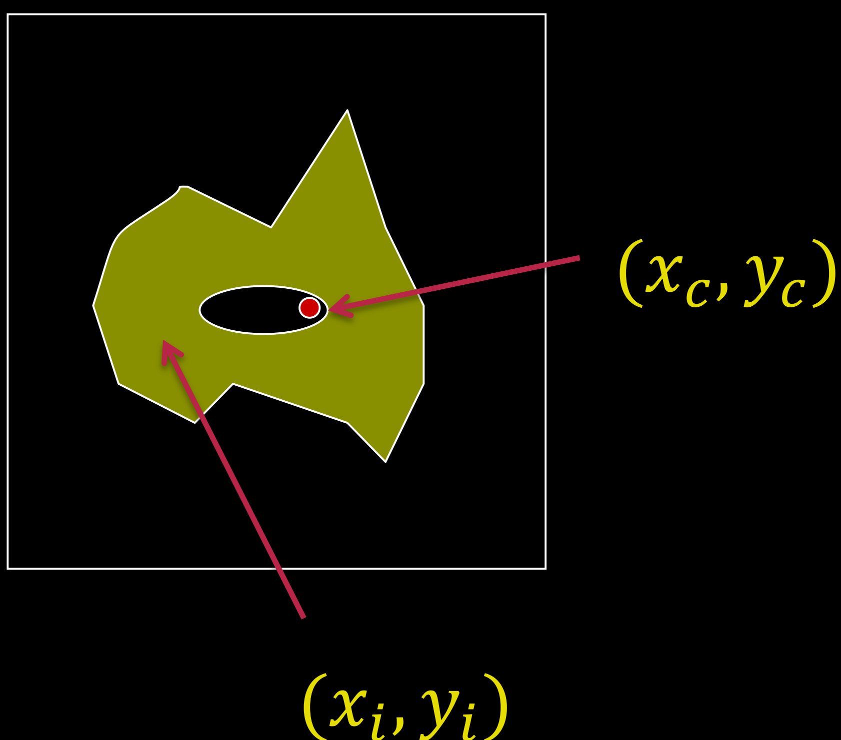
- Bounding box ratio
  - Bounding box height divided by the width



One BLOB

BLOB Features

- Center of mass (x_c, y_c)

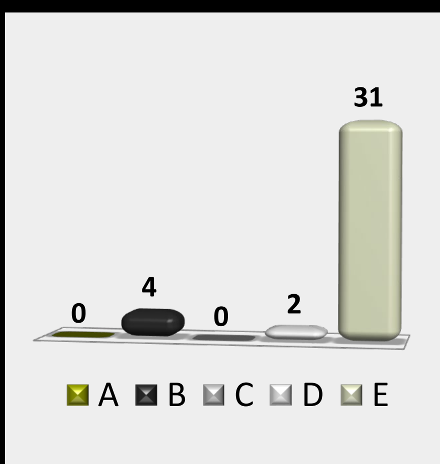


$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$y_c = \frac{1}{N}\sum_{i=1}^{N} y_i$$

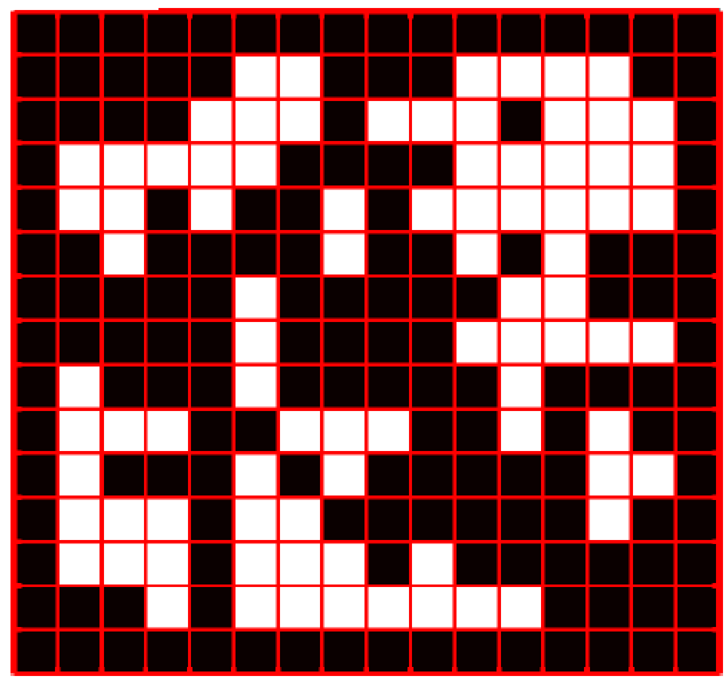
Center of mass

A) (12, 1.5)
B) (5, 8.5)
C) (6.5, 3.5)
D) (4.5, 0.5)
E) (7, 4.5)

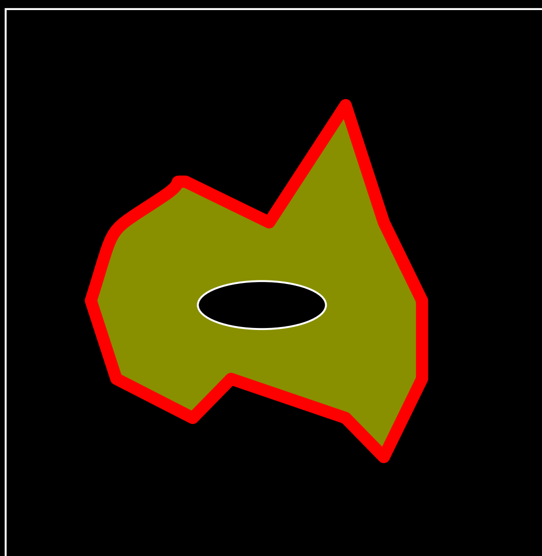


The smallest BLOB is found using 4-connectivity. What is the center of mass of this BLOB. The image has origin (0,0) and uses a (x,y) coordinate system.

1. $(12, 1.5)$
2. $(5, 8.5)$
3. $(6.5, 3.5)$
4. $(4.5, 0.5)$
5. $(7, 4.5)$



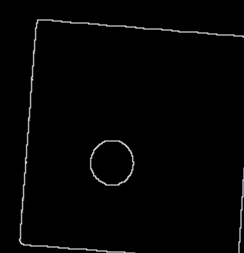
BLOB Features



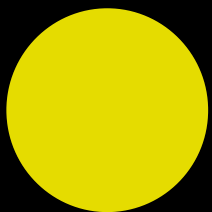
One BLOB

- Perimeter
  - Length of perimeter
  - How can we compute that?
- In practice (in Matlab) it is computed differently and more accurately

$$\sum ((f(x,y) \oplus \mathrm{SE}) - f(x,y))$$



BLOB Features - circularity

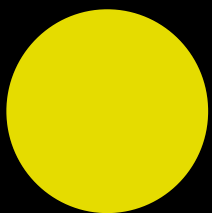


Circle like

Not circle like

- How much does it look like a circle?
- Circle
  - Area $A = \pi r^2$
  - Perimeter $P = 2\pi r$
- New object assumed to be a circle
  - Measured perimeter P_m
  - Measured area A_m
- Estimate perimeter from (measured) area
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$

BLOB Features - circularity



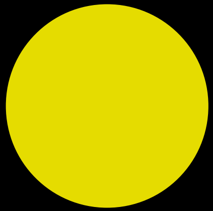
Circle like

Not circle like

- Compare the perimeters
  - Measured perimeter P_m
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$
- Circularity 1:

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

Circularity

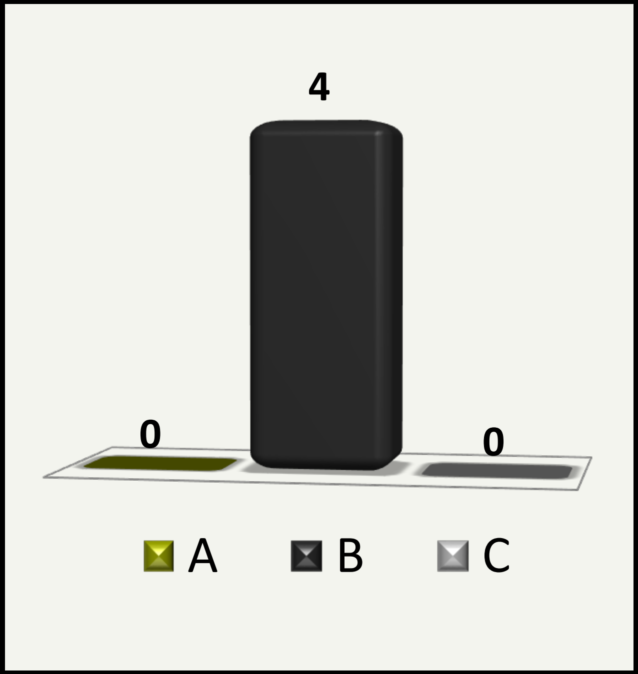


$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

A) $P_m < P_e$

B) $P_m = P_e$

C) $P_m > P_e$



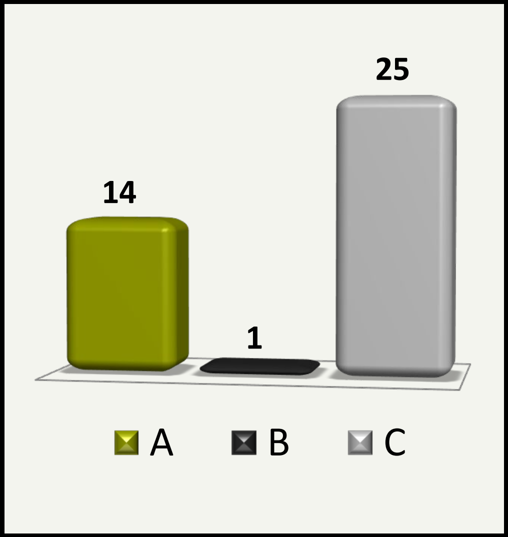
Circularity 2

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

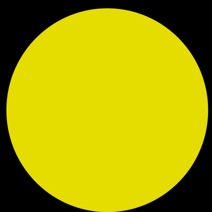
A) $P_m < P_e$

B) $P_m = P_e$

C) $P_m > P_e$



BLOB Features - circularity



Circle like

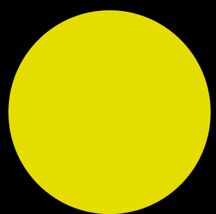
Not circle like

- Compare the perimeters
  - Measured perimeter P_m
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$
- Circularity:

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

- This measure will normally be ≥1

BLOB Features – circularity inverse



Circle like

Not circle like

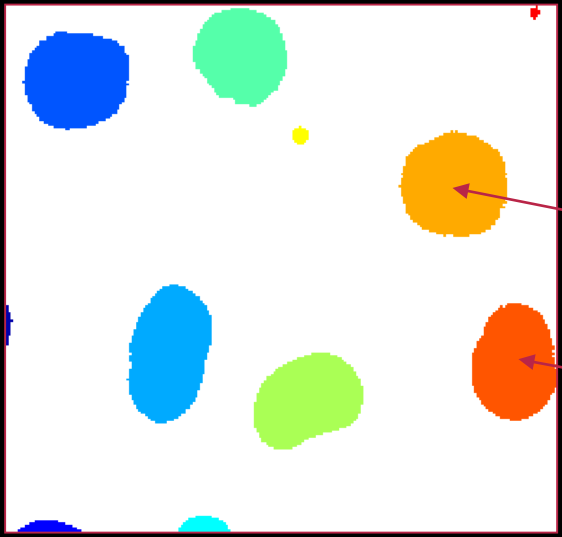
- Compare the perimeters
  - Measured perimeter P_m
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$
- Circularity (inverse):

$$\text{Circularity inverse} = \frac{P_e}{P_m} = \frac{2\sqrt{\pi \mathrm{A}_m}}{P_m}$$

- This measure will normally be ≤1

After feature extraction

Area, compactness, circularity etc calculated for all BLOB



Feature vector = [2,1,…,3]

Feature vector = [4,7,…,0]

One feature vector per blob

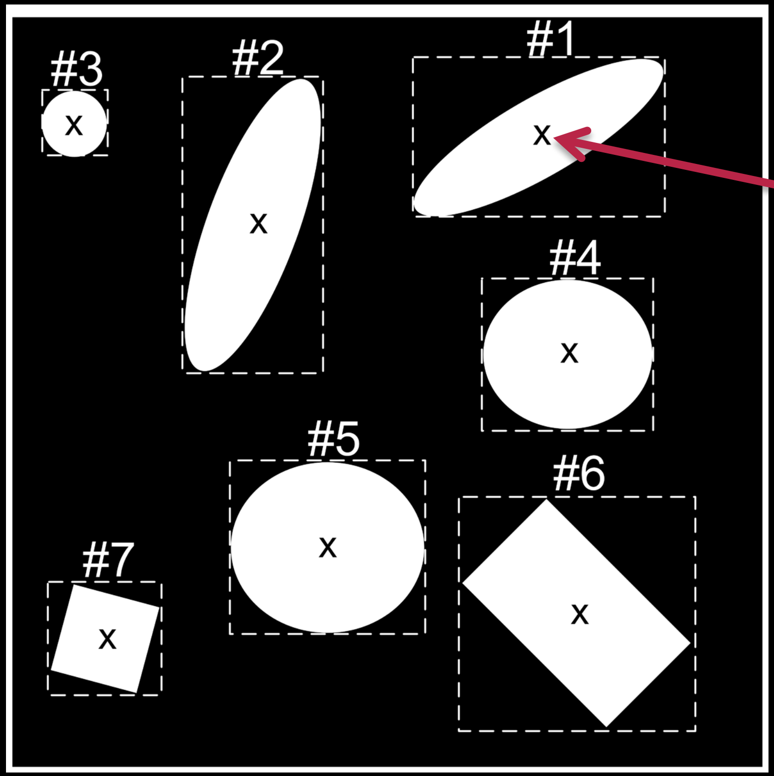
BLOB Classification

- Classification
  - Put a BLOB into a *class*

- *Classes* are normally pre-defined
  - *Car*
  - *Bus*
  - *Motorcycle*
  - *Scooter*

- Object recognition

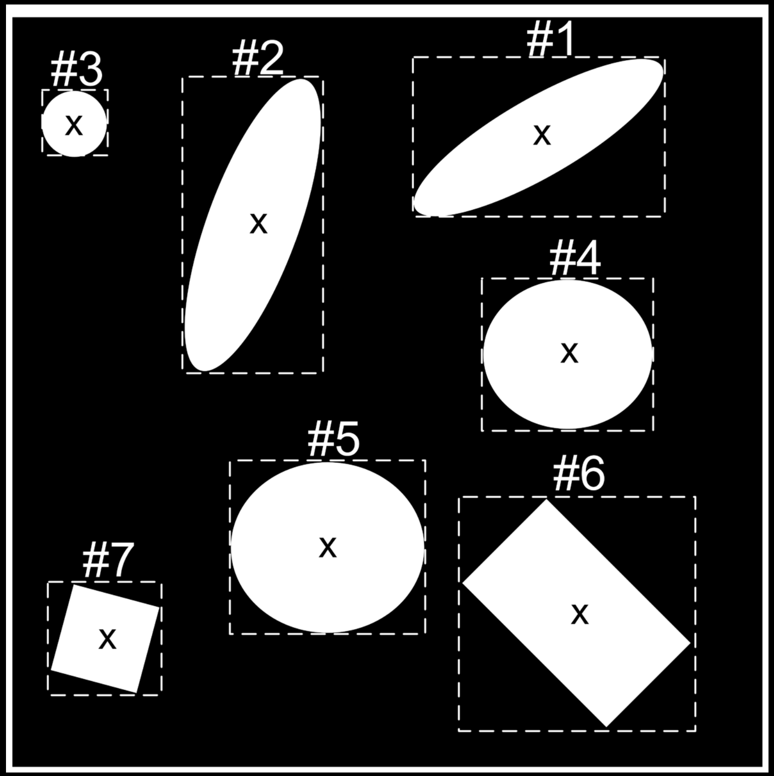
Object recognition: Circle example



BLOB number	Circu-larity	Area (pixels)
1	0.31	6561
2	0.40	6544
3	0.98	890
4	0.97	6607
5	0.99	6730
6	0.52	6611
7	0.75	2073

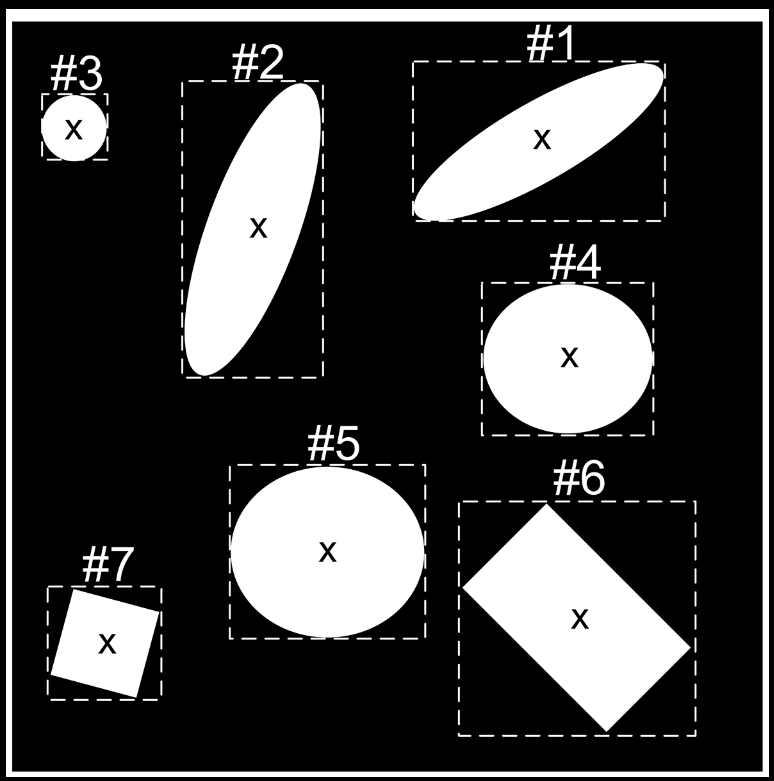
Which objects are circles?

Circle classification



- Two classes:
  - Circle
  - Not-circle
- Lets make a model of a *proto-type* circle

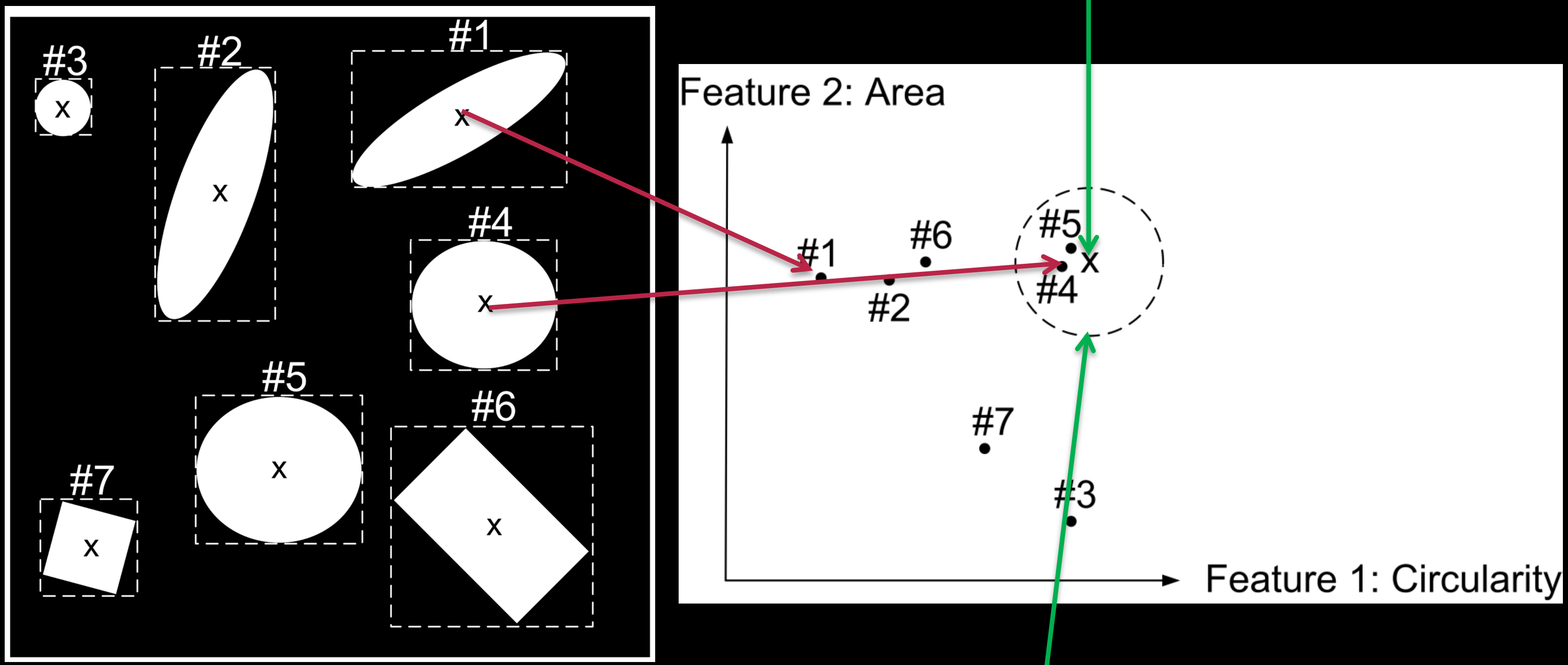
Circle classification



- Proto-type circle
  - Circularity : 1
  - Area: 6700

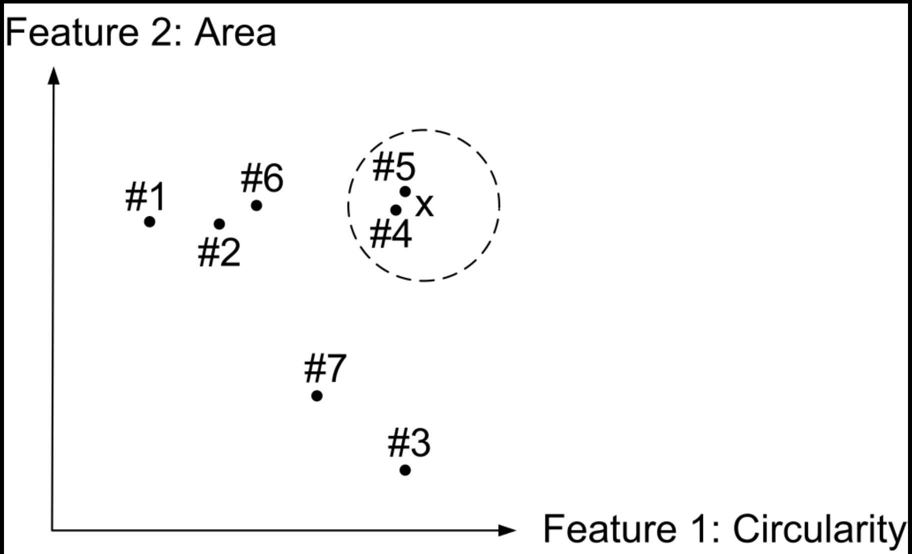
Feature Space

Proto-type circle



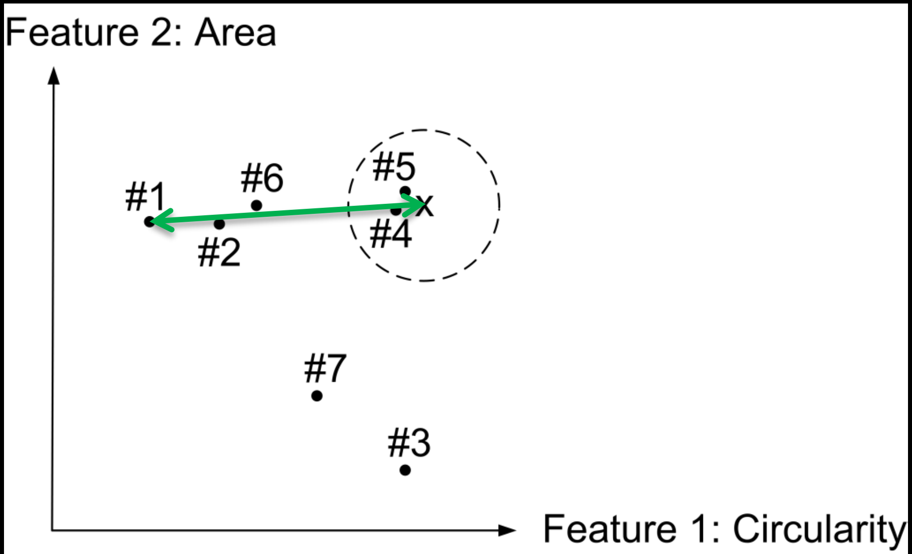
Objects in here are classified as circles

Feature space



- Proto-type circle
  - Circularity : 1
  - Area: 6700
- Some slack is added to allow non-perfect circles
  - Circularity: 1 +/- 0.15

Feature space - distances



- How do we decide if an object is inside the circle?
- Feature space distance
- Euclidean distance in features space

Blob 1: circularity: 0.31, Area : 6561

$$D = \sqrt{(0.31 - 1)^2 + (6561 - 6700)^2}$$

Dominates all! – normalisation needed

BLOB Feature Selection

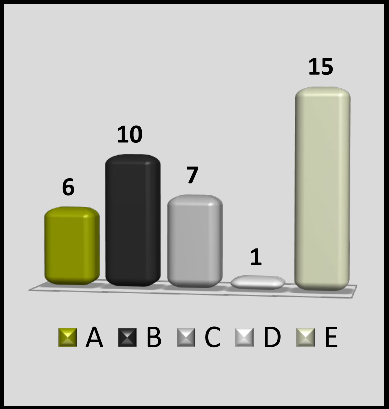
A) Compactness and circularity

B) Bounding box ratio and circularity

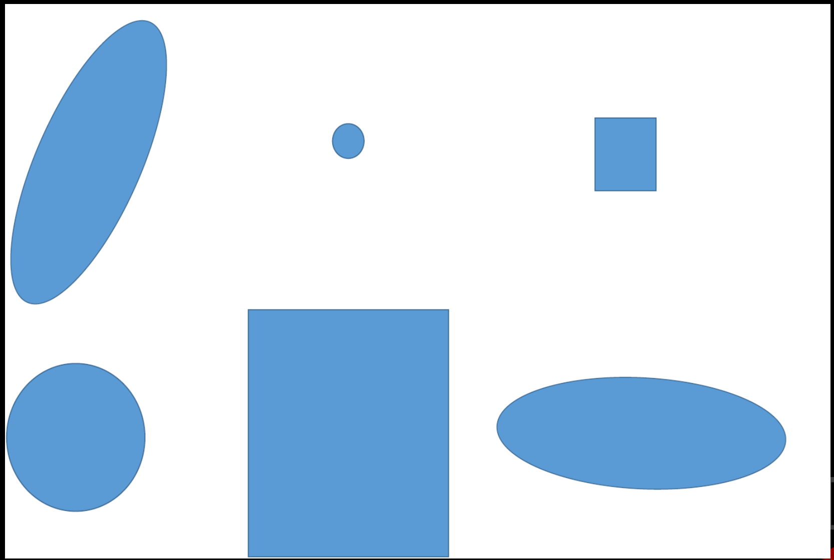
C) Area and compactness

D) Compactness and bounding box ratio

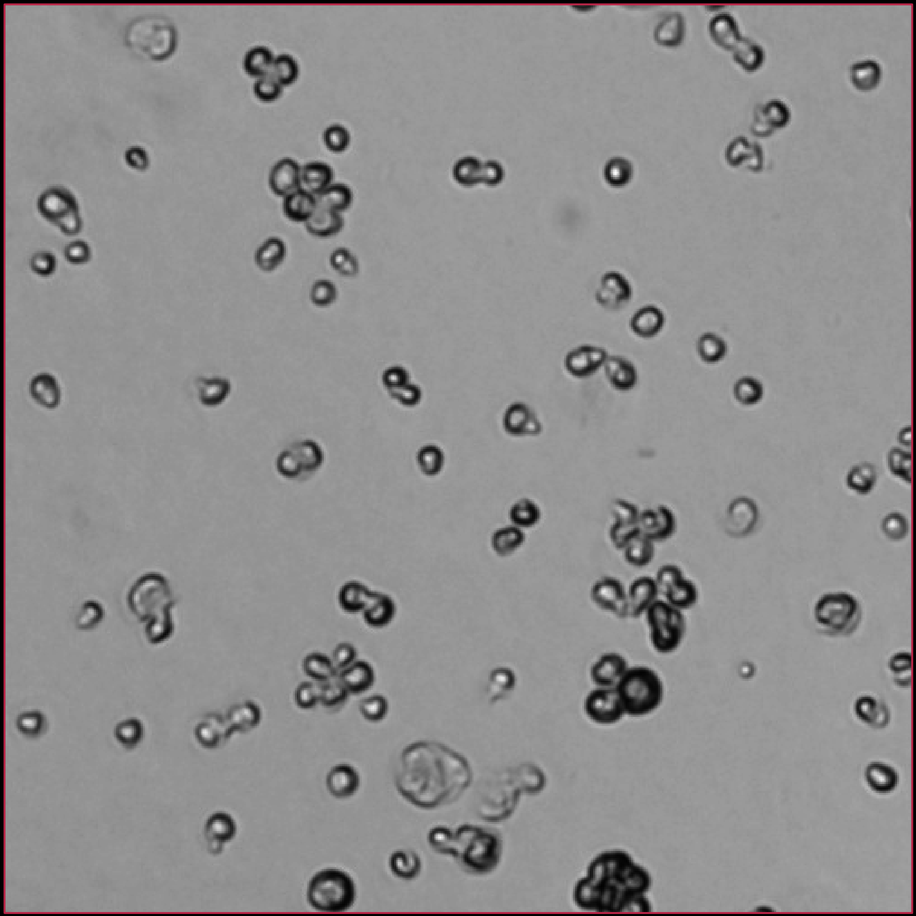
E) Area and bounding box ratio



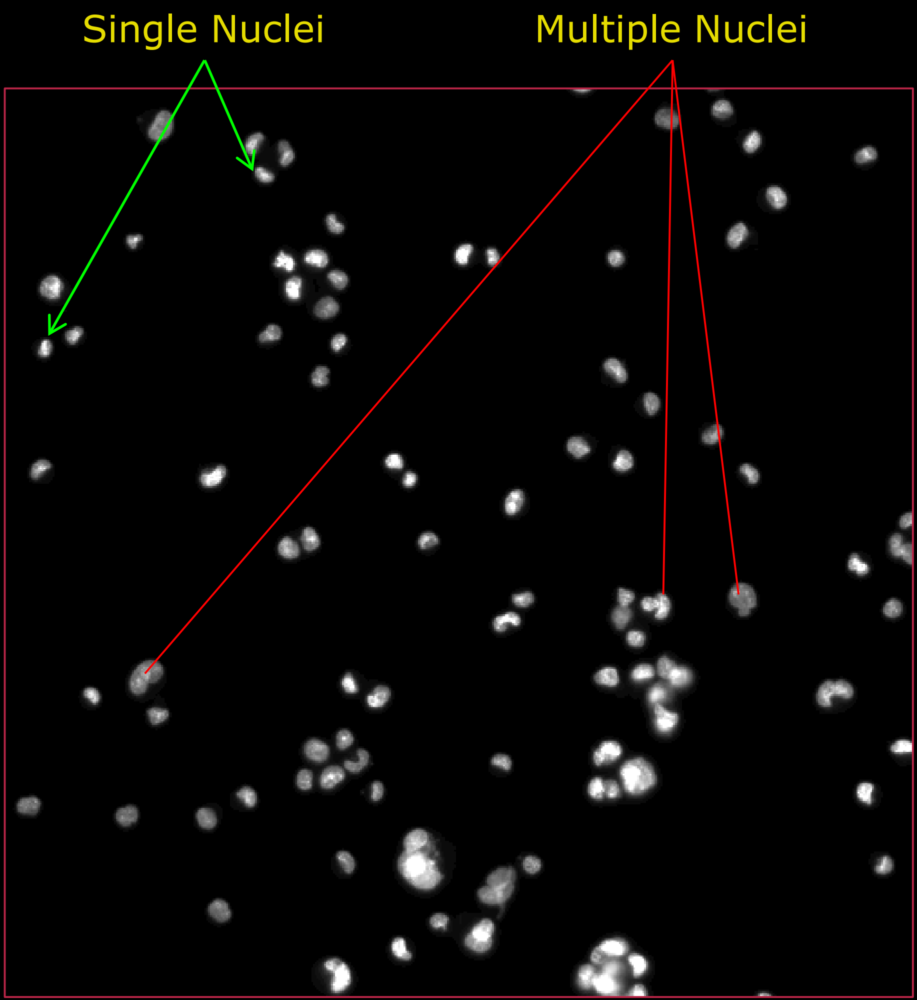
What set of BLOB features separate the below BLOBS best?



Cell classification



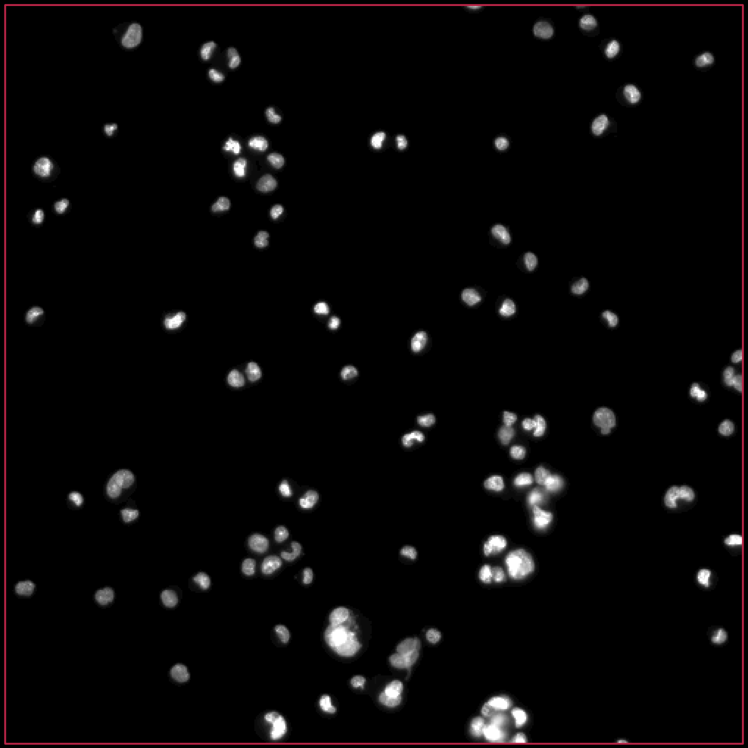
UV Microscopy



Fluorescence Microscopy (DAPI)

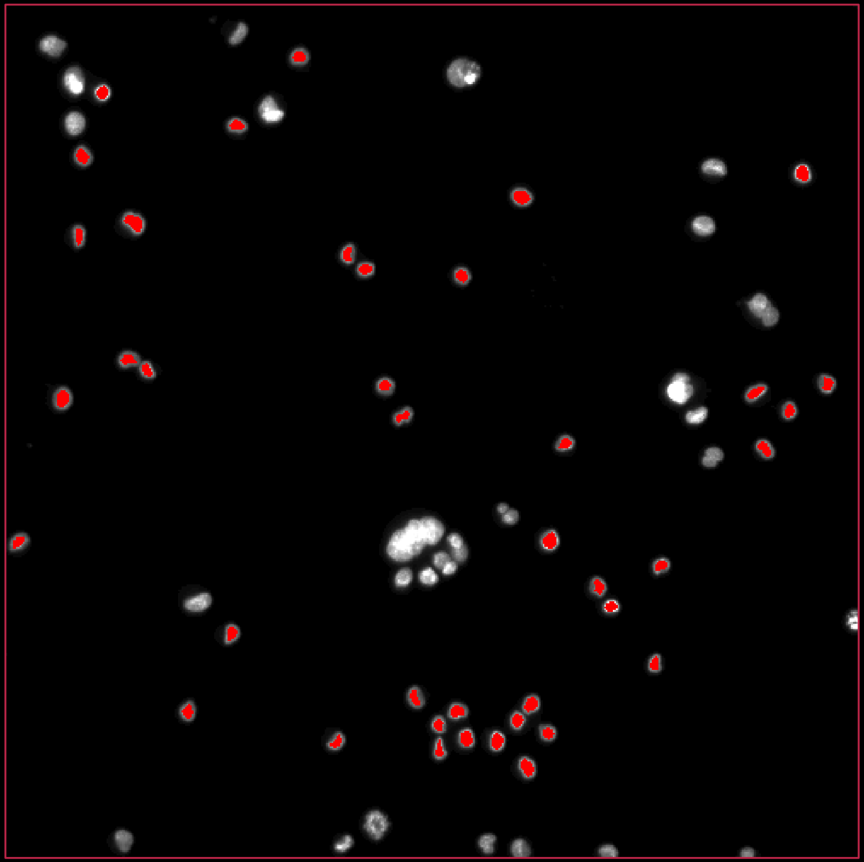
Images from ChemoMetec A/S

Nuclei classification



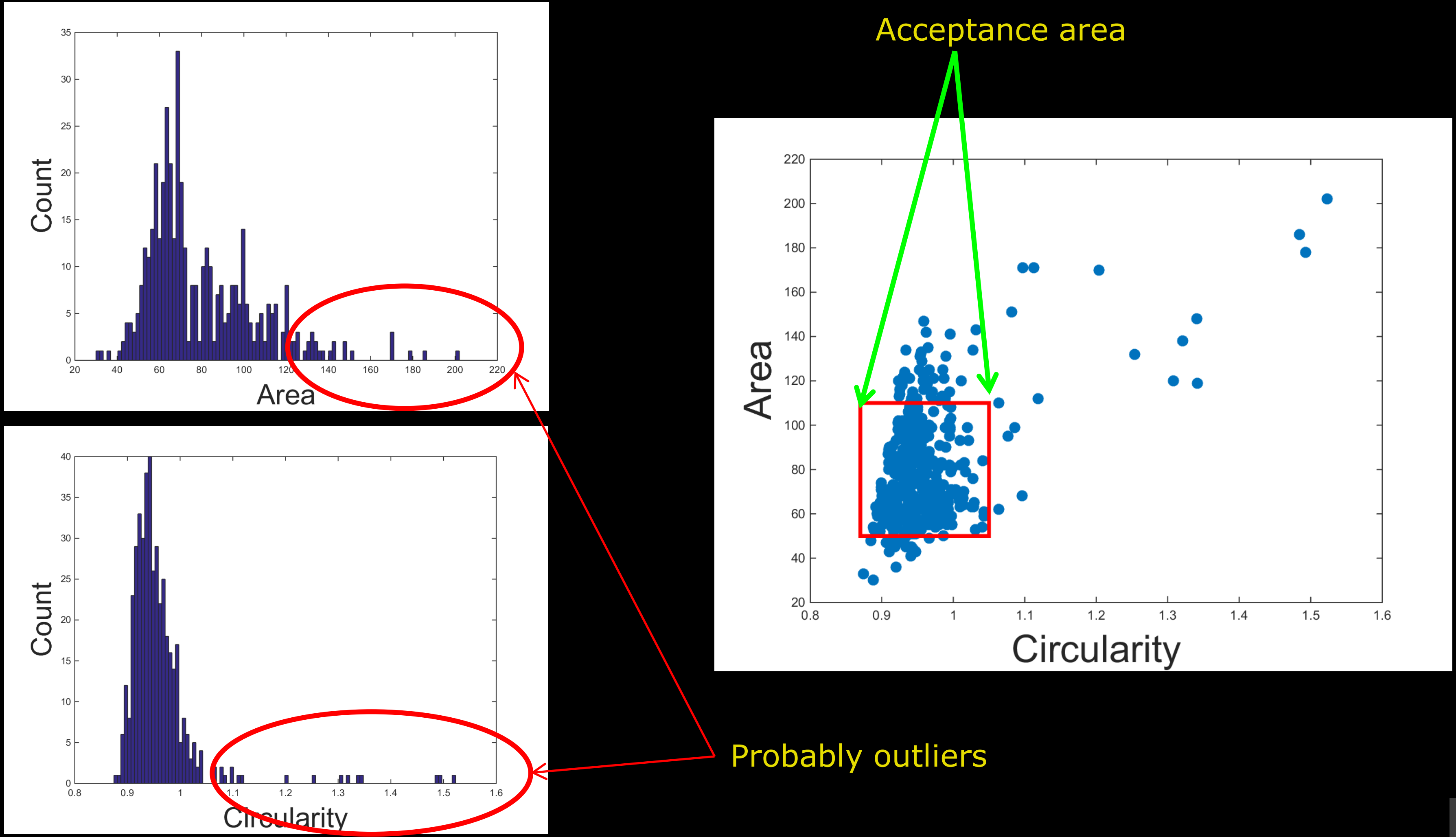
- DAPI image
- Two classes
  - Single nuclei
  - Noise
    - Multiple nuclei together
    - Debris
    - Other noise

Training and annotation

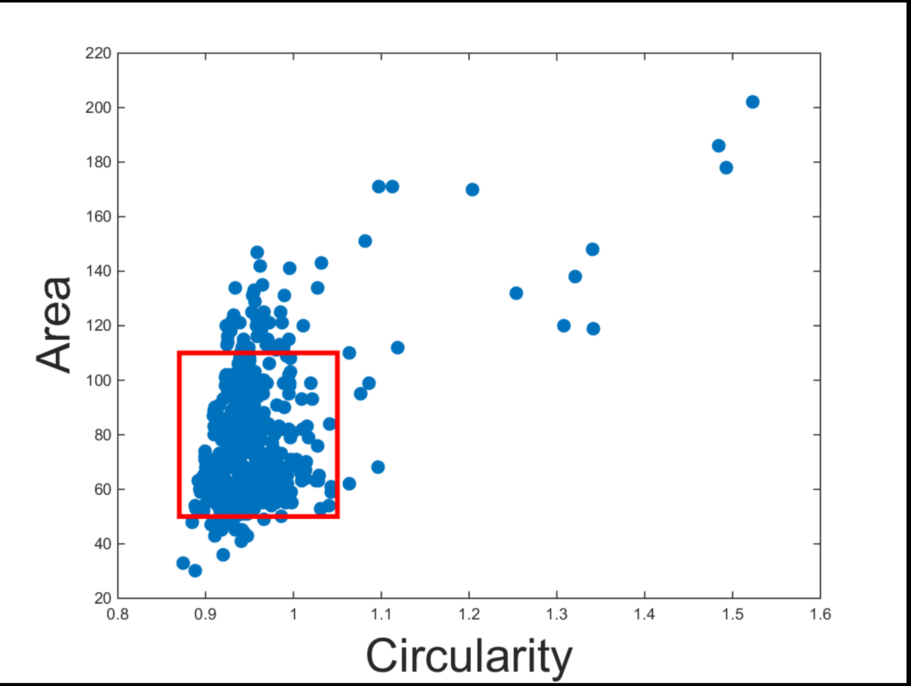


- Selection of true single nuclei marked
- Thresholding
- BLOB Analysis
  - Circularity
  - Area

Training data - analysis

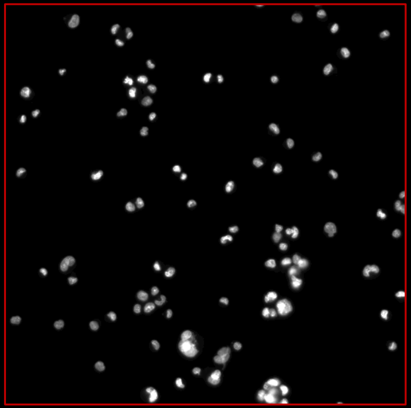


Feature ranges



Feature	Min	Max
Area	50	110
Circularity	0.87	1.05

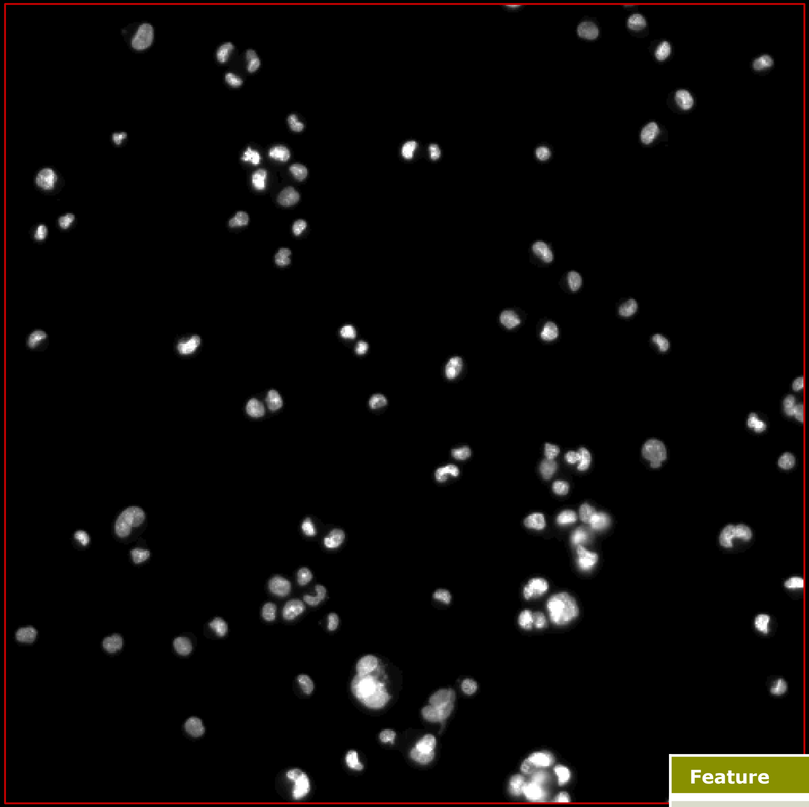
Using the classifier



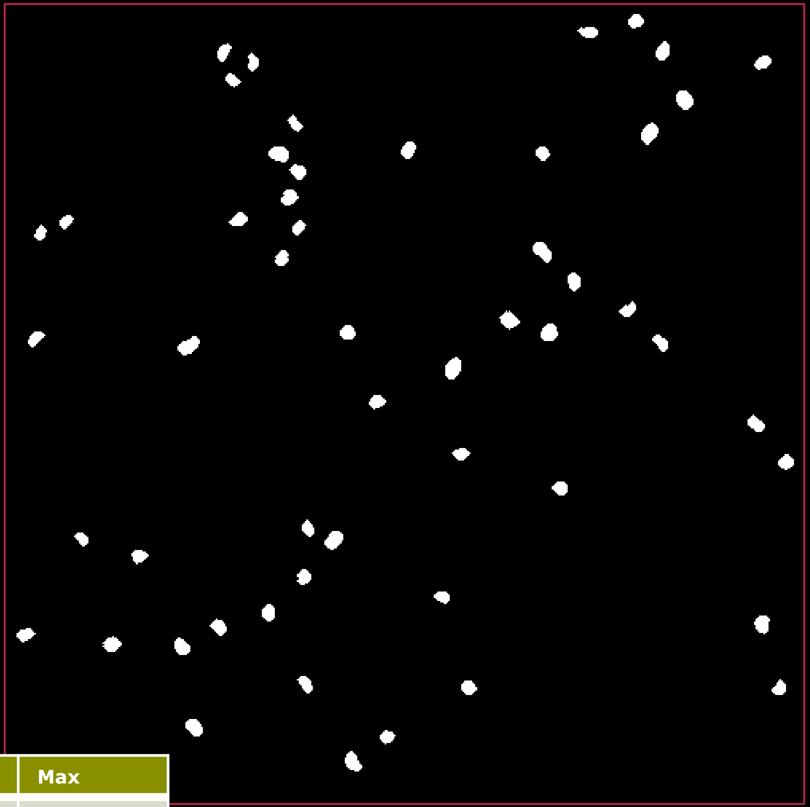
DAPI input image

- Threshold input image
- Morphological opening (SE 5x5)
- Morphological closing (SE 5x5)
- BLOBs found using 8-neighbours
- Border BLOBS removed
- Border features computed
  - Area + circularity
- BLOBs with features inside the acceptance range are single-nuclei

Using the classifier



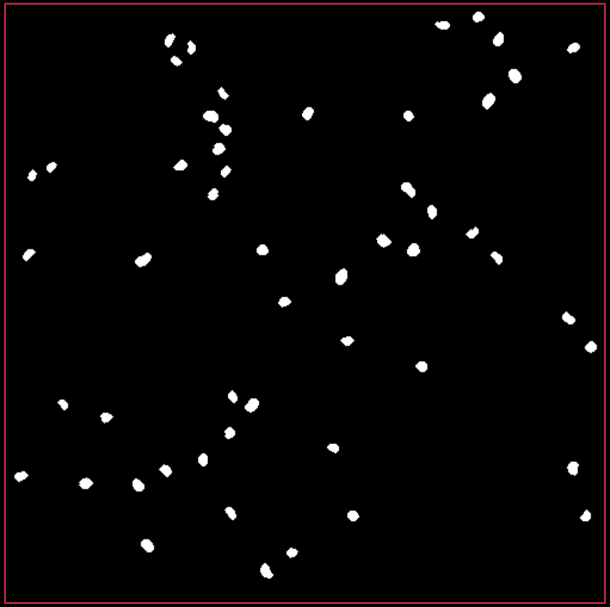
DAPI input image



Found single nuclei

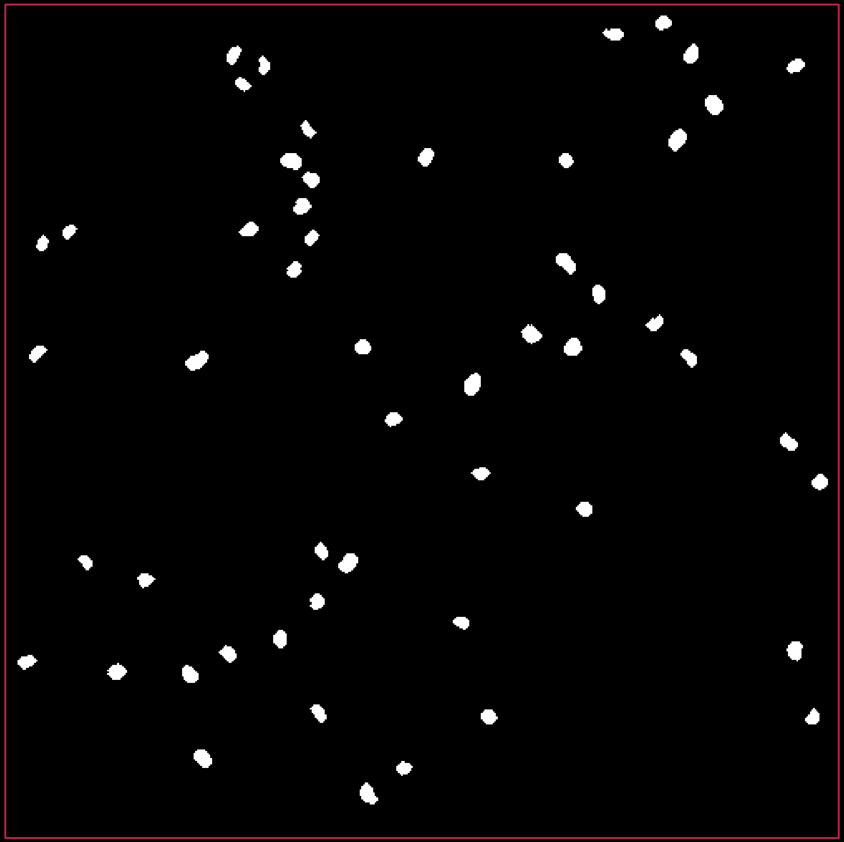
Feature	Min	Max
Area	50	110
Circularity	0.87	1.05

How well does it work?

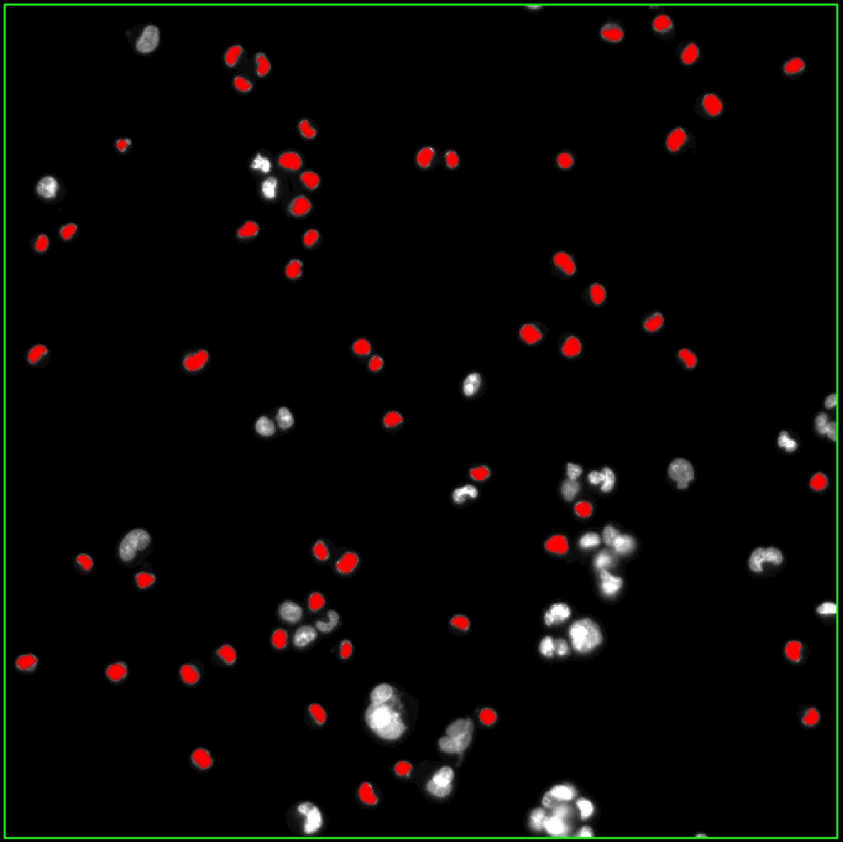


- We say we have a great algorithm!
- Strangely the doctor/biochemist do not trust this statement!
  - They need numbers!
- How do we report the performance?

Creating ground truth – expert annotations



Found single nuclei



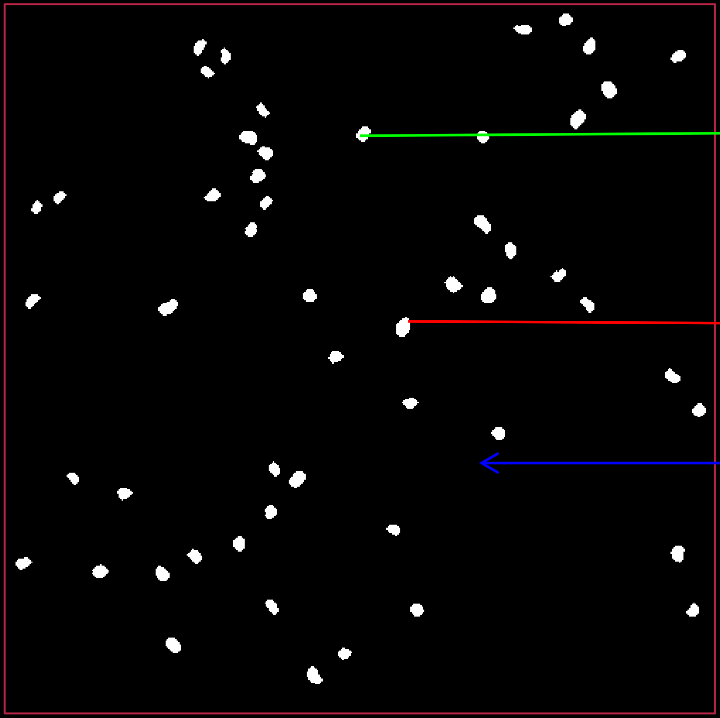
Expert opinion on true single nuclei

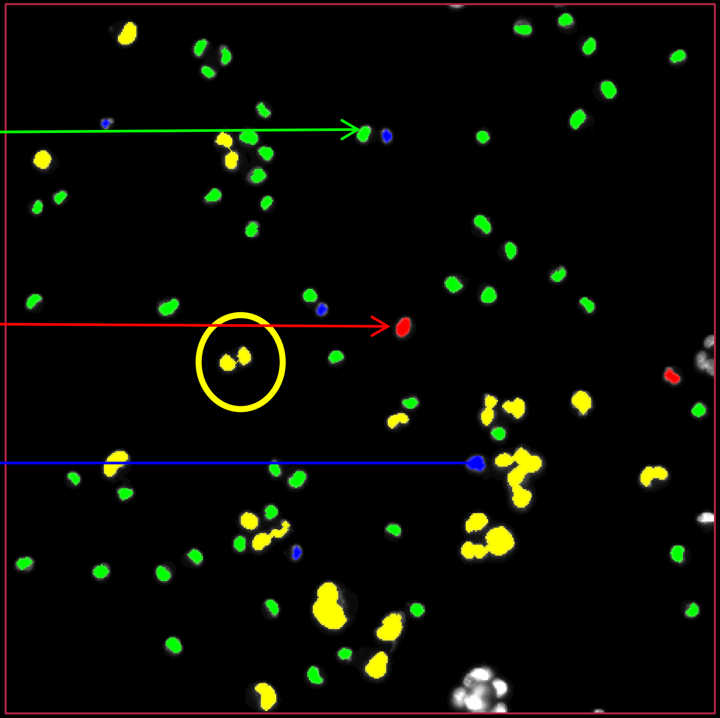
Red markings: Single nuclei

Not marked: Noise

Four cases

- True Positive (TP): A nuclei is classified as a nuclei
- True Negative (TN): A noise object is classified as noise object
- False Positive (FP): A noise object is classified as a nuclei
- False Negative (FN): A nuclei is classified as a noise object

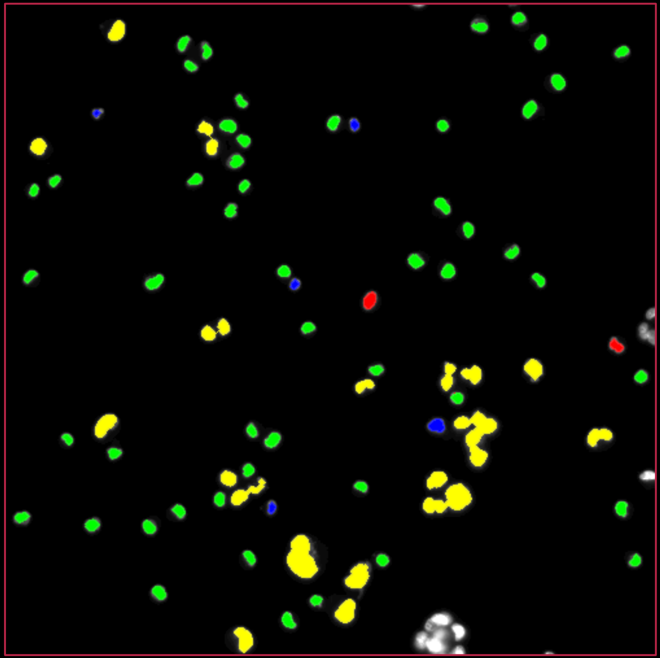




Found single nuclei

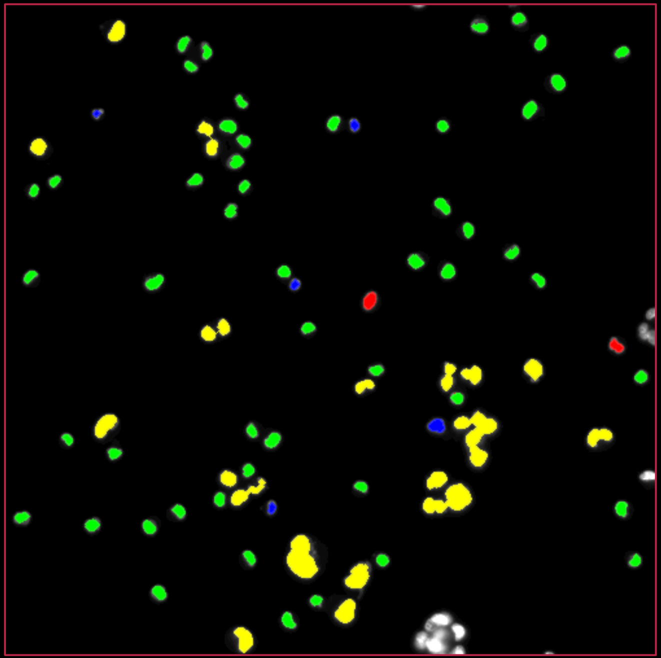
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise		
Actual single-nuclei		



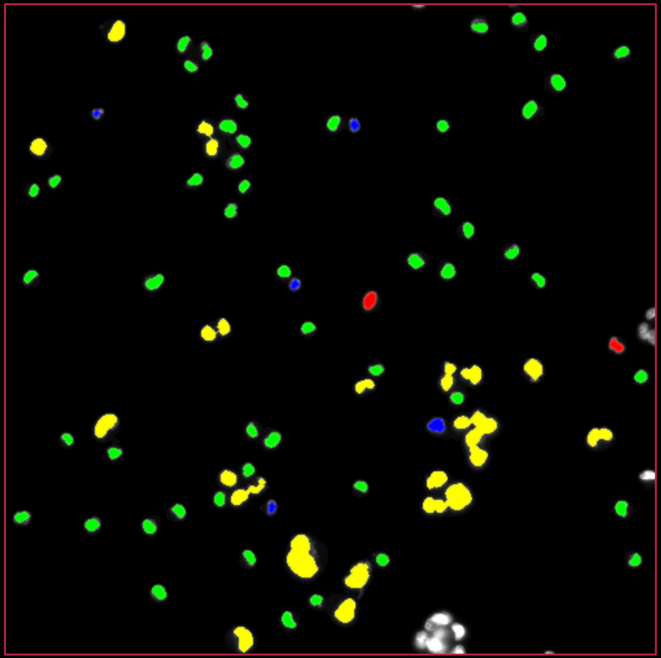
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	
Actual single-nuclei		



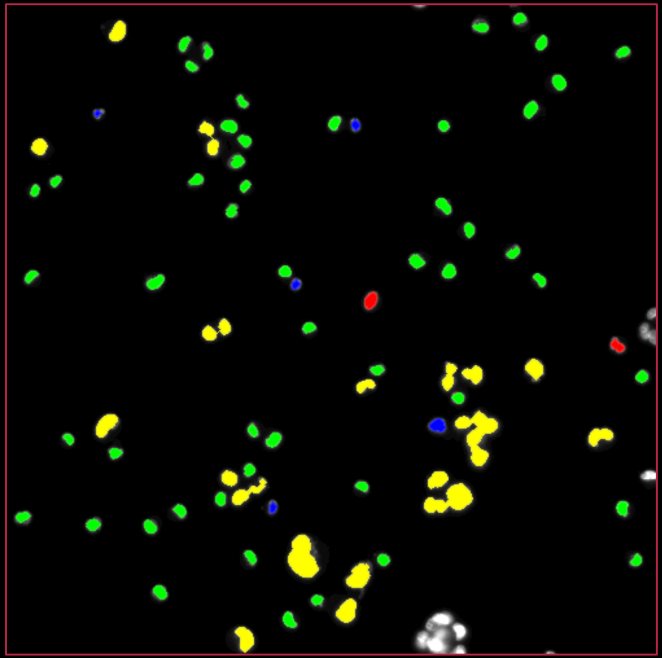
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	
Actual single-nuclei		TP=51



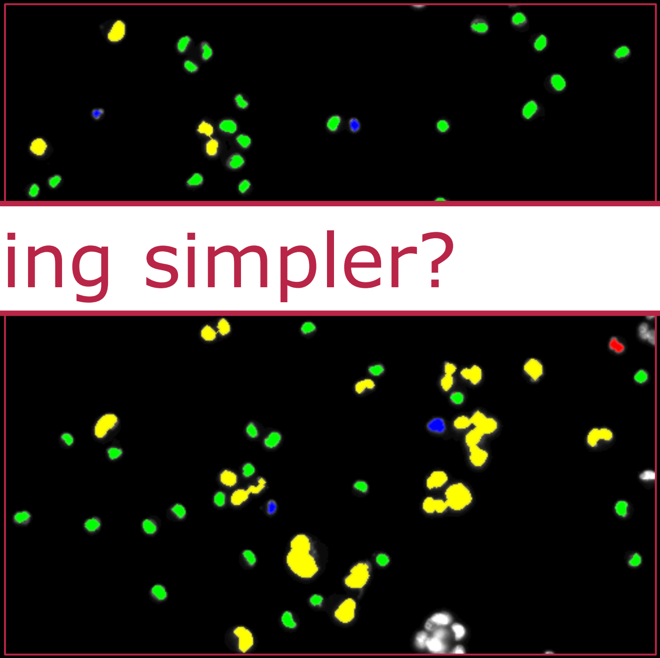
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei		TP=51



Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51



Something simpler?

Accuracy

- Tells how often the classifier is correct

$$\text{Accuracy} = \frac{TP+TN}{N}$$

- N is the total number of annotated objects

$$N = TN + TP + FP + FN$$

Accuracy

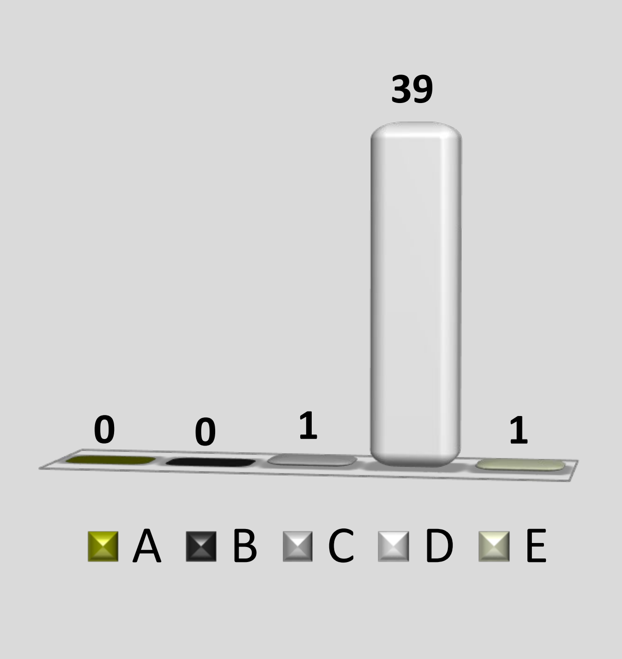
A) 42%

B) 65%

C) 77%

D) 91%

E) 97%



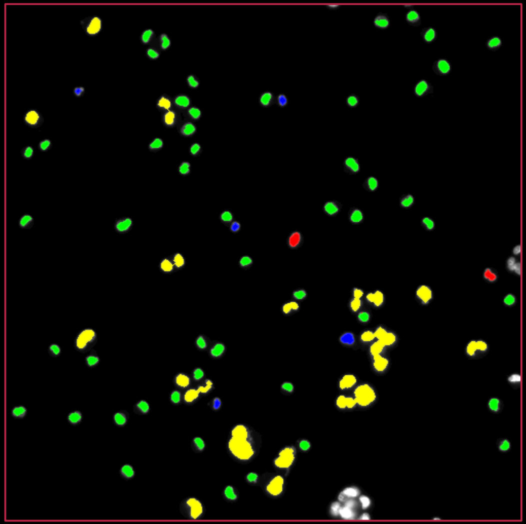
	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

True positive rate (sensivity)

- How often is a positive predicted when it actually is positive

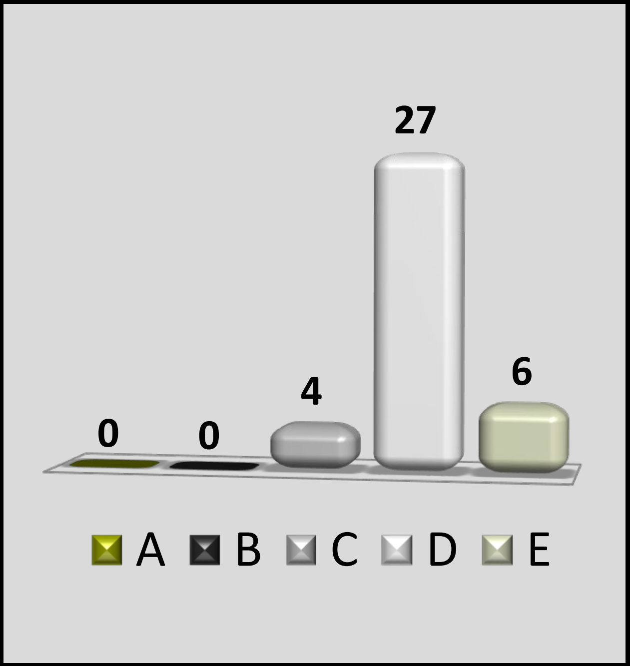
$$\text{Sensivity} = \frac{TP}{FN+TP}$$

All the experts true single-nuclei



Sensitivity

A) 62%
B) 65%
C) 71%
D) 91%
E) 93%



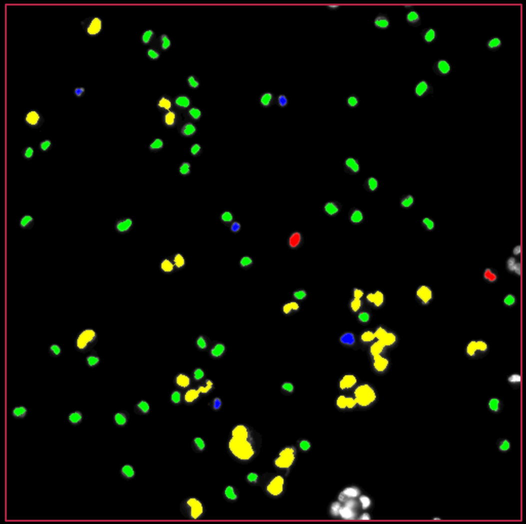
	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

Specificity

- How often is a negative predicted when it actually is negative

$$\text{Specificity} = \frac{TN}{TN+FP}$$

All the experts true noise objects



Specificity

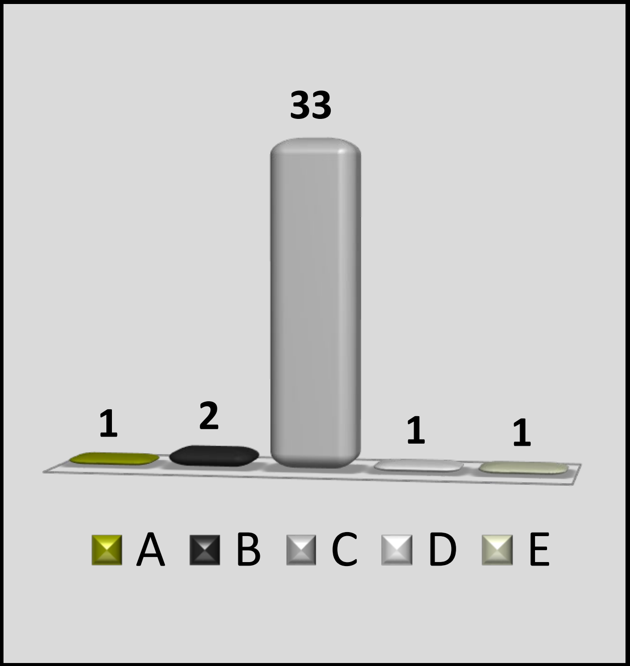
A) 77%

B) 81%

C) 90%

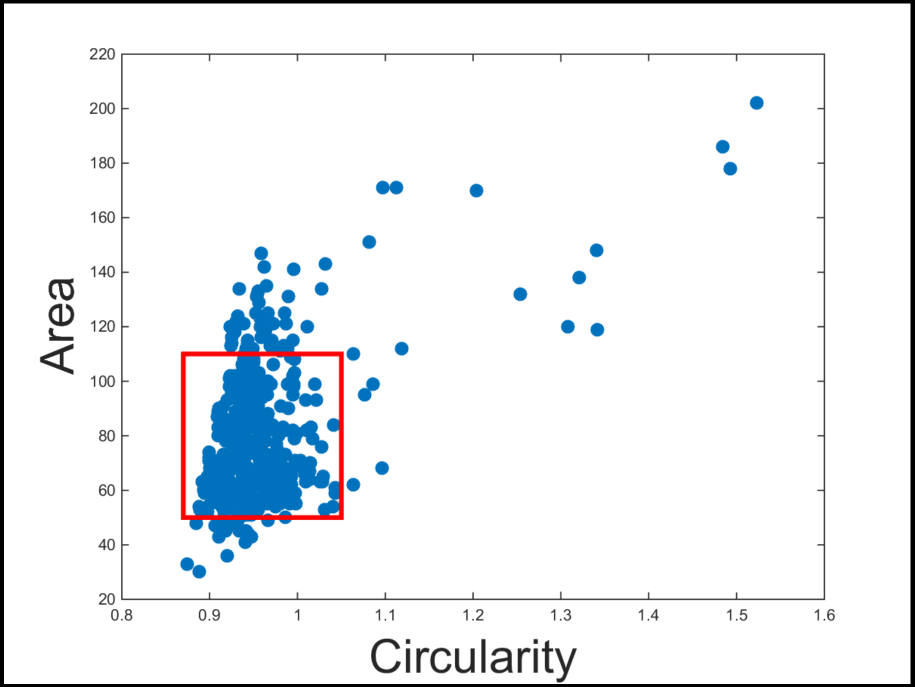
D) 92%

E) 97%



	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

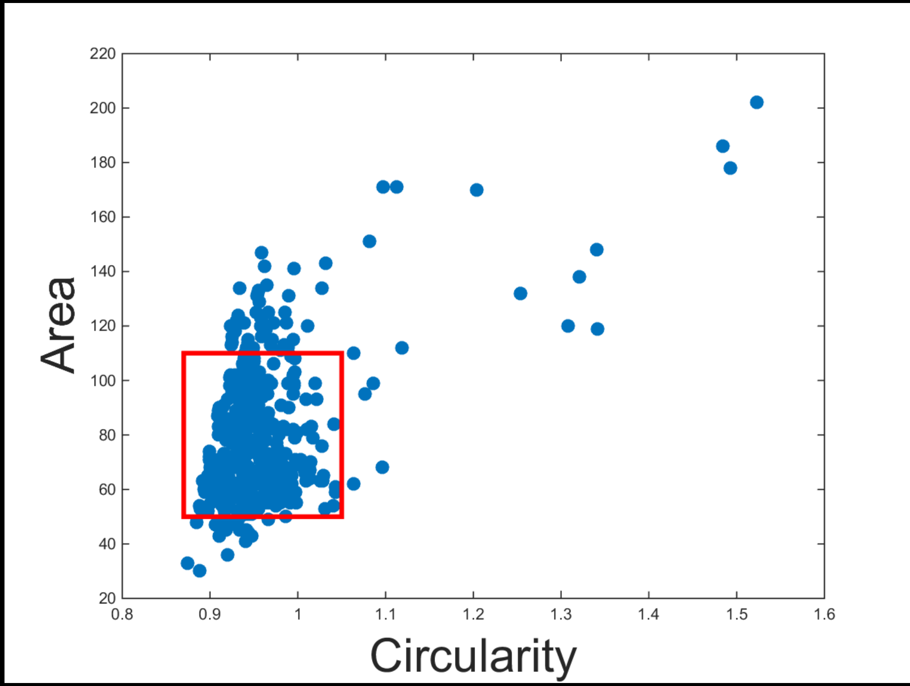
Optimising the classification



- Changing the classification limits
- The rates will be changed:
  - Accuracy
  - Sensitivity
  - Specificity
  - …
- Very dependent on the task what is optimal

Dependencies

- Increasing true positive rate
  - Increased false positive rate
  - Decreased precision

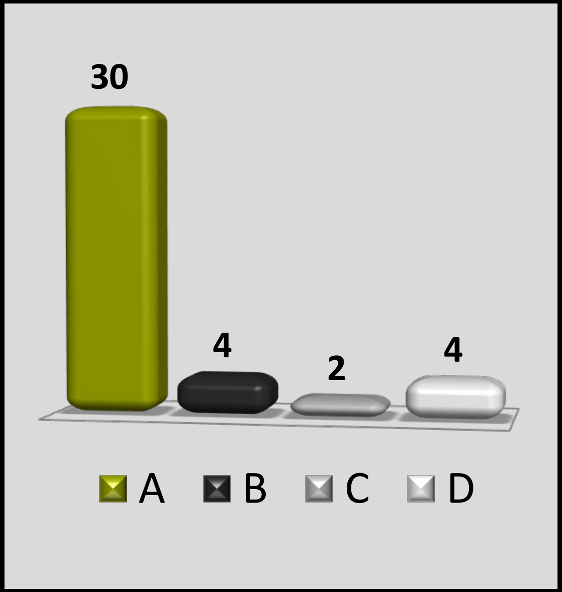


Example – cell analysis

- We want only single-nuclei cells
  - For further analysis
- We do not want to do an analysis of a noise object
- We are not interested in the true number of single nuclei

What measure is the most important?

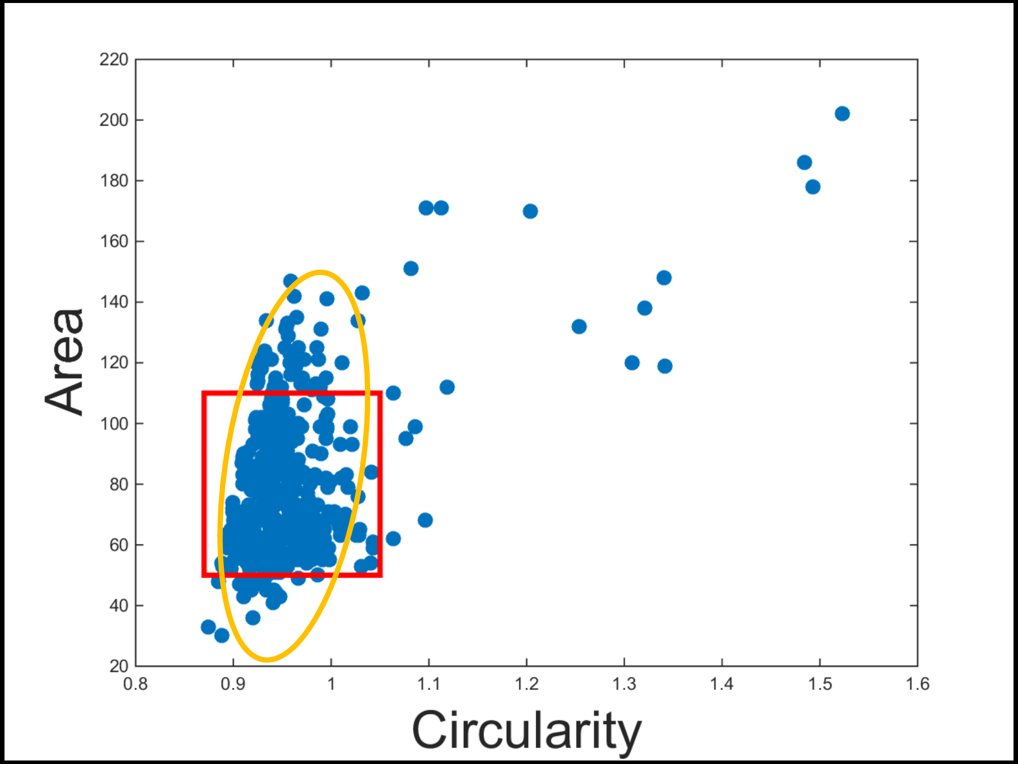
A) Low false positives
B) High true positives
C) High true negatives
D) Low false negatives



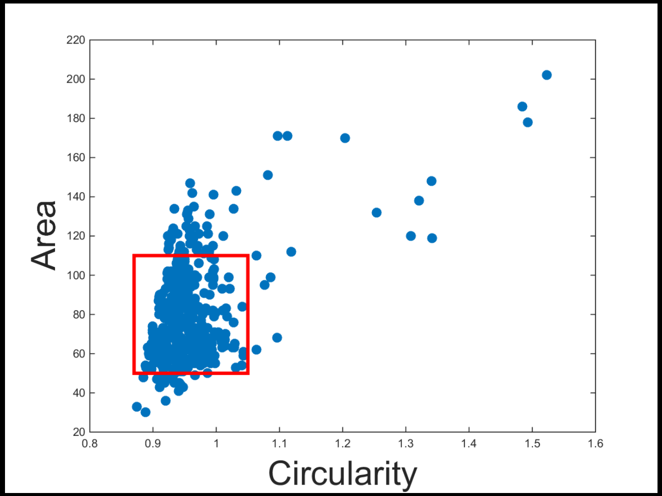
- We want only single-nuclei cells
  - For further analysis
- We do not want to do an analysis of noise objects
- We are not interested in the true number of single nuclei

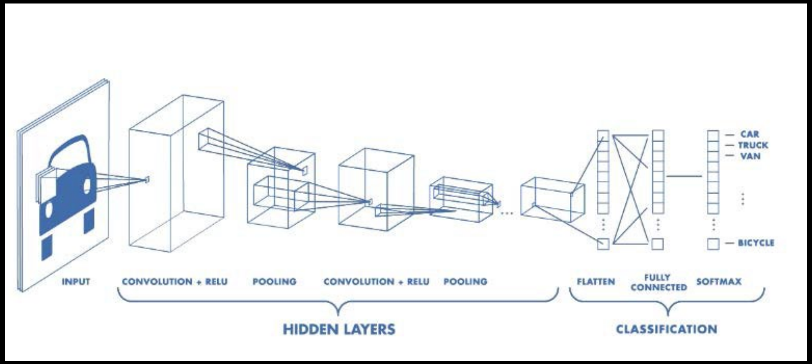
Advanced classification

- Fitting more advanced functions to the samples
- Multivariate Gaussians
- Mahalanobis distances



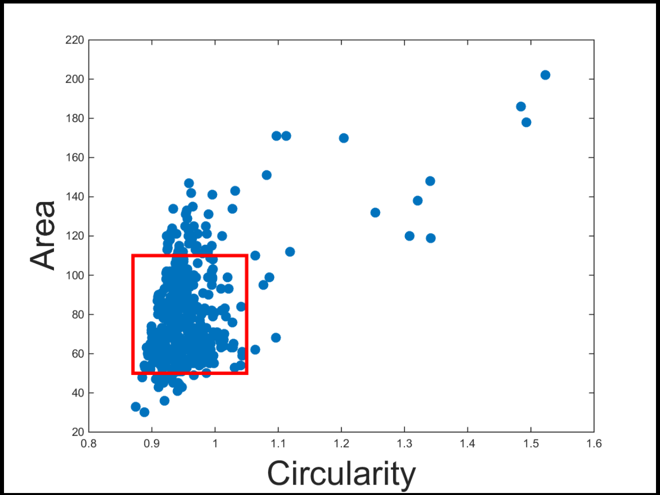
Feature Engineering vs. Deep learning





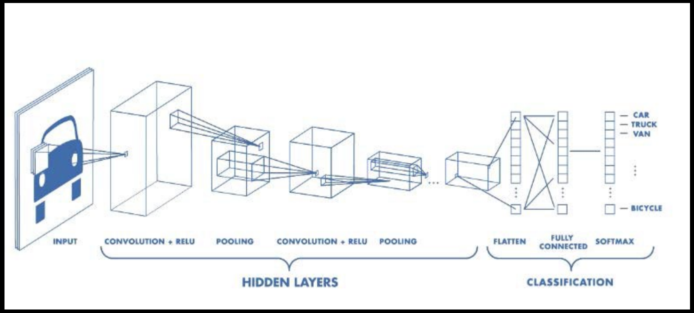
- Until around 5-7 years ago feature engineering was the way to go
- Now deep learning beats everything
- However – feature engineering is still important

Feature engineering



- Given a classification problem
  - Cars vs. Pedestrians
- Use background knowledge to select relevant features
  - Area
  - Shape
  - Appearance
  - …
- Use multivariate statistics to classify
- Depending on the selected features

Deep learning

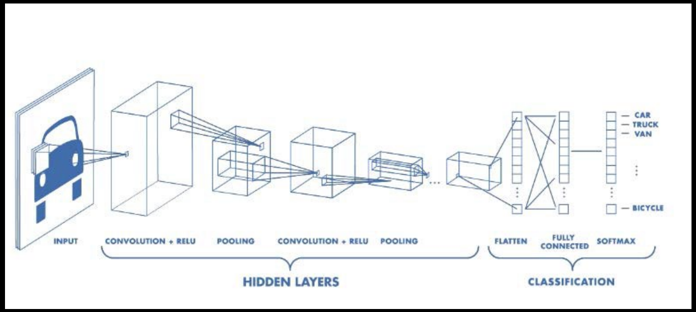


- You start with a dummy classifier
- Feed it with lots and lots of data with given labels
- The network learns the optimal features
- Layer/network engineering

Feature Engineering vs. Deep learning

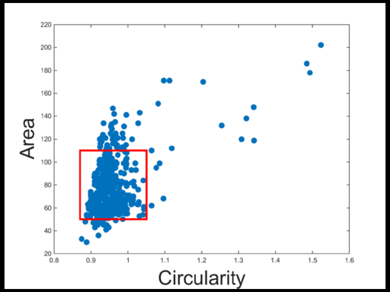
Deep Learning

- When you have lot of annotated data
- Where it is not clear what features work



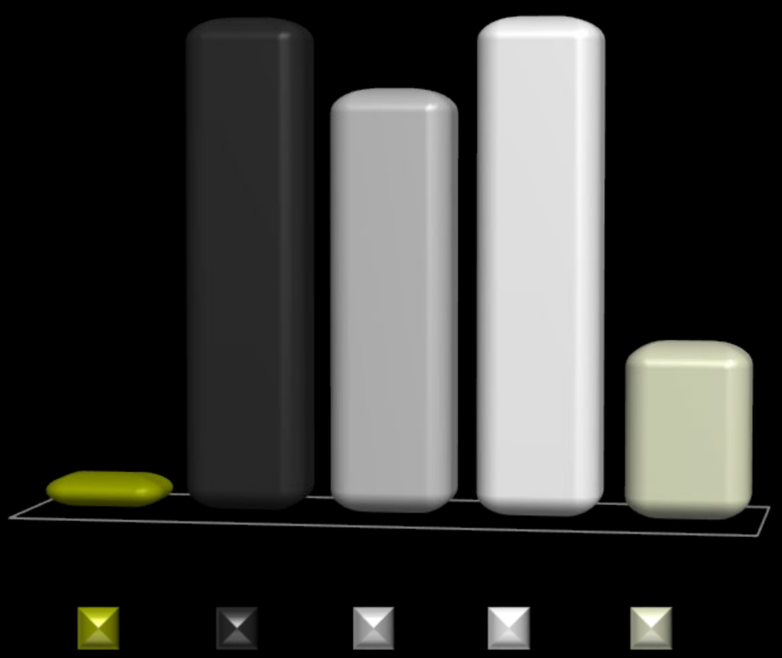
Manual features

- When you have limited data
- When it is rather obvious what features can discriminate



The level of medical examples

A) Stop stop! Too much medical stuff
B) I would like some more non-medical examples
C) Its ok
D) I like the medical examples
E) Great with medical examples



Next week

- Pixel Classification