KU LEUVEN

TOBCat

Industriële Toepassingen van Objectcategorisatie

EAVISE

Embedded & Applied Vision Engineering

Hands-on workshop on object categorization using Linux, C++ & OpenCV

14 februari 2014

IWT-Tetra-project TOBCAT (nr. 120135)

IWT

Your lunch is sponsered by

data vision

PROGRAM

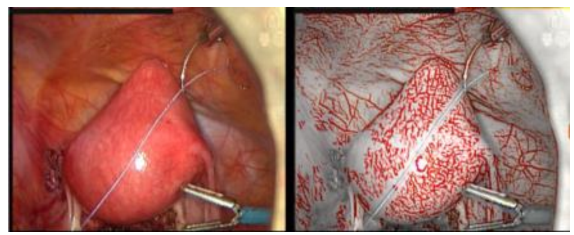
09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop

KU LEUVEN

EAVISE

Embedded Artificially intelligent VISion Engineering

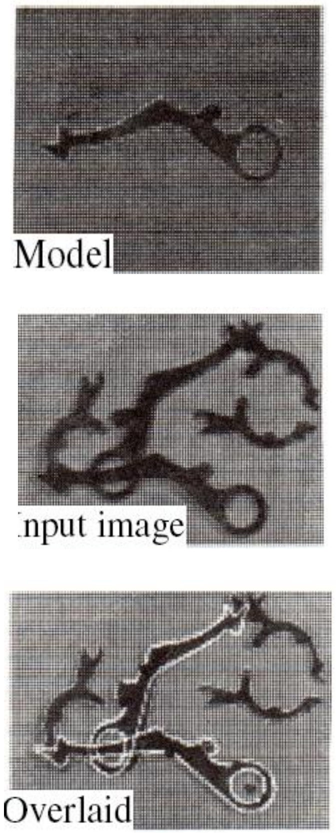
- **Translating** state-of-the-art image processing algorithms to solutions for specific problems in industrial applications.
- **Implementing advanced image processing** techniques on **embedded systems**.
- Optimizing vision algorithms to reach **real time performance**.
- Applying new **Artificial Intelligence** techniques in computer vision applications.



PROGRAM

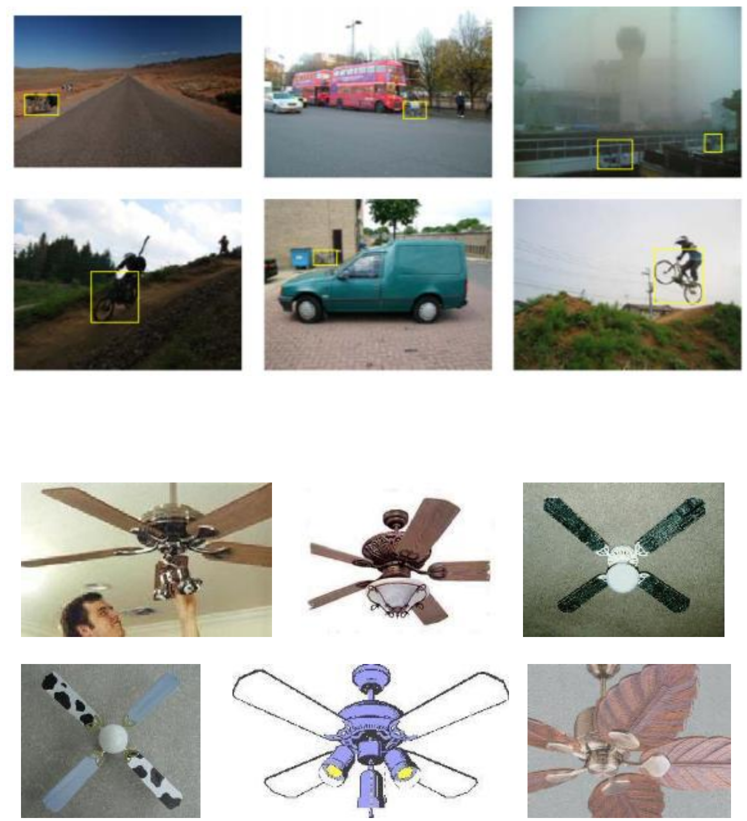
09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop

RECENT EVOLUTION OF VISUAL OBJECT DETECTION



1980 's

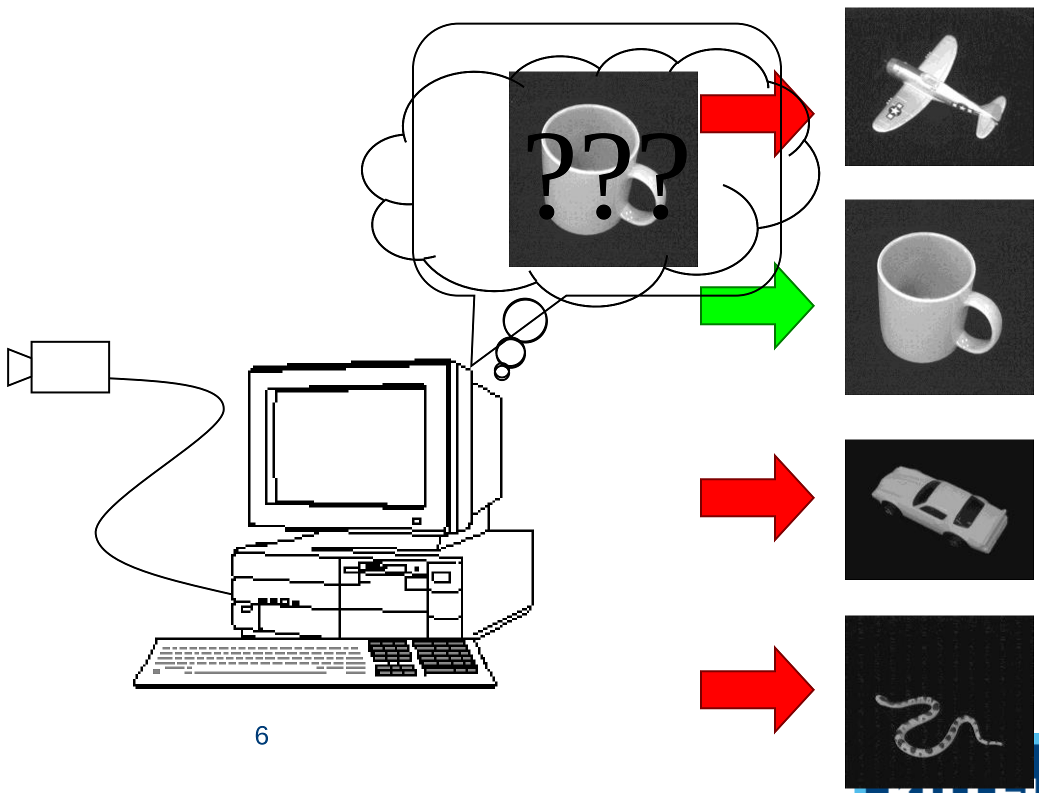
1990 's to begin 2000 's



For the moment

WHAT IS OBJECT CATEGORIZATION?

Object recognition Object identification	}	≠	{ Object detection Object categorisation Object classification



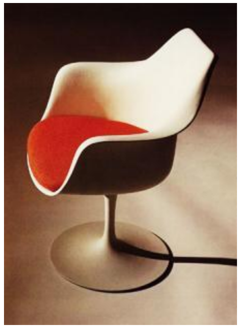
WHAT IS OBJECT CATEGORIZATION?

- FOCUS → objects within a same class show in between variations in color, shape, size, … e.g. cars



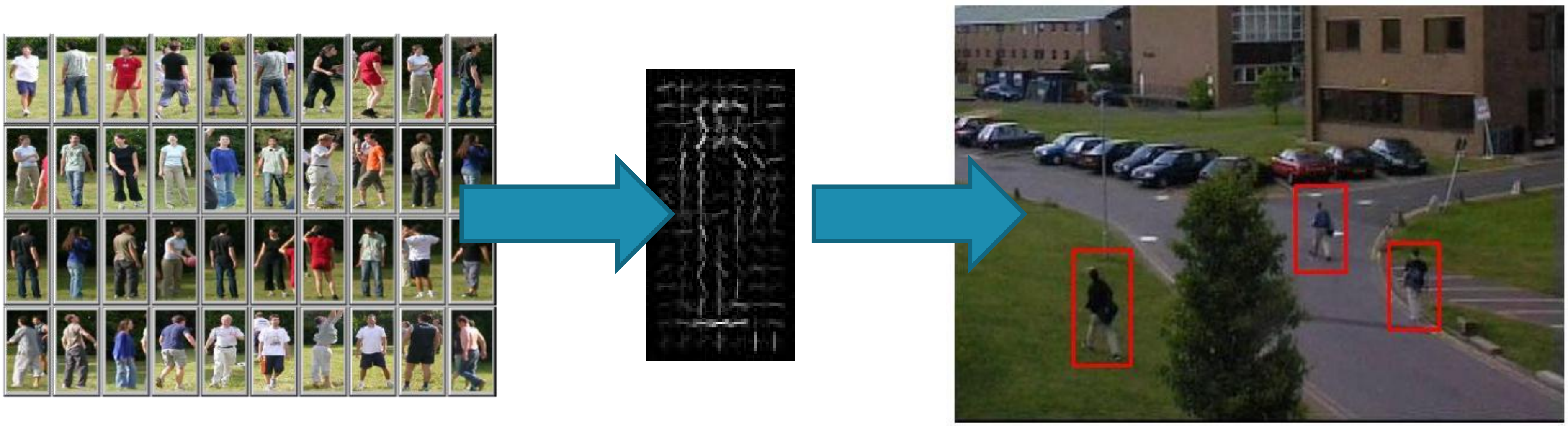

- It becomes harder when more and more variation occurs





OVERAL APPROACH WITH OBJECT CATEGORIZATION TECHNIQUES

- **Training step**: learning a general description from and object class and store it into a **model**
- **Detection step**: searching in new images for objects by comparing the existing model with the input image



OVERAL APPROACH WITH OBJECT CATEGORIZATION TECHNIQUES

A LOT OF VARIATION CHALLENGES


Lighting


Object pose

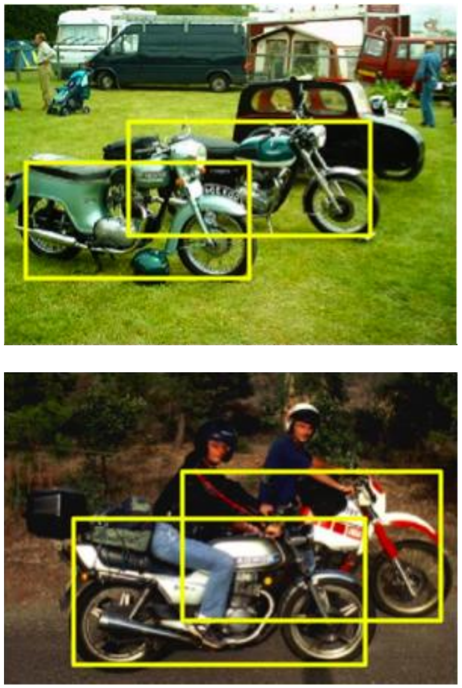

Clutter

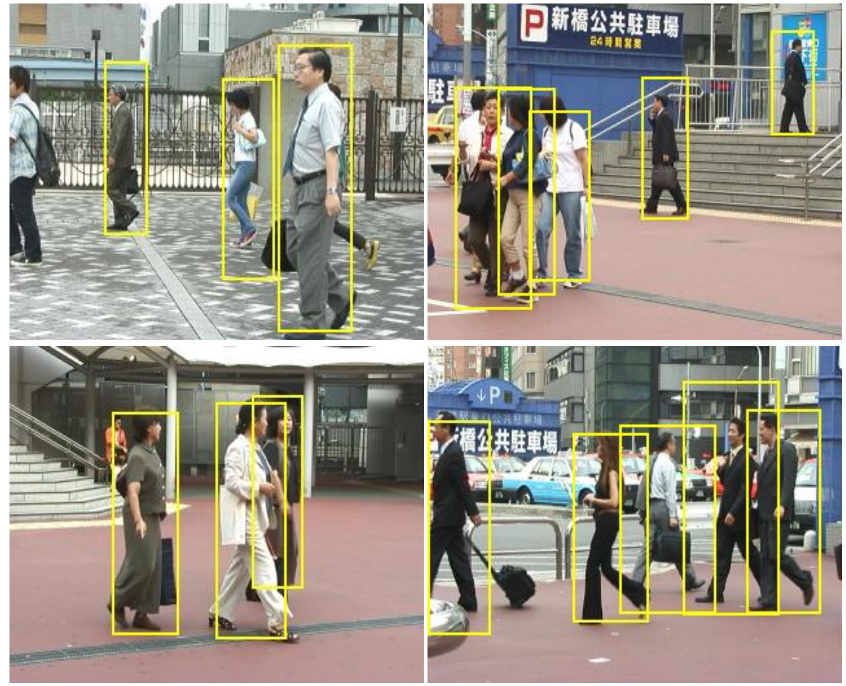

Occlusion


Intra-class appearance


Viewpoint

GETTING A ROBUST DETECTOR



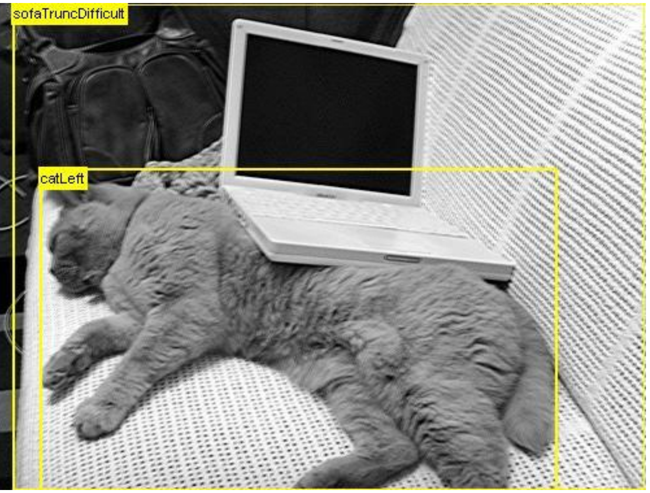


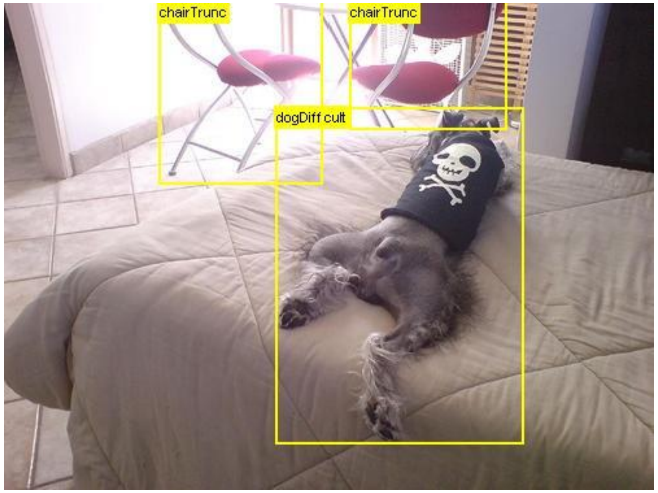
- State-of-the-art techniques are able to do alot:
  - Learning variation (appearance, scale, shape, …) contained in object classes.
  - Compensating for clutter occlusion and overlapping objects.

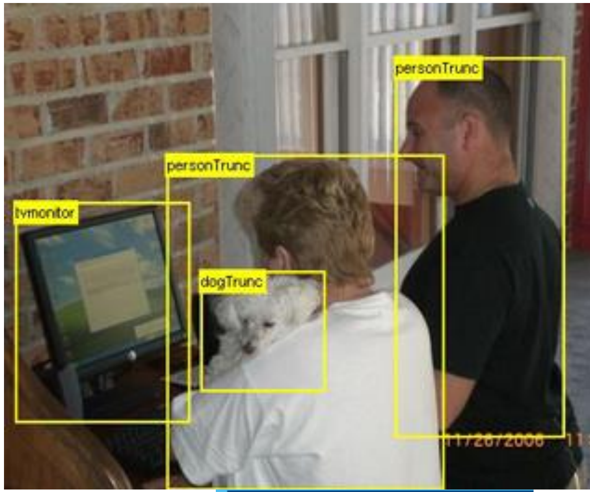
AIM OF TOBCAT PROJECT

Industriële Toepassingen van Objectcategorisatie

- **Introducing** these modern state-of-the-art techniques of object classification to the target group of industrial companies.
- Making the available technology **transparent and easy to use** for industrial companies, making them able to use the technology themselves.
- Introducing object categorization in companies of the user group, so that they can solve their problems using these techniques ➔ **goal of these workshops**

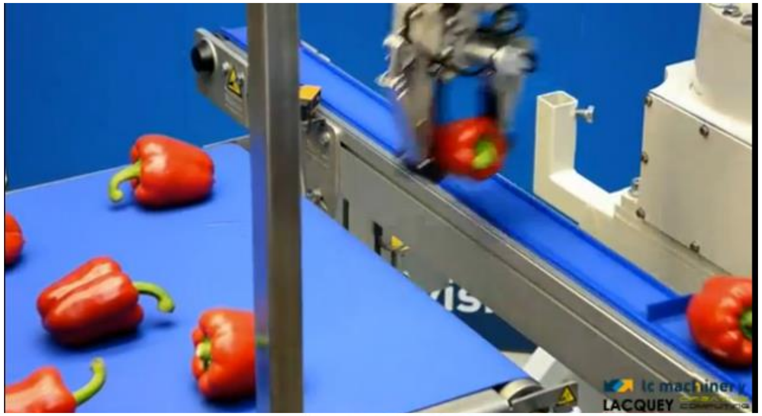


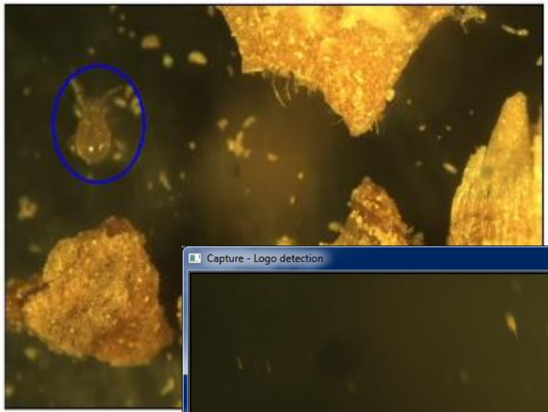


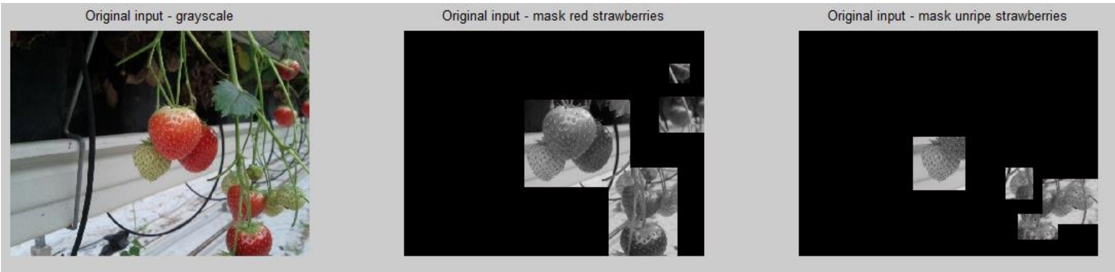


APPLICATIONS IN TOBCAT (1)

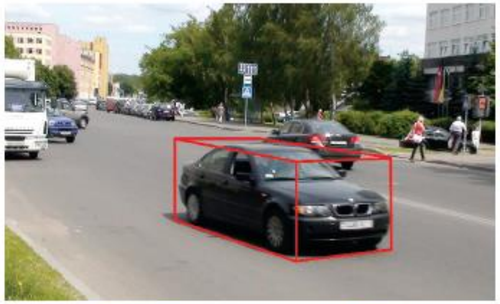
APPLICATIONS IN TOBCAT (2)

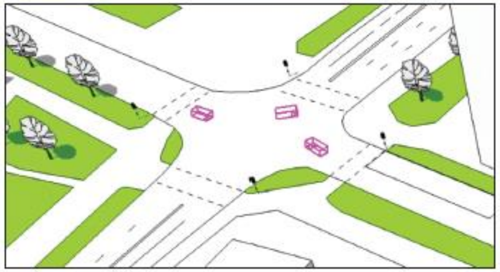






APPLICATIONS IN TOBCAT (3)





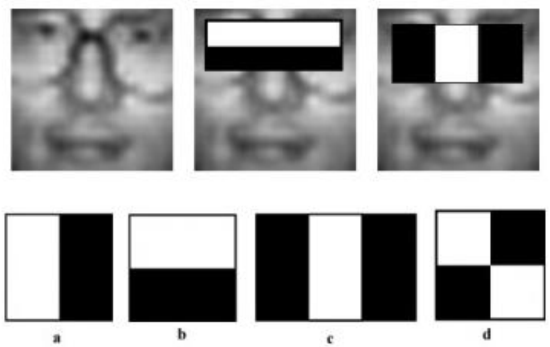
PROGRAM

09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop

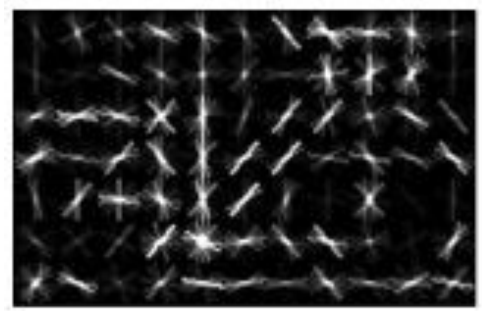
STATE-OF-THE-ART ALGORITMES

1. *Viola&Jones : Haar/AdaBoost [CVPR2001] (workshop)*
2. Dalal&Triggs : HOG/SVM [CVPR2005]
3. Felzenswalb : deformable part models [CVPR2010]
4. Dollár : integral channel features [BMVC2009]

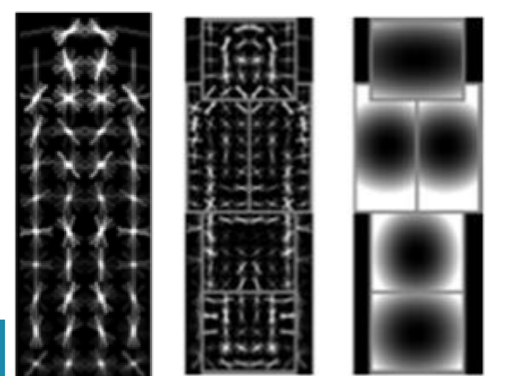
1.



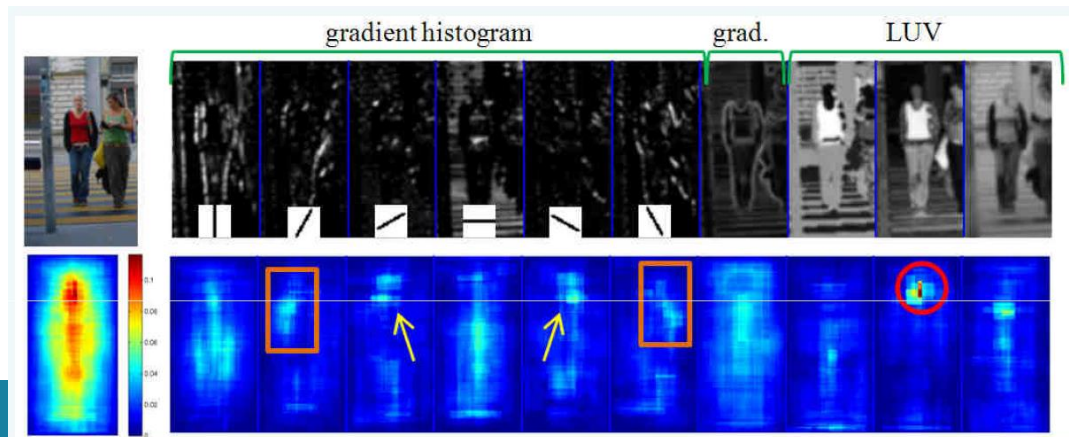
2.



3.



4.



TODAY'S USED TECHNIQUE VIOLA & JONES

Short wrap-up of all steps needed in the algorithm

It all starts from a sliding window approach

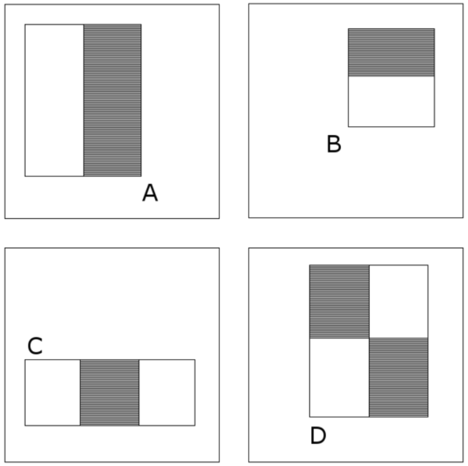
1. Selecting features from window

2. Building a set of weak classifiers

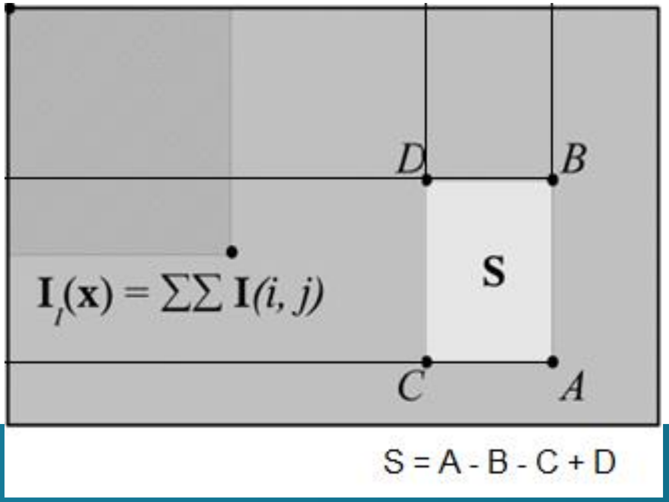
3. Combining weak classifiers to a single strong classifier

KU LEUVEN

TODAY'S USED TECHNIQUE VIOLA & JONES

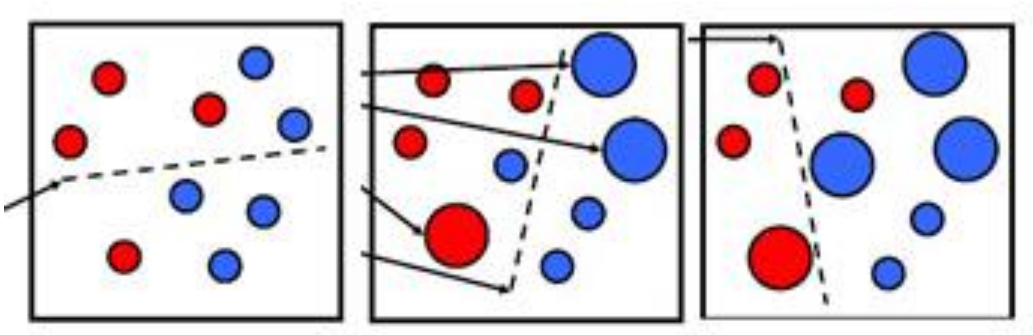


1. Selecting features from window
   - Using HAAR-like wavelets
   - Small filters on image by comparing pixel values in square regions
   - Sum pixel intensity values grey area - sum pixel intensity values white area
   - 24x24 pixels → +-50,000 features
   - Use of integral image
   - Fast calculation of sums



TODAY'S USED TECHNIQUE VIOLA & JONES

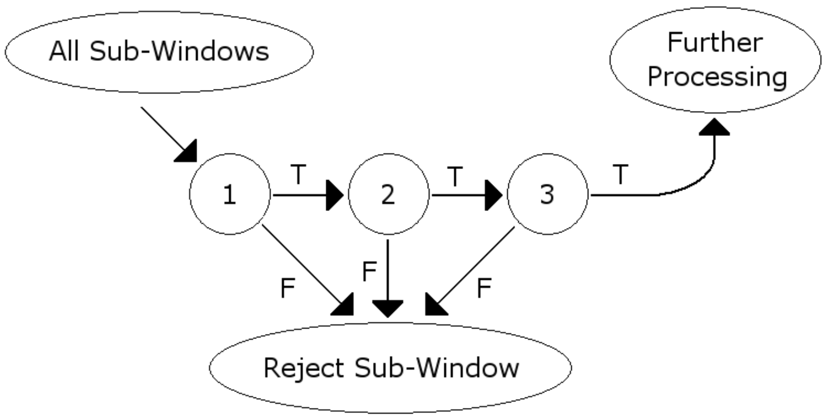
2. Building a set of weak classifiers
   - AdaBoost algorithm
   - Which feature or combination of features can be used to separate objects and non-objects in a rough way
   - Do this until a certain preferred level of separation is reached, e.g. 50% good separation.



KU LEUVEN

TODAY'S USED TECHNIQUE
VIOLA & JONES

3. Combining weak classifiers to a single strong classifier
   - Cascade / waterfall structure
   - Weak classifiers → faster calculation / less features
   - To reduce the error *(individually very high)*
   - 'Early rejection' principle



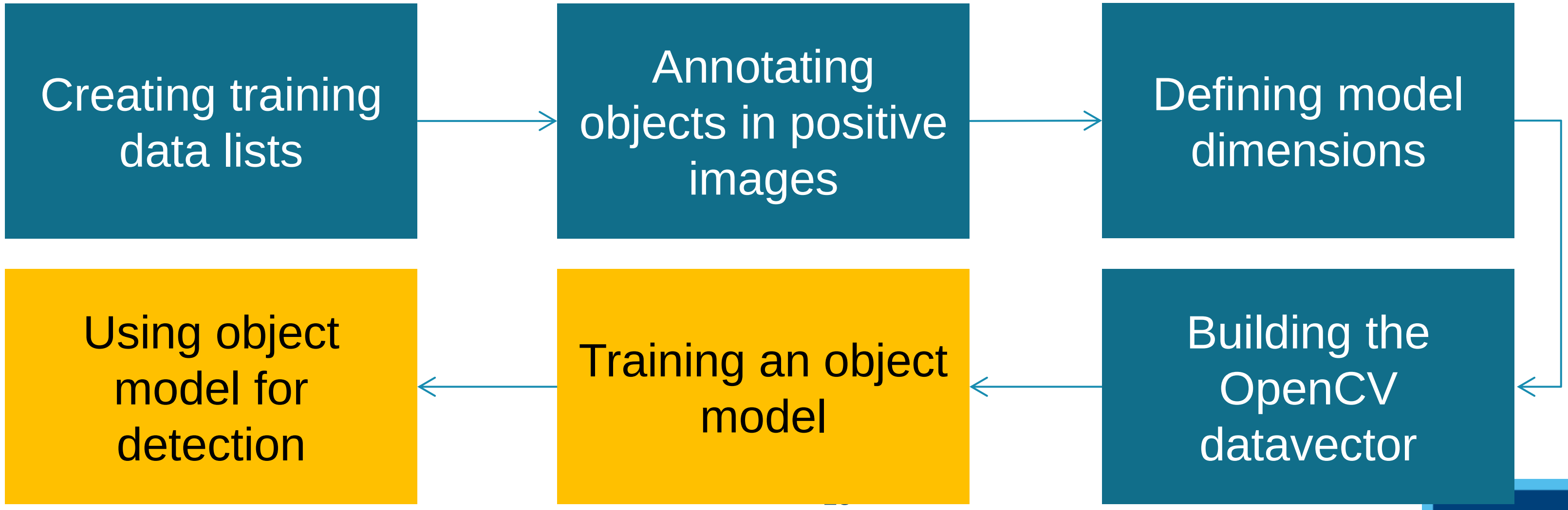
KU LEUVEN

PROGRAM

09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop

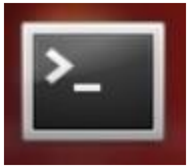
IDEA OF FIRST HANDS-ON SESSION

- From a dataset, prepping all data to be able to built a complete object model.
- *Goal: make a company able to detect an object class on different backgrounds.*
- Required steps:



SOME GUIDELINES FOR HANDS-ON PARTS OF THE WORKSHOP

- Login on computers using tobcat account, pwd = tobcat
- Open a terminal window



  - Standard ~/ directory
  - We will work from **/home/tobcat/workshop/**
- Some of the most used commands
  - cd <path> → changing folder
  - ls → summing the contents of a folder
  - ./<executable_name> [green color in ls] → code snippets
  - If executable is not green → chmod +x <executable>

KU LEUVEN

SOME GUIDELINES FOR HANDS-ON PARTS OF THE WORKSHOP

- As a C++ development environment we use Code::Blocks.
  - Preinstalled on the system
  - Folder software contains all configured projects
  - Folder code_blocks contains code for second hands-on
- Re-occuring problem = Code::Blocks ‘forgets’ OpenCV
  - Project – Build Options – Linker settings – Additional Linker Commands
  - Add `pkg-config opencv --libs` *[with correct quotes!]*
- If there are any software problems, do not hesitate to call for an assistant or to interrupt the hands-on!

OBJECT ANNOTATION TOOL & PREPROCESING STEPS

Lets changed the directory towards

../workshop/data/mini_model/

There is an existing structure

- Positive folder contains images **with** objects
- Negative folder contains images **without** objects

- This structure needs to be manually composed
- Names of folders are not important, however choosing a meaningful name can help to understand everything.

OBJECT ANNOTATION TOOL & PREPROCESING STEPS

Which steps do we have to take in order to be able to train an object model of a specific object class?

1. All code snippets work using txt files with references to the actual data

   SNIPPET – ./folder_listing

   NEEDED – positives.txt / negatives.txt / testset.txt

2. Object annotation – segmenting positive objects from their background information

   SNIPPET – ./annotate_images

   NEEDED – annotation of each object – universal format

KU LEUVEN

OBJECT ANNOTATION TOOL & PREPROCESING STEPS

```
NAME     #DETECTIONS     X1 Y1 W1 H1 ... Xn Yn Wn Hn
D:\cookies\positives\_1.png 6 160 1 138 132 321 5 136 141 153 139 151
D:\cookies\positives\_1a.png 5 90 50 150 146 25 199 168 155 1 354 192
D:\cookies\positives\_2.png 6 141 14 148 138 309 2 141 146 165 164 150
D:\cookies\positives\_2a.png 3 87 47 152 151 33 209 158 138 4 358 135
D:\cookies\positives\_3.png 6 131 43 156 129 299 4 142 137 180 180 149
D:\cookies\positives\_3a.png 3 81 34 143 154 25 206 174 146 6 347 137
D:\cookies\positives\_4.png 6 132 57 153 129 199 195 143 137 261 349 1
D:\cookies\positives\_4a.png 3 77 36 150 157 31 195 160 154 8 349 138
D:\cookies\positives\_5.png 6 117 69 143 152 253 5 154 149 345 145 152
D:\cookies\positives\_5a.png 3 77 39 147 156 34 201 153 150 5 355 142
D:\cookies\positives\_6.png 6 87 89 149 154 180 219 153 143 228 14 148
D:\cookies\positives\_7.png 6 197 19 148 146 75 116 146 153 173 239 14
```


KU LEUVEN

OBJECT ANNOTATION TOOL & PREPROCESING STEPS

Which steps do we have to take in order to be able to train an object model of a specific object class?

3. The annotated data has to be translated to an OpenCV specific data storage format
   - Universal format for model training
   - Reshapes training data to average dimensions

     SNIPPET – ./average_dimensions & ./create_samples

     NEEDED – datavector.vec

OBJECT ANNOTATION TOOL & PREPROCESING STEPS

Usefull tools - snippets for companies

1. **./video2images** – lots of data is captured as video material. This snippet will make sure that videos can be cut into frames without compression loss.
2. **./generate_negatives** – a lot of companies collect images from objects but not the actual backgrounds without objects
   - Reads an annotation file
   - Cuts the annotations from the positive images
   - Uses the cut result as negative background images
   - Has influence on performance! (unnatural image constructions)

LUNCH

The lunch is offered to us by data vision

- A system of self service (dessert / soup / lunch)
- We eat in dining room ‘de fruytenborg’
- Coffee afterwards is included

PITCH – Data Vision

A small company pitch by data vision

PROGRAM

09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop

TRAINING PROCES + TESTING DETECTOR WITH OBJECT MODEL

Until now, we prepared data for training an actual object model.

- ./train_cascade →SNIPPET

- Test with ‘simple’ model
  - Get the hang of it!
  - Variation in candies itself → segmentation here would already be a difficult task
  - On a test set background from our lab

- We will take a closer look at the output of the training

TRAINING PROCES + TESTING DETECTOR WITH OBJECT MODEL

For the second hands-on session, we will focus on an already trained object model:

- Go to …/data/candy_model/
- 160 positive images 1000 negative images
- 18 stage classifier = # combined weak detectors

First we will test the interface of OpenCV for object detection, play with important parameters, then we will do it ourselves.

1. Preprocessing image – grayscale / histogram equalization
2. Detection and parameter influence in code
3. Visualization and parameter influence in code

KU LEUVEN

ROTATION INVARIANCE

1 model = 1 orientation

- How can we deal with 1 single model
- Should we place all rotations in a single model?
- Should we rotate the image or the patch?

Live simulation of the rotation invariant candy detector

- Influence of parameters
- Real time performance possible using specific knowledge?
- Taking a look at parameters in source code

PROGRAM

09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop

SOME DOWNSIDES OF TECHNIQUES (ROTATION, CLUTTER, OCCLUSION)

Rotation was already discussed before

Technique is partially resistant to clutter

- Depends strongly on training data
- Only perfect objects → imperfect objects will never be detected

Technique is not resistant to occlusion.

- Detectors fail pretty soon when occlusion happens
- However the DPM approach is a valid alternative and a bit more robust than V&J framework

SOME DOWNSIDES OF TECHNIQUES (ROTATION, CLUTTER, OCCLUSION)

In beginning of session we discussed 4 techniques, so what can you expect from OpenCV and C++ possibilities?

- Viola & Jones in OpenCV
  - Well supported – tutorials / documentation / bug free
  - Large community gives great support

- SVM + HOG
  - Partial components in OpenCV, a detection framework
  - Not combined to an effective training/detection framework
  - Machine learning SVM → bad support/code

SOME DOWNSIDES OF TECHNIQUES (ROTATION, CLUTTER, OCCLUSION)

In beginning of session we discussed 4 techniques, so what can you expect from OpenCV and C++ possibilities?

- DPM model of Felzenszwalb
  - OpenCV only has detection – Latent SVM module
  - Based on Pascal VOC Challenge models & software
  - Not latest implementation, no new models since challenge was stopped
  - Training - original project:

- ICF Dollar
  - OpenCV ‘development’ branch …

KU LEUVEN

SOME DOWNSIDES OF TECHNIQUES (ROTATION, CLUTTER, OCCLUSION)

In beginning of session we discussed 4 techniques, so what can you expect from OpenCV and C++ possibilities?

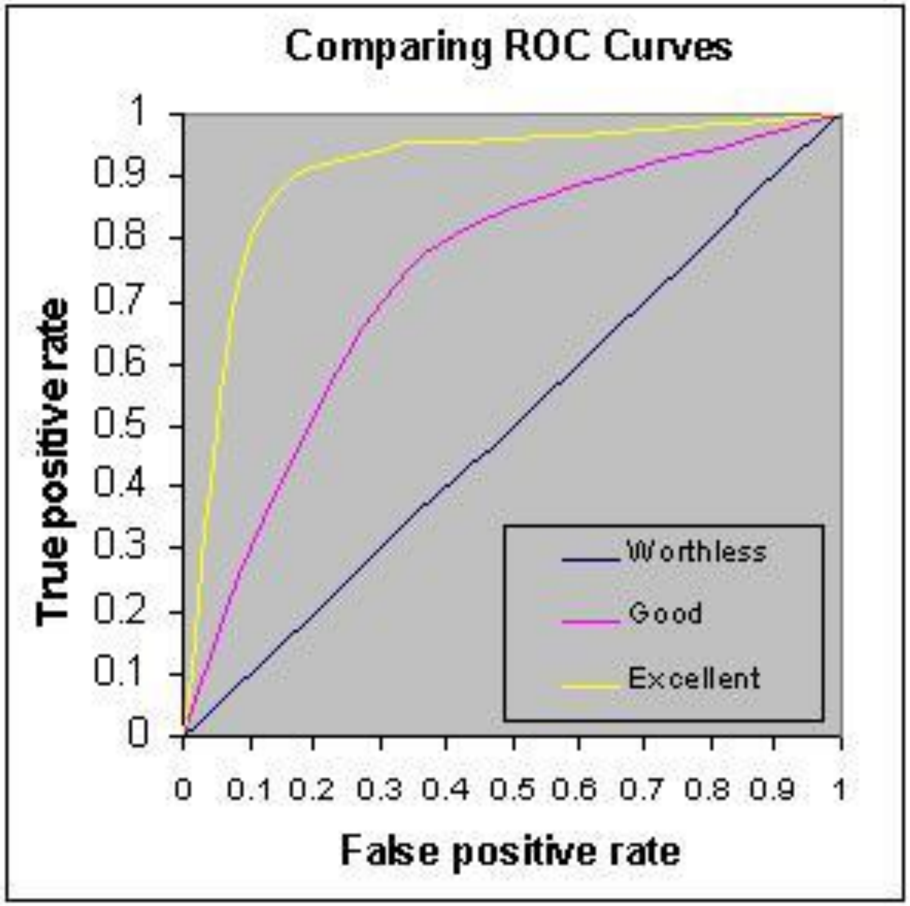
- All software will be made available on TOBCAT website, also code developed in future.
- Also through a github account (source code repository)

  https://github.com/StevenPuttemans/tobcat

PROGRAM

09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop

EVALUATING OBJECT DETECTORS: RECEIVER OPERATING CHARACTERISTIC



INTRODUCTION TO ROC CURVES

- *ROC* = *Receiver Operating Characteristic*
- Started in electronic signal detection theory (1940s - 1950s)
- Has become very popular in biomedical applications, particularly radiology and imaging
- Also used in machine learning applications to assess classifiers
- Can be used to compare tests/procedures

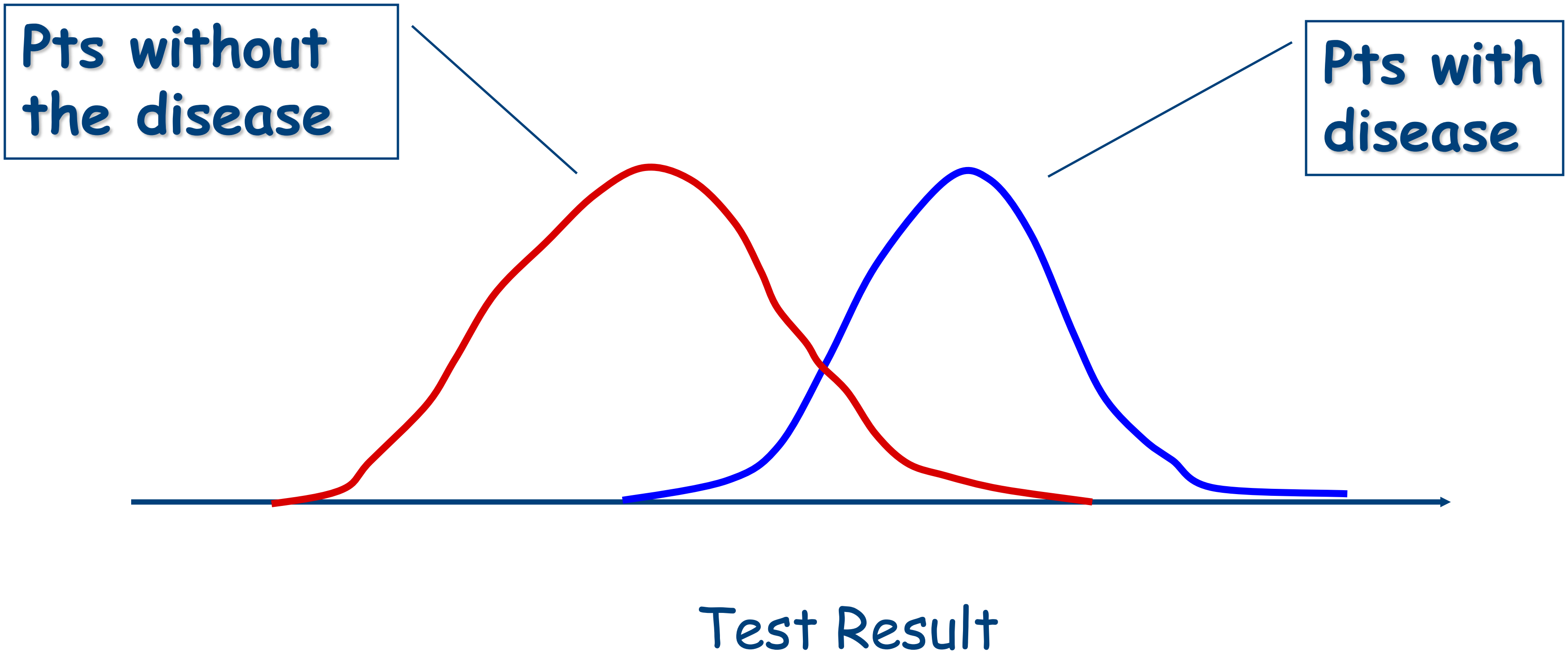
ROC CURVES: EXAMPLE CASE

- Consider diagnostic test for a disease
- Test has 2 possible outcomes:
  - ‘positive’ = suggesting presence of disease
  - ‘negative’
- An individual can test either positive or negative for the disease

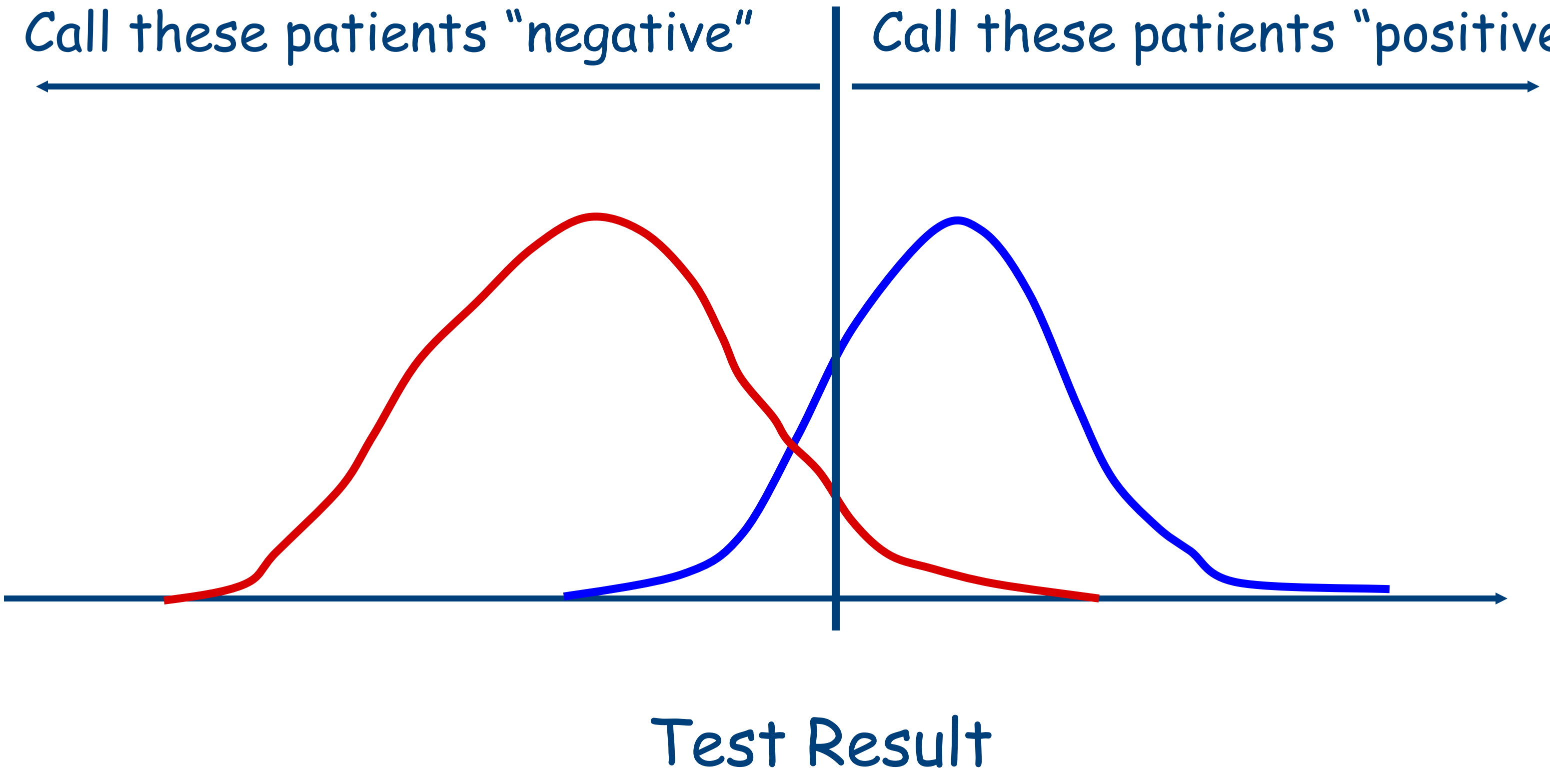
True disease state vs. Test result

Disease \ Test	positive	negative
Disease	True Positive TP	False Negative FN (Type II error)
No disease	False Positive FP (Type I error)	True Negative TN

SPECIFIC EXAMPLE

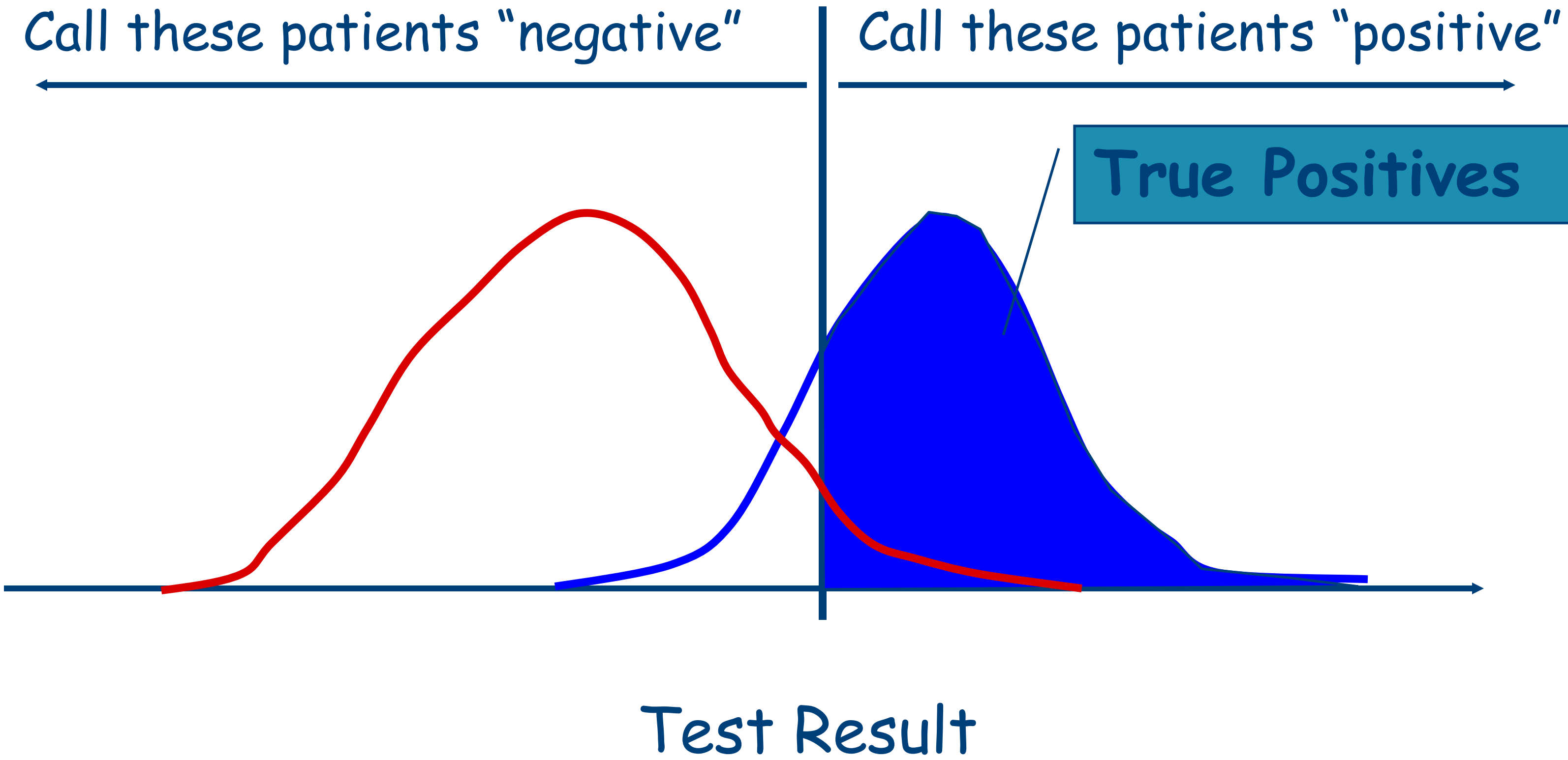


THRESHOLD



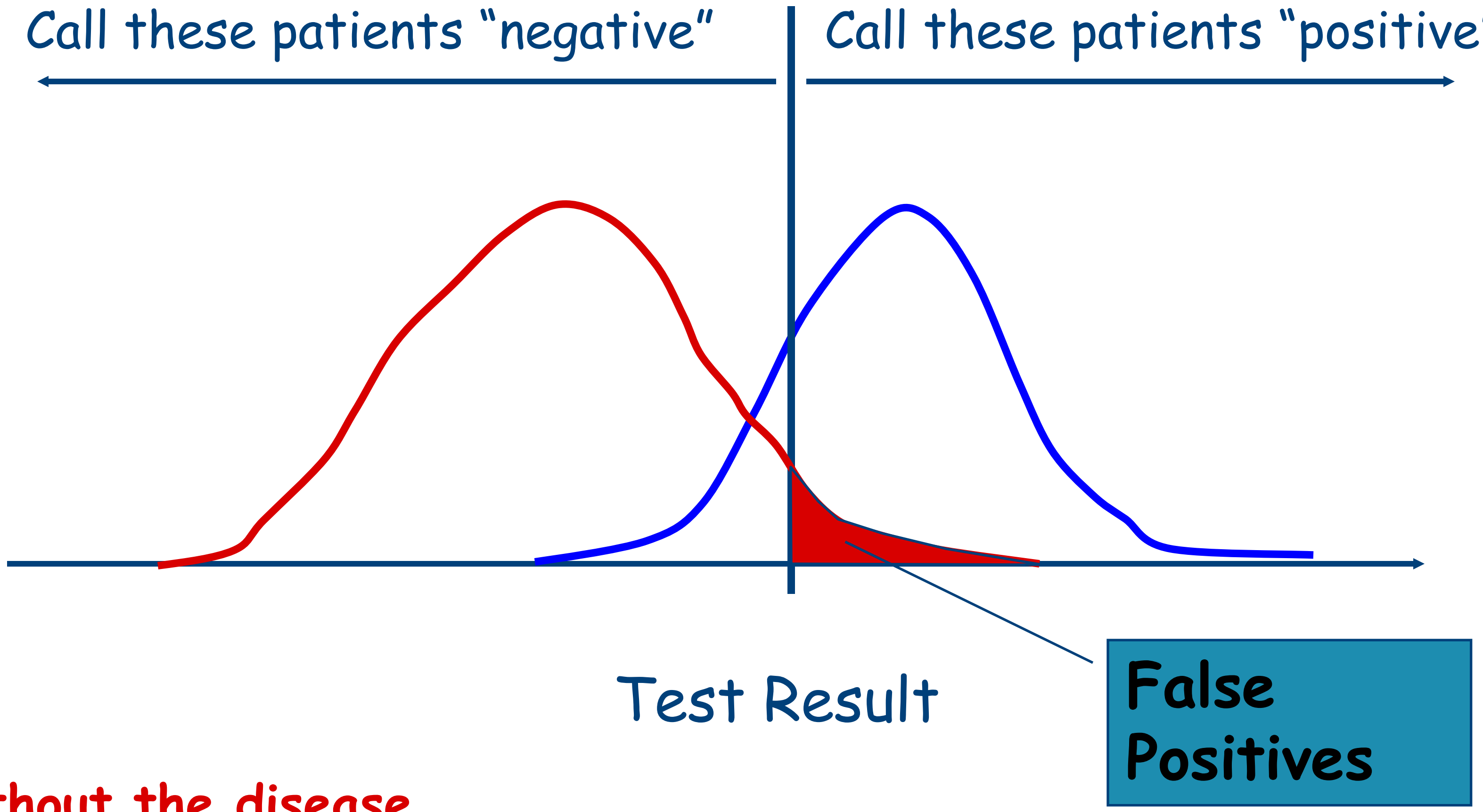
Call these patients "negative"
Call these patients "positive"
Test Result

SOME DEFINITIONS ...



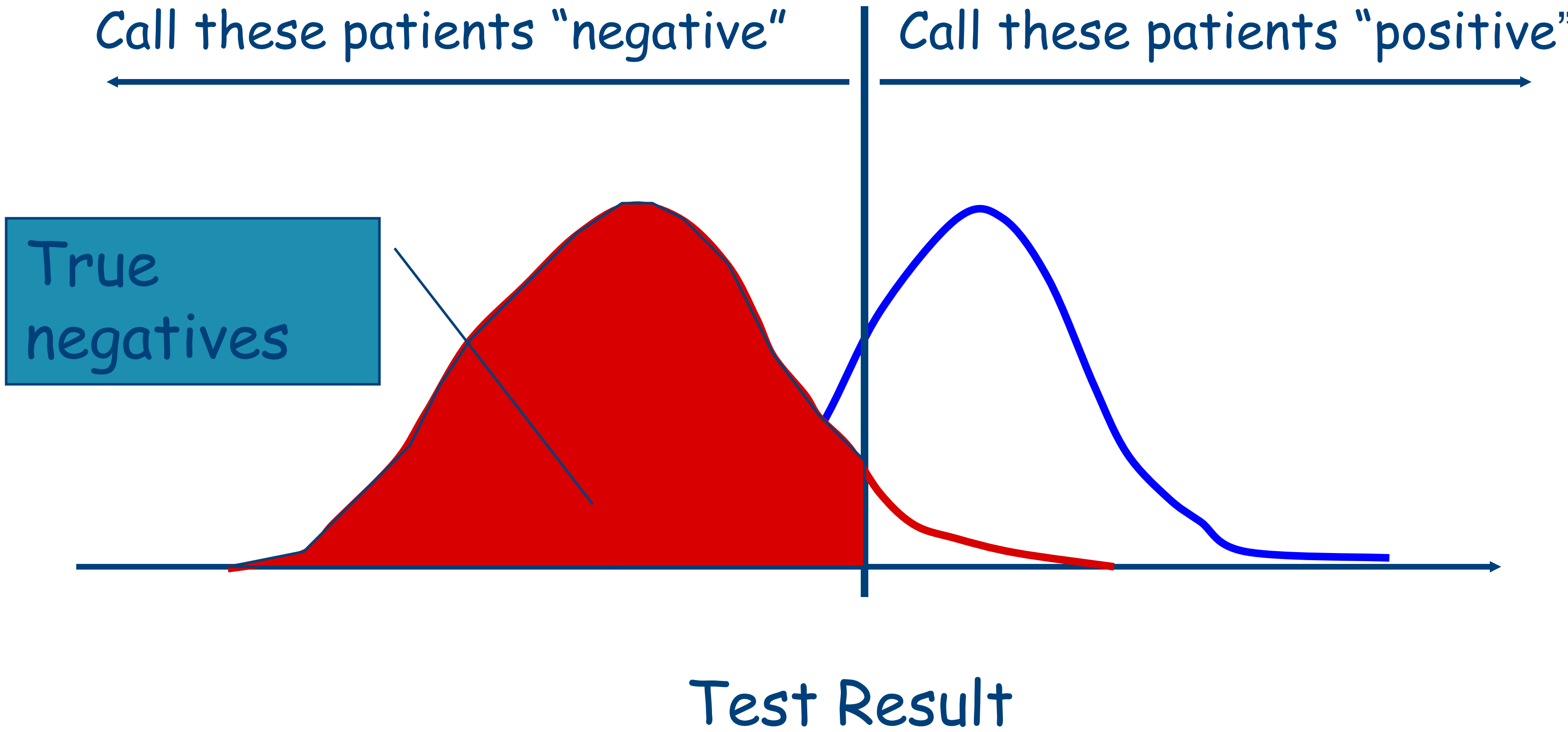
without the disease
with the disease

SOME DEFINITIONS ...



without the disease
with the disease

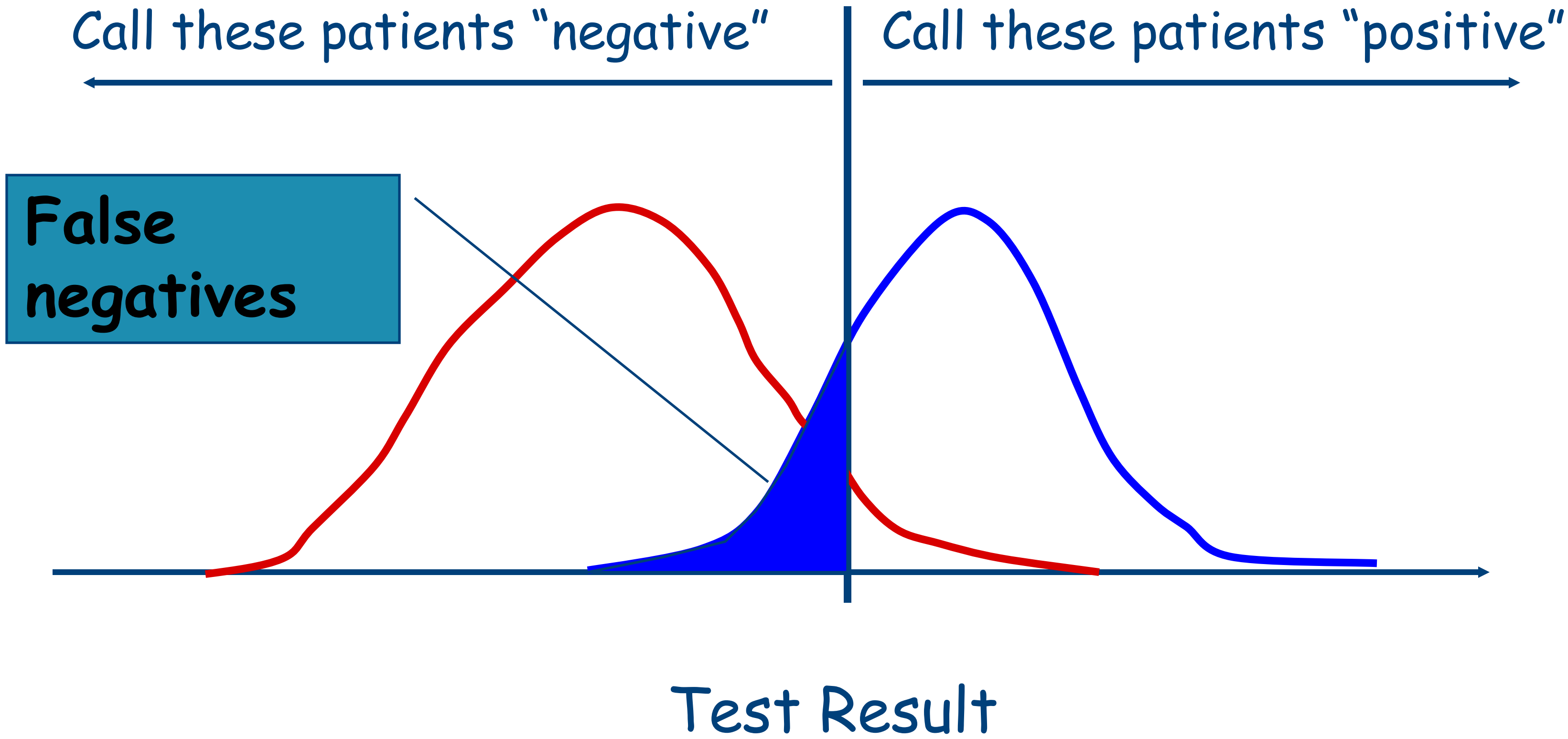
SOME DEFINITIONS ...



without the disease

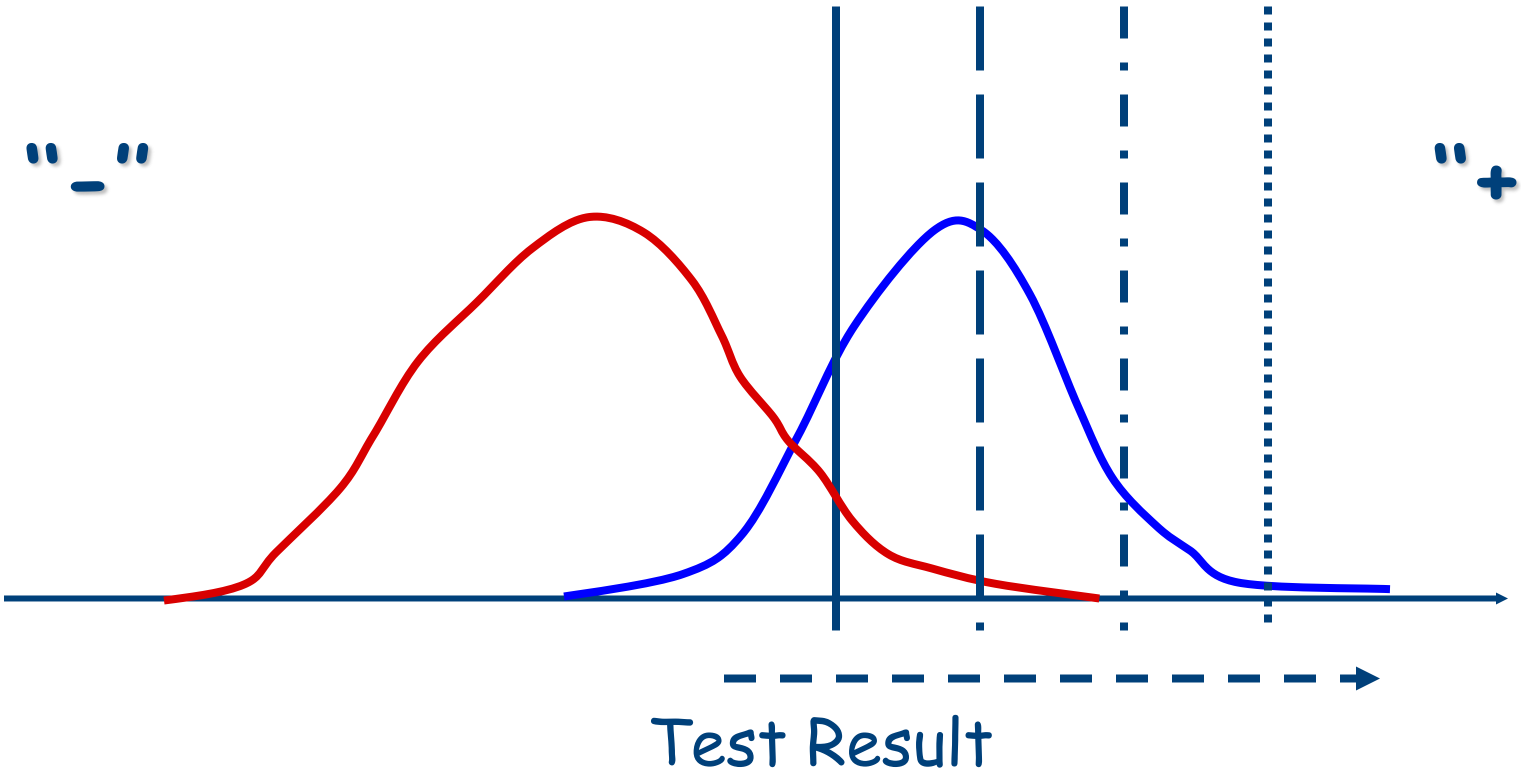
with the disease

SOME DEFINITIONS ...



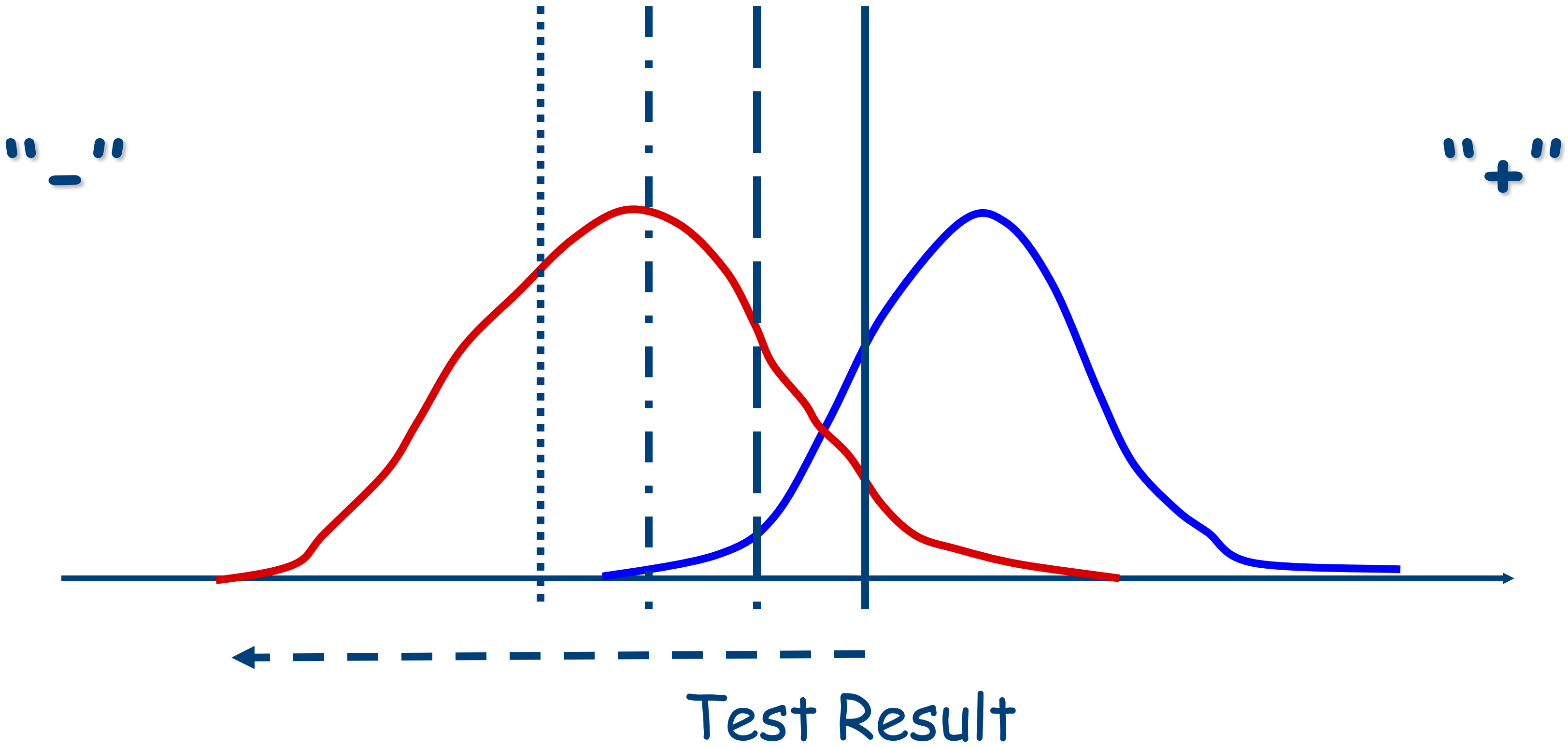
without the disease
with the disease

MOVING THE THRESHOLD: RIGHT



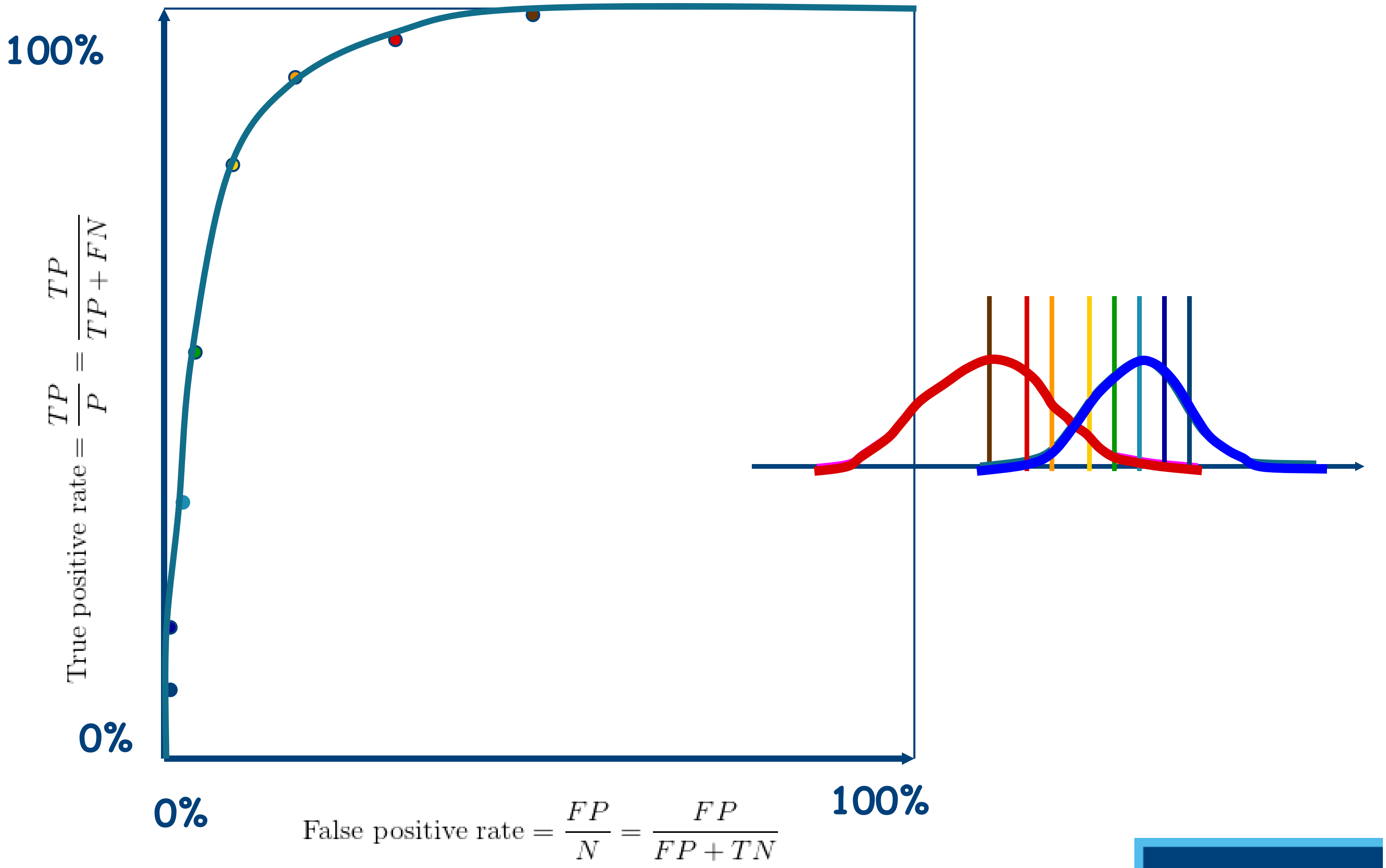
without the disease
with the disease

MOVING THE THRESHOLD: LEFT



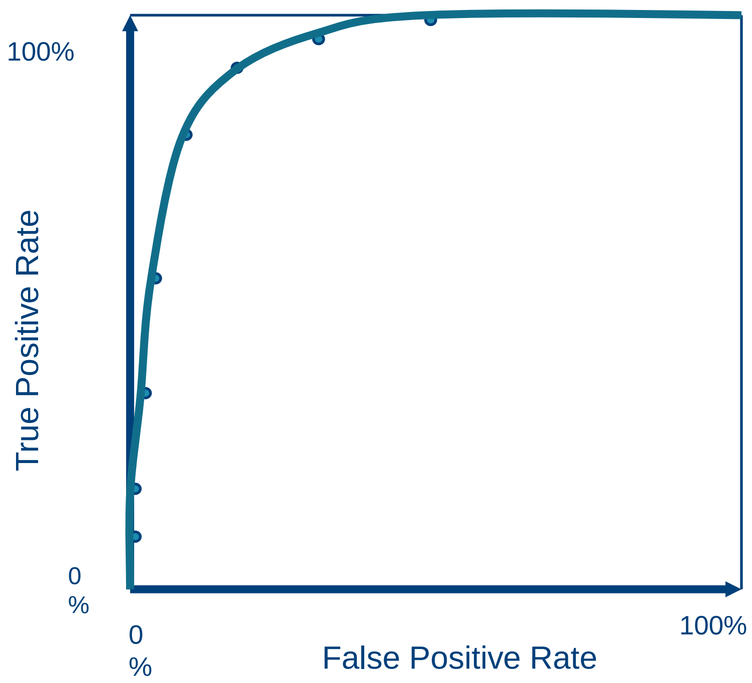
without the disease
with the disease

ROC CURVE

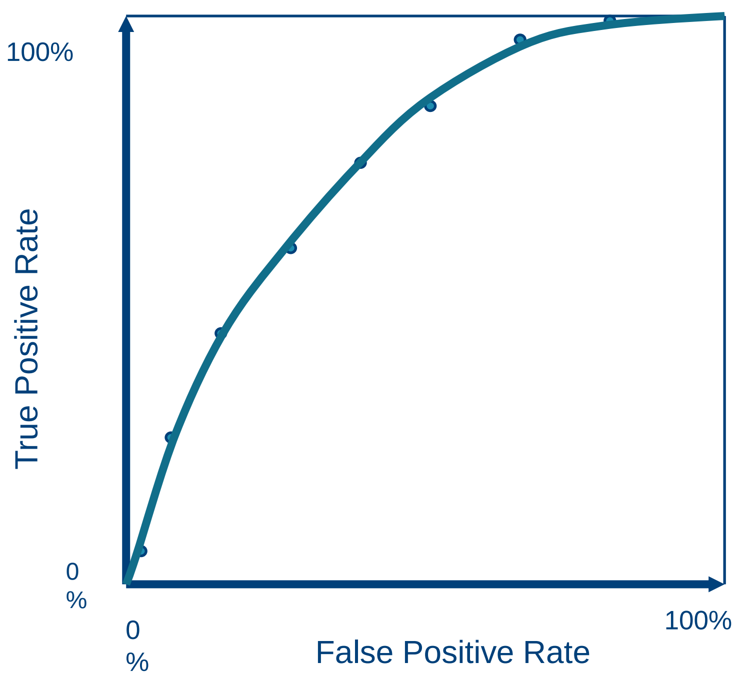

100%
0%
True positive rate $= \frac{TP}{P} = \frac{TP}{TP+FN}$
0%
100%
False positive rate $= \frac{FP}{N} = \frac{FP}{FP+TN}$

ROC CURVE COMPARISON

A good test:



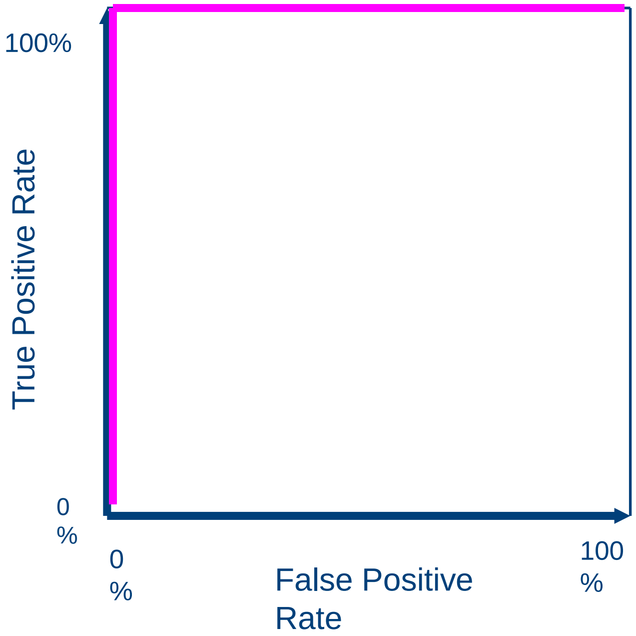
A poor test:



KU LEUVEN

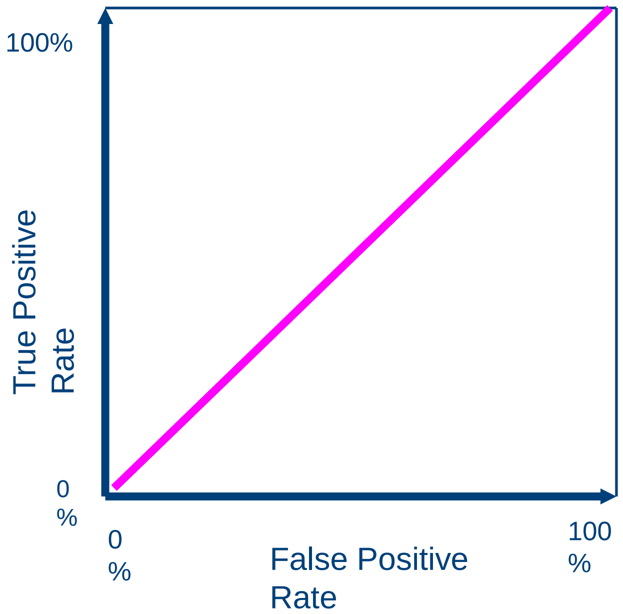
ROC CURVE EXTREMES

Best Test:



The distributions don't overlap at all

Worst test:

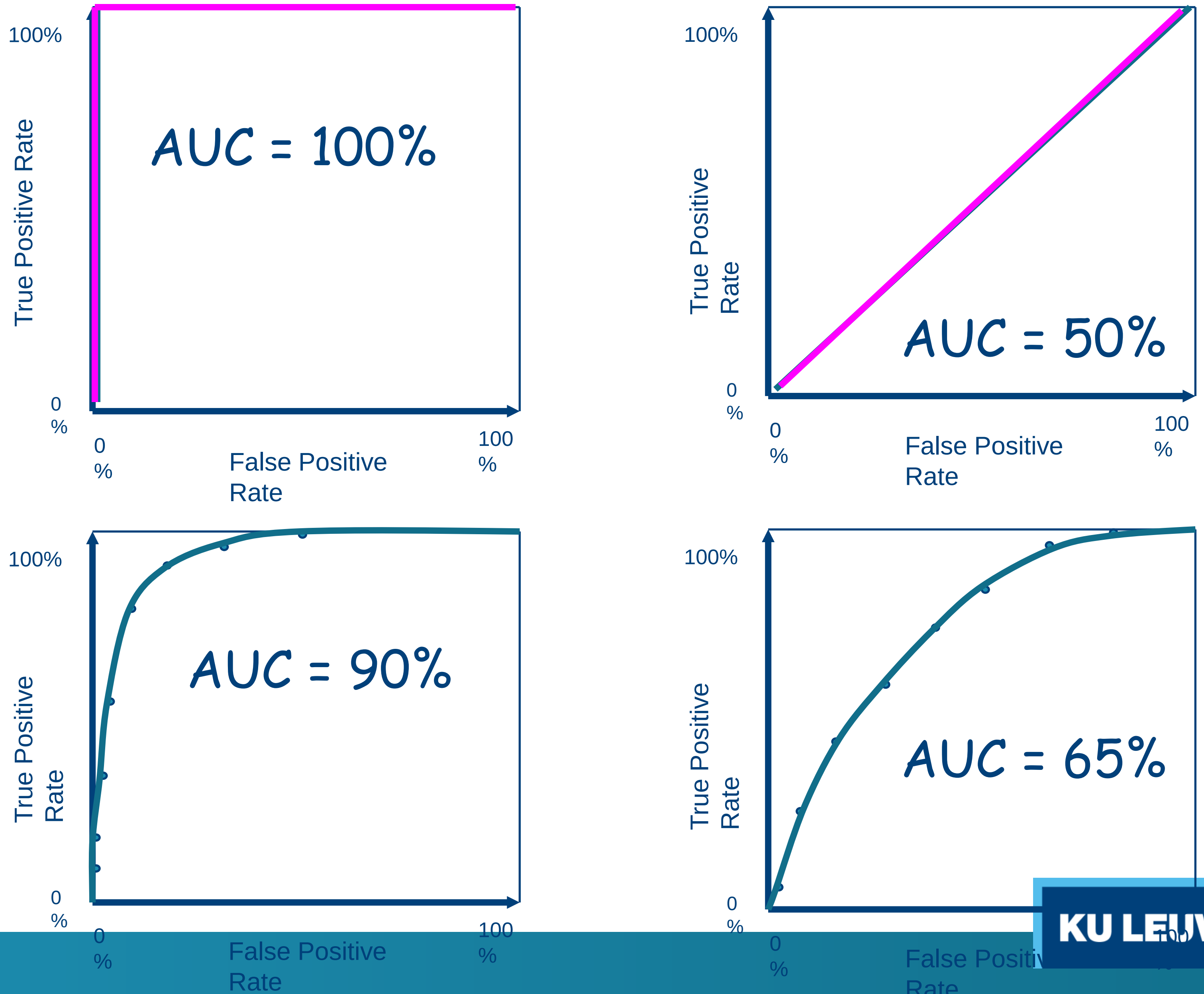


The distributions overlap completely

AREA UNDER ROC CURVE (AUC)

- *Overall measure* of test performance
- *Comparisons* between two tests based on differences between (estimated) AUC
- For continuous data, AUC equivalent to *Mann-Whitney U-statistic* (nonparametric test of difference in location between two populations)

AUC FOR ROC CURVES



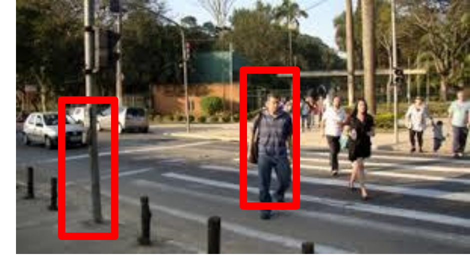
APPLICATION ON OBJECT DETECTORS

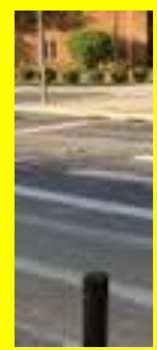
- Detector scans image in a *sliding window* fashion:

· Sliding window over image
· Each sub-window is analyzed by detector

WHAT THE DETECTOR SEES

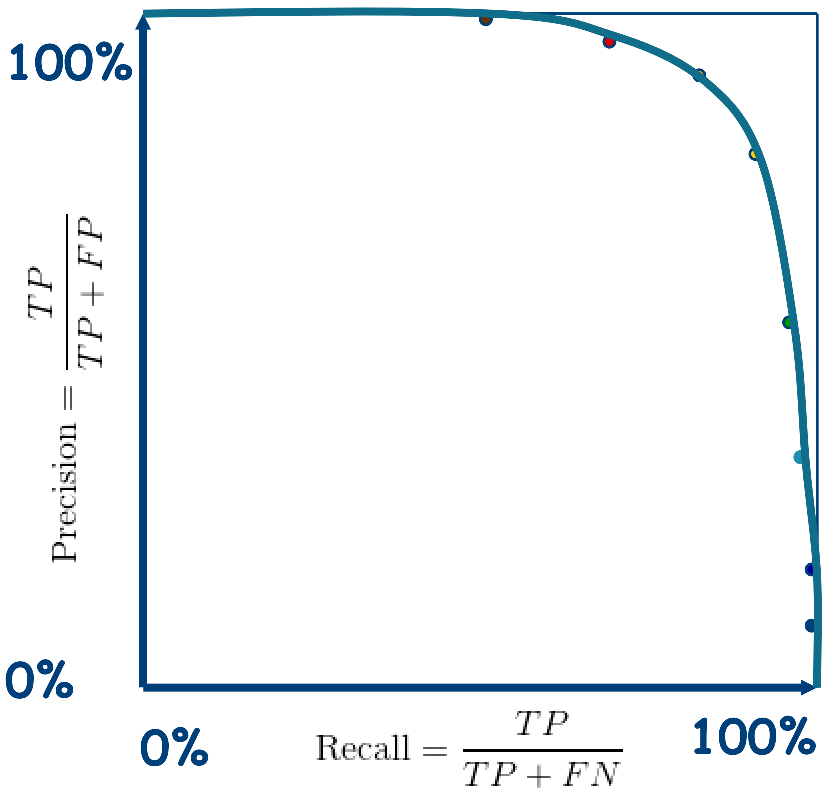
EVALUATING DETECTOR RESULTS

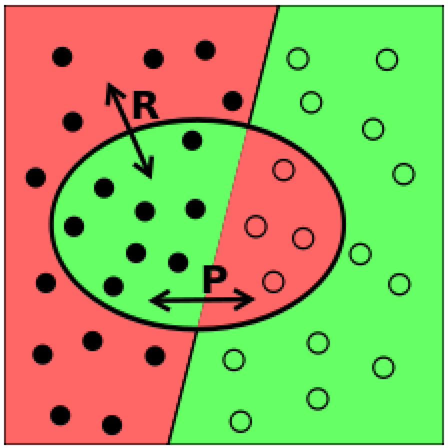


Detector result / Ground Truth	detected	not detected
Object present	 True Positive	 False Negative
Object not present	 False Positive	 True Negative

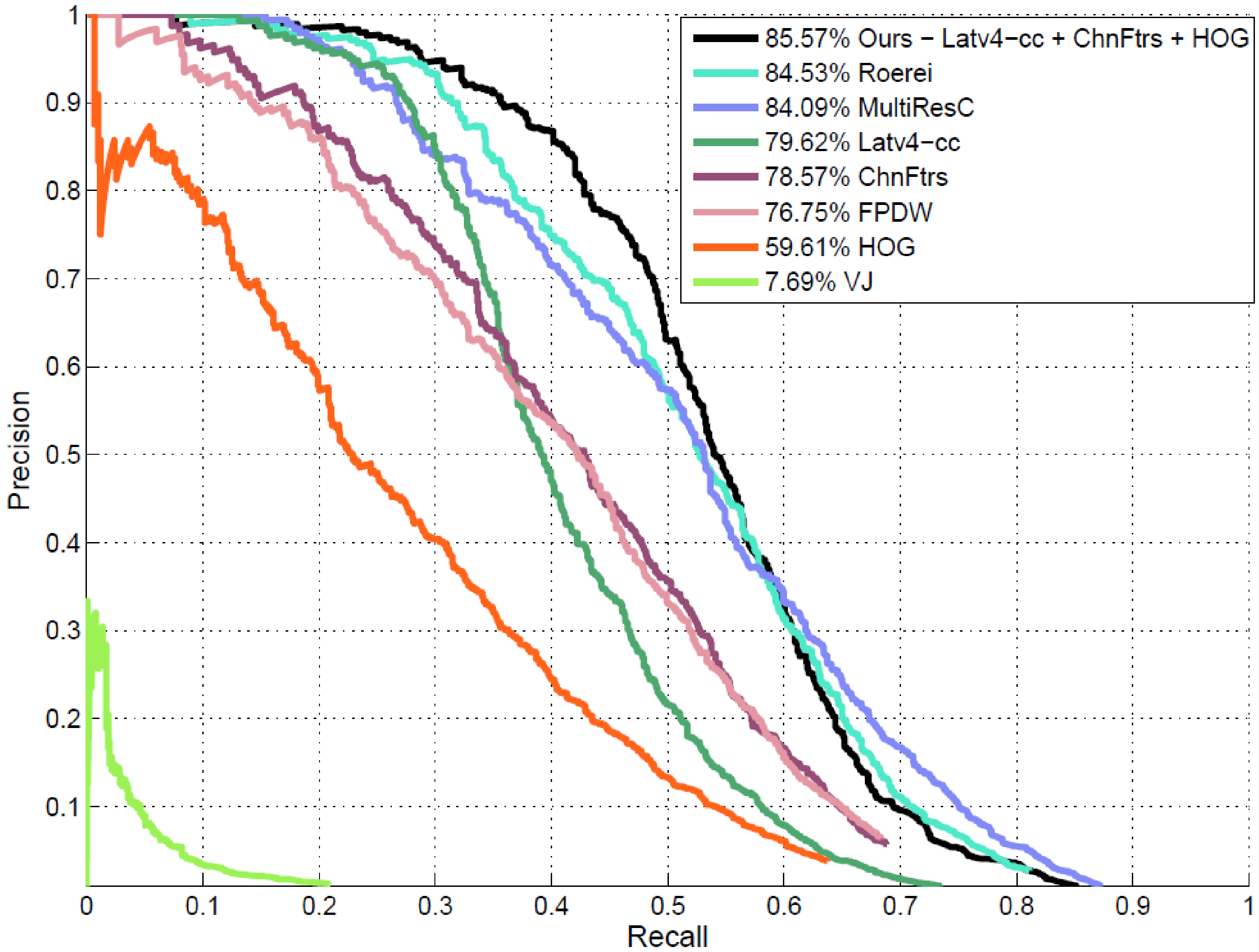
PROBLEM WITH ROC CURVES FOR DETECTORS

- Number of true negatives is not easily countable for images
- Alternative: precision-recall curve



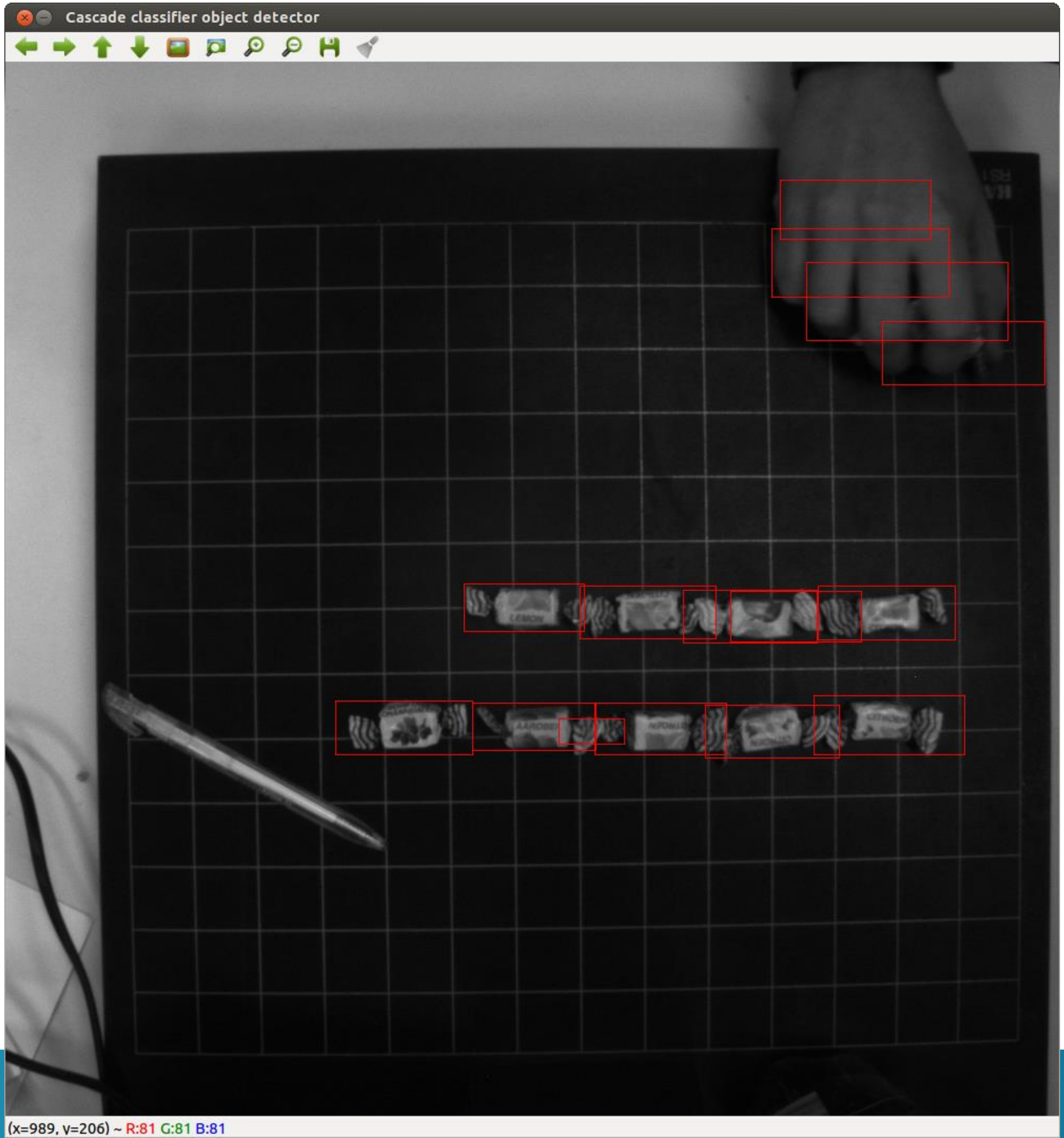


PRECISION-RECALL CURVES FOR PEDESTRIAN DETECTORS



Results of state-of-the-art pedestrian detectors on standard test set “Caltech”

EVALUATION OF A CANDY DETECTOR



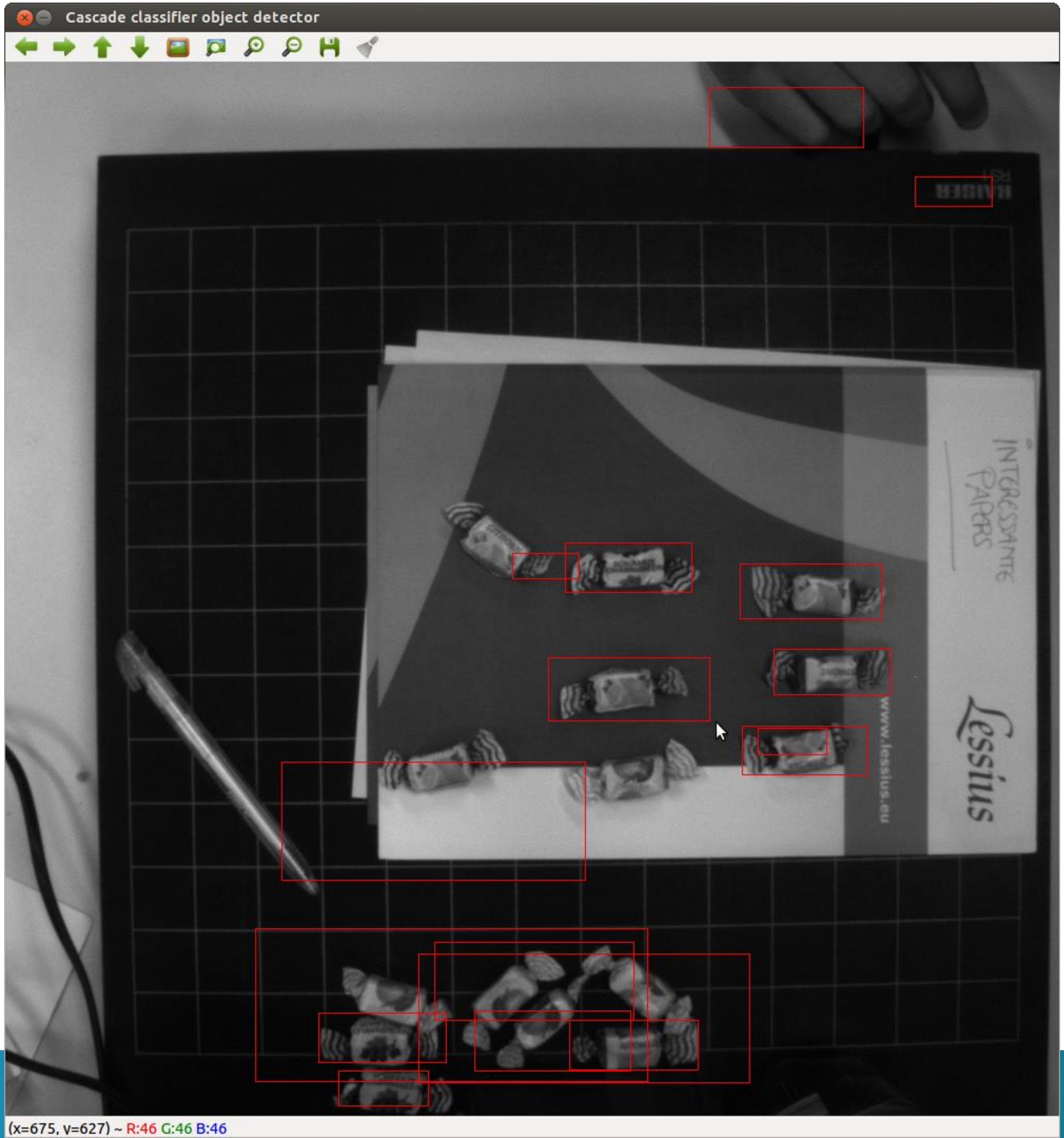
Threshold = 5

TP?
(object detected)

FP?
(detection of a non object)

FN?
(object not detected)

EVALUATION OF A CANDY DETECTOR



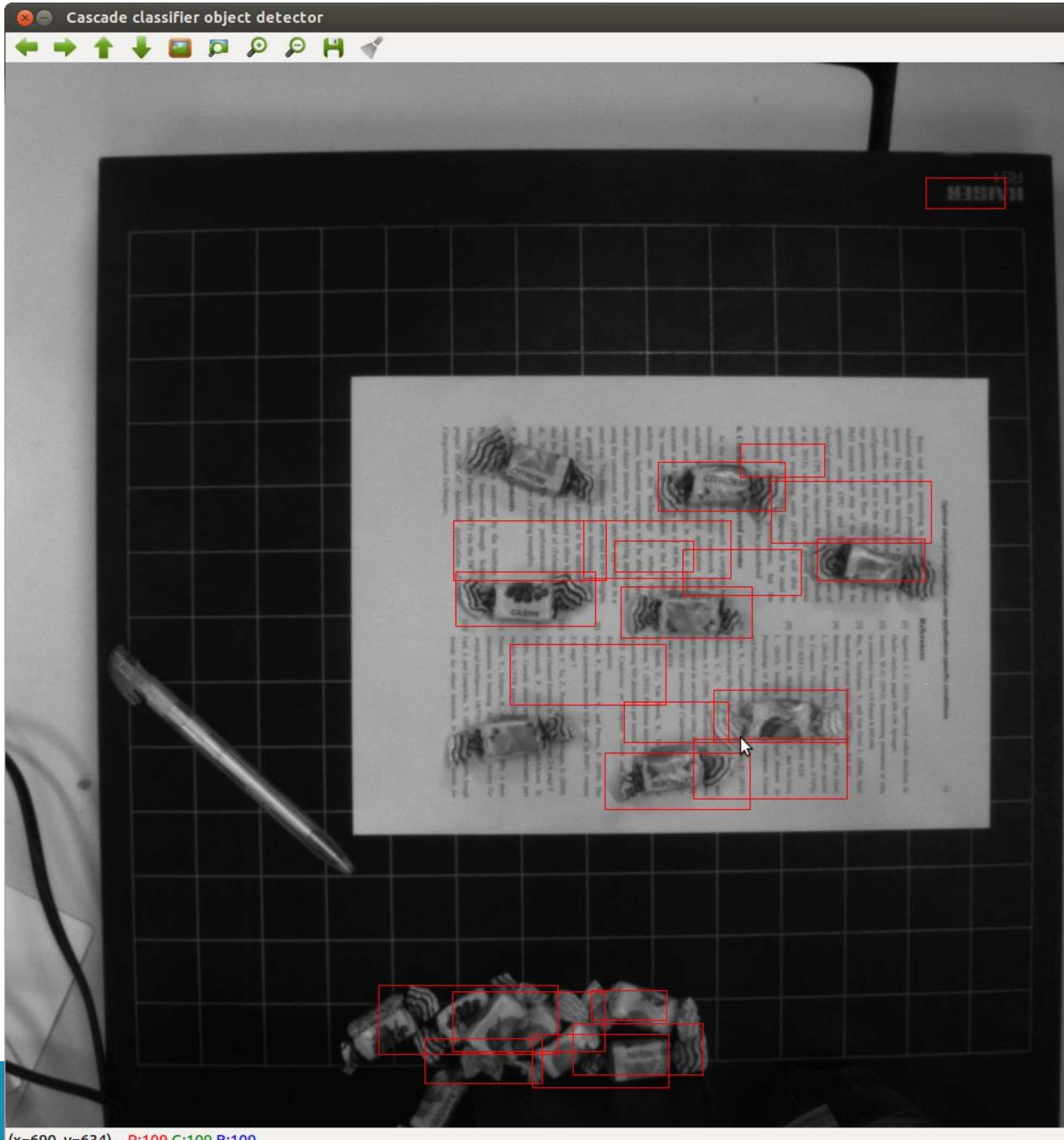
Threshold = 5

TP?
(object detected)

FP?
(detection of a non object)

FN?
(object not detected)

EVALUATION OF A CANDY DETECTOR



Threshold = 5

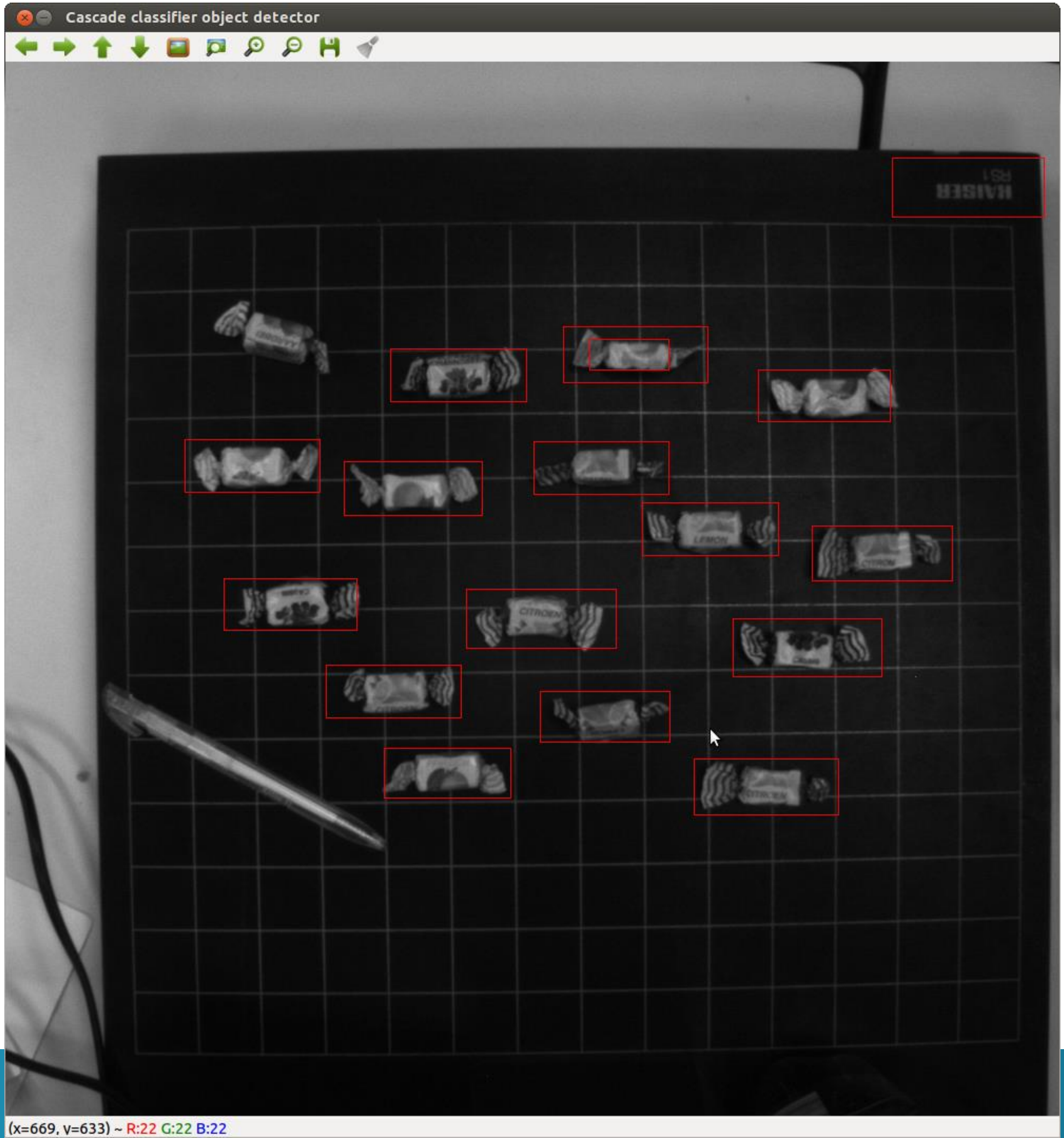
TP?
(object detected)

FP?
(detection of a non object)

FN?
(object not detected)

KU LEUVEN

EVALUATION OF A CANDY DETECTOR



Threshold = 5

TP?
(object detected)

FP?
(detection of a non object)

FN?
(object not detected)

EVALUATION OF A CANDY DETECTOR

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

CONCLUSION

- To evaluate an object detector we need to:
  - Annotate a set of images
  - Train the detector on a subset of those images (training set)
  - Use the remaining images (test set) to calculate the TP, FP & FN rates
  - Follow up by calculating precision & recall values
  - Plot the precision-recall curves based on different threshold values for a parameter
- **Attention OpenCV**: some detectors (e.g. Viola&Jones) don’t give automated scores for each detection, which makes creating PR some a hard taks to do.

PROGRAM

09u30 Welcome and coffee
10u00 Official welcoming & introduction EAVISE
10u15 Introduction object categorization + a look at the algorithm
11u00 Break with coffee
11u15 First hands-on: object annotation tool and preprocessing of the necessary data
12u30 Warm lunch & coffee (sponsered by data vision)
13u30 Second hands-on: a deeper look at the training process, training an object model and testing the actual detector
15u00 Break with refreshments
15u15 Some downsides to the techniques / discussion on the quality of an object detector model
16u15 Questions & evaluation of workshop
16u30 End of workshop