



UNIVERSIDAD  
DE LA RIOJA

**Eurocast 2017**

# **DetectionEvaluationJ**

A tool for measuring the goodness  
of object detection algorithms

C. Domínguez, **M. García**, J. Heras, **A. Inés**, E. Mata, and V. Pascual  
**Department of Mathematics and Computer Science, University of La Rioja, Spain**

# Introduction

Object detection algorithms are applied in diverse computer vision applications, surveillance, traffic monitoring, melanoma detection...

In order to evaluate the quality of those algorithms, we compare the detected Regions of interest (ROIs) with the gold standard using different metrics

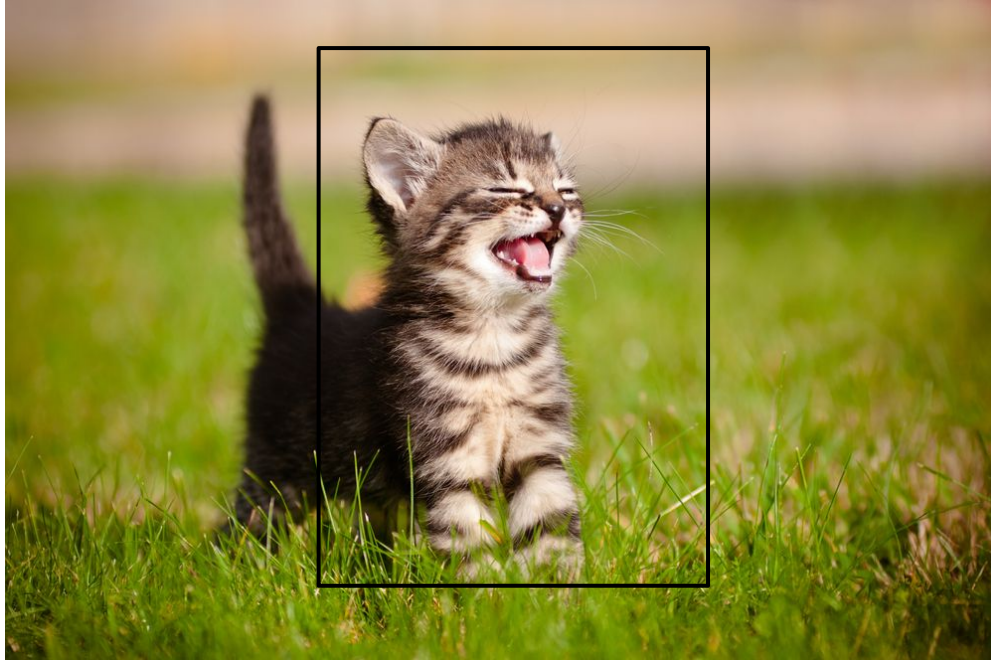


# Introduction



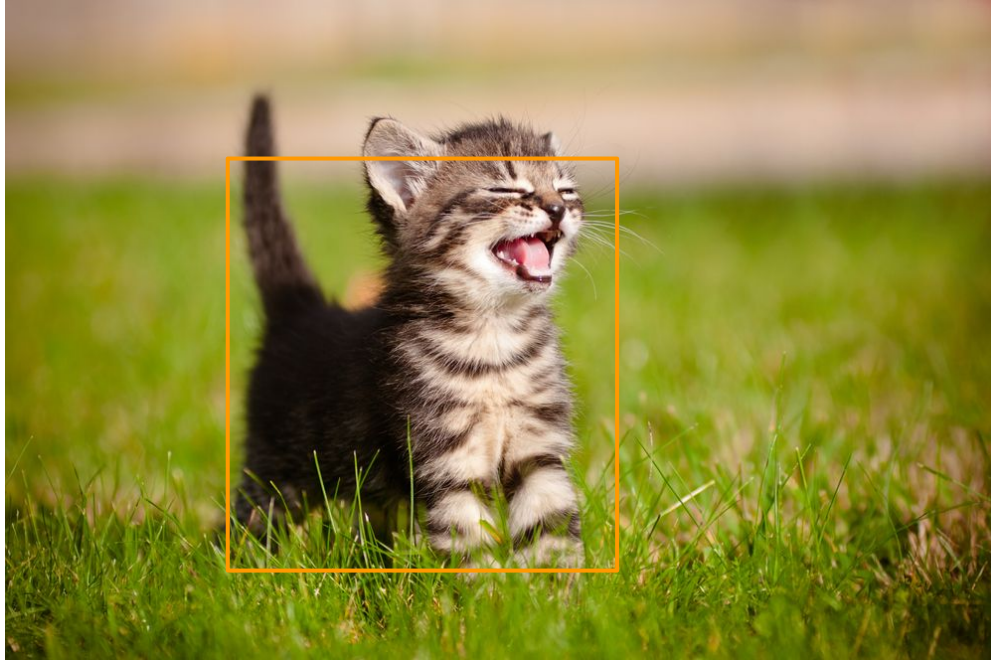
# Introduction

Detection  
algorithm



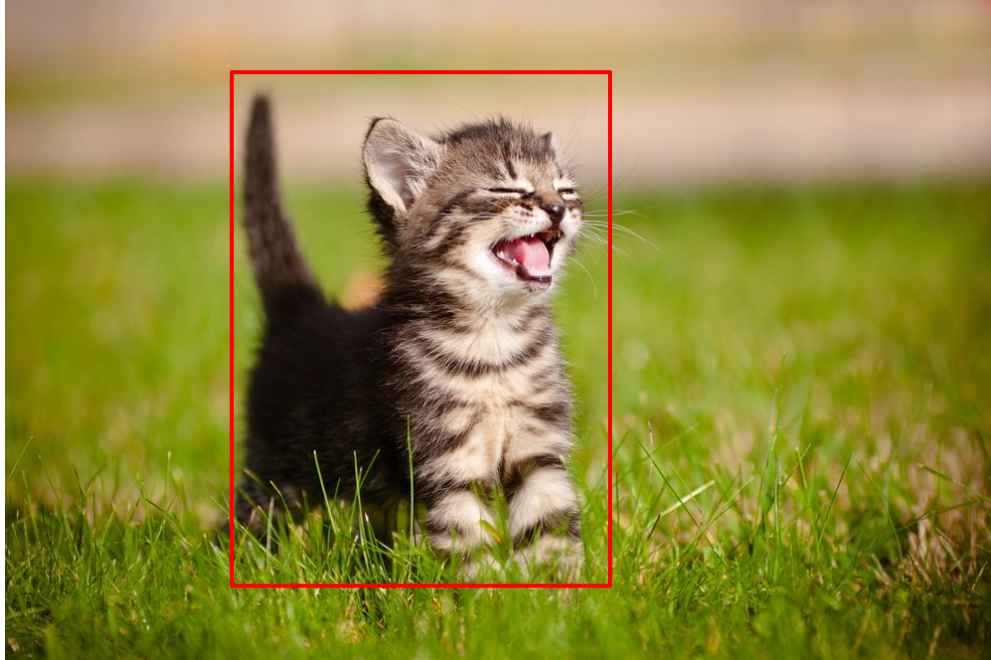
# Introduction

Detection  
algorithm



# Introduction

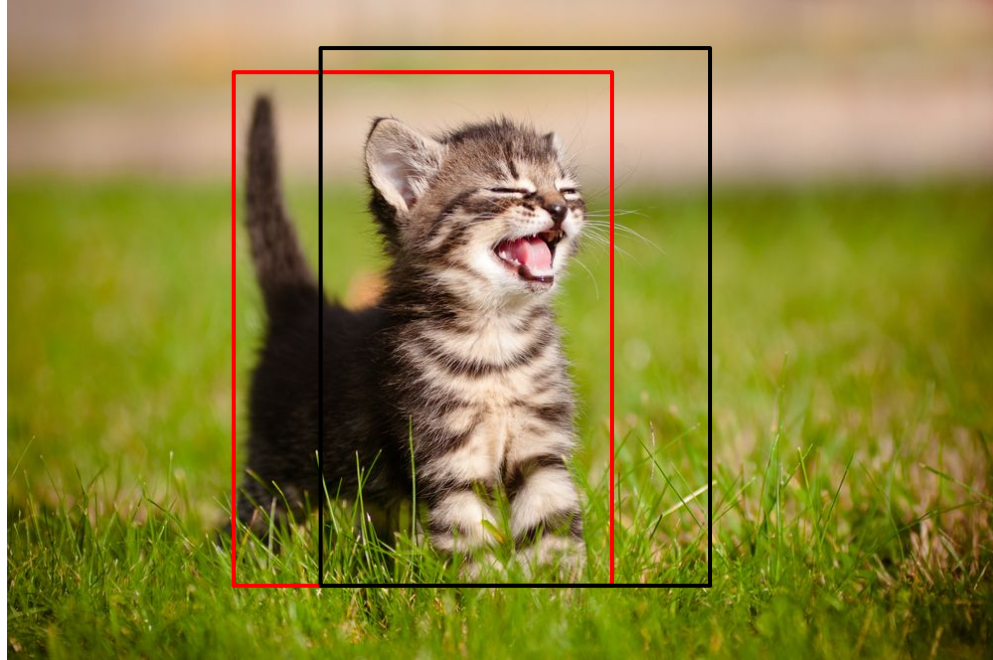
Gold standard



# Introduction

Detection  
algorithm

Gold standard

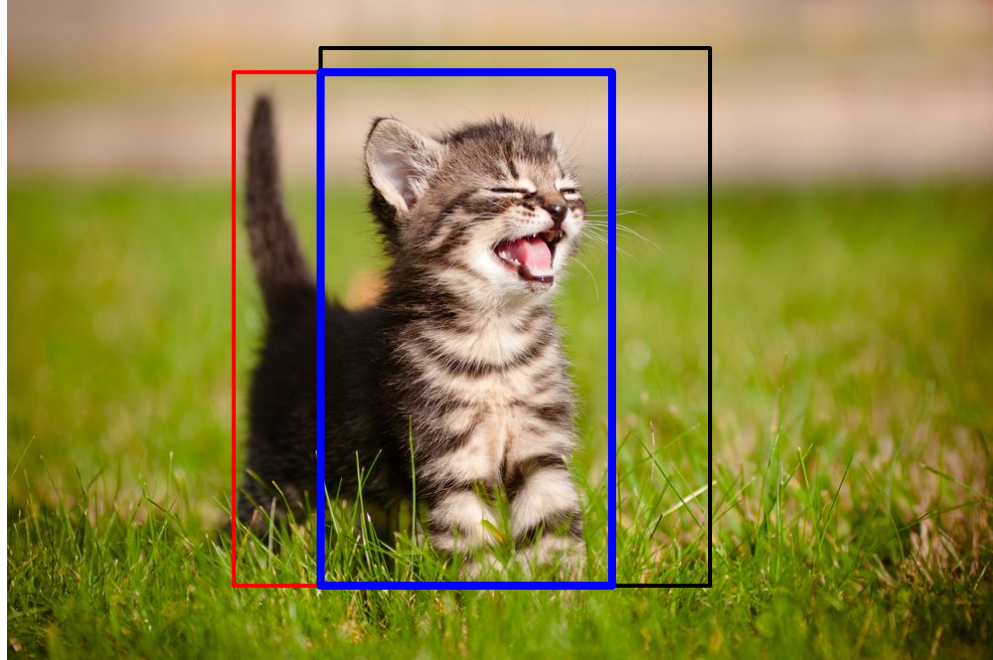


# Introduction

Detection  
algorithm

Gold standard

True positive



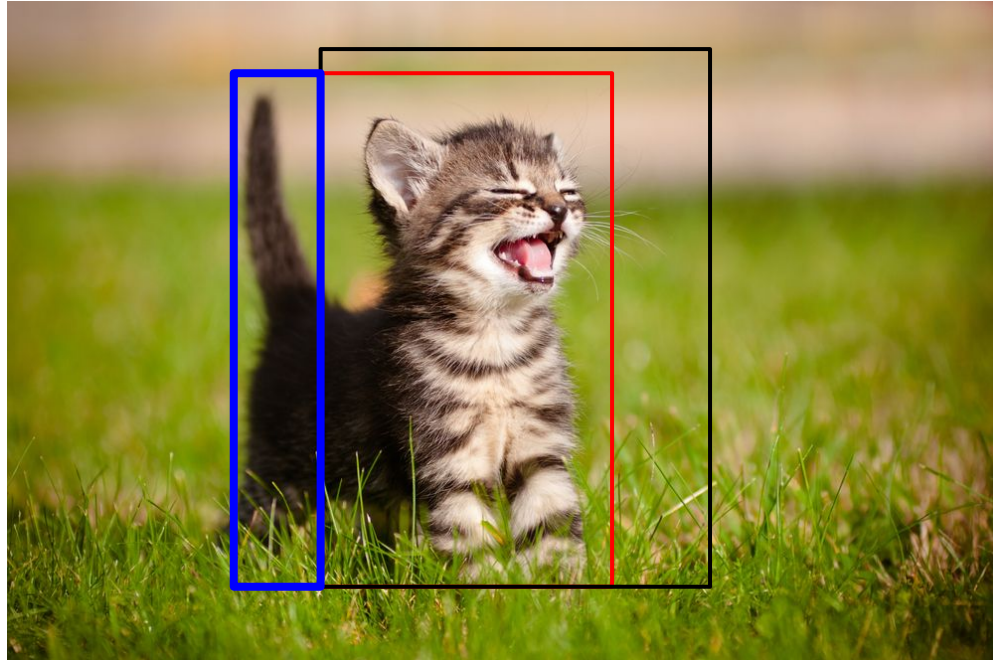


# Introduction

Detection  
algorithm

Gold standard

False negative

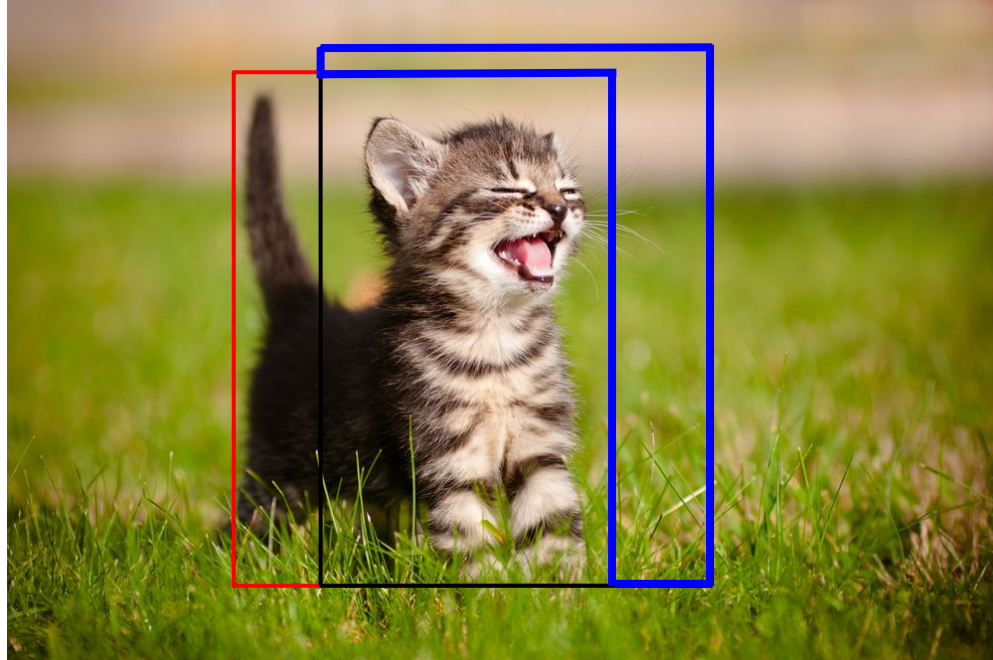


# Introduction

Detection  
algorithm

Gold standard

False positive



## Problems:

- ▶ Measuring the performance of object detection algorithms is a common task
- ▶ There is not a simple tool to carry out this task automatically
- ▶ Measuring the quality of the algorithms manually is not sensible
- ▶ Developing ad hoc tools is not a solution either

## Goal:

- ▶ Develop a simple-to-use tool to evaluate the performance of object detection algorithms using several metrics

# ImageJ

- ▶ Open-source Java-based image processing program
- ▶ Extensible via plugins and recordable macros
- ▶ Available at <http://imagej.net/>



# DetectionEvaluationJ

ImageJ plugin for measuring the performance of object detection algorithms

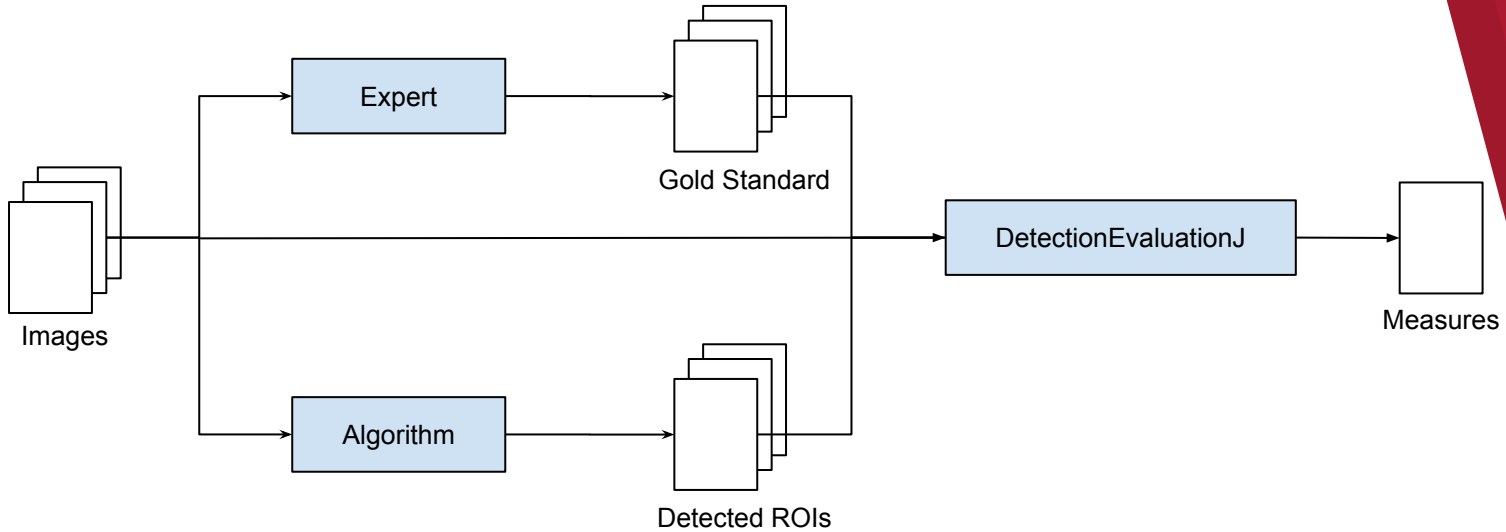
Input:

- ▶ An image (or set of images)
- ▶ The gold standard
- ▶ The ROIs obtained by the detection algorithm

Output:

- ▶ Report that summarises the quality of the detection algorithm based on several metrics

# DetectionEvaluationJ workflow



# DetectionEvaluationJ

DetectionEvaluationJ input:

- ▶ Images
- ▶ Gold Standard: can be fixed, exported and imported using DetectionEvaluationJ
- ▶ Detected ROIs:
  - ▷ ROIs detected with ImageJ
  - ▷ ROIs detected by other programs encoded using ROIXML



## DetectionEvaluationJ results

- ▶ The user can measure the goodness of the detected regions using the pixel-level metrics
- ▶ Several detection algorithms can be loaded to compare their quality
- ▶ This plugin can also be applied to study inter-rater agreement among experts

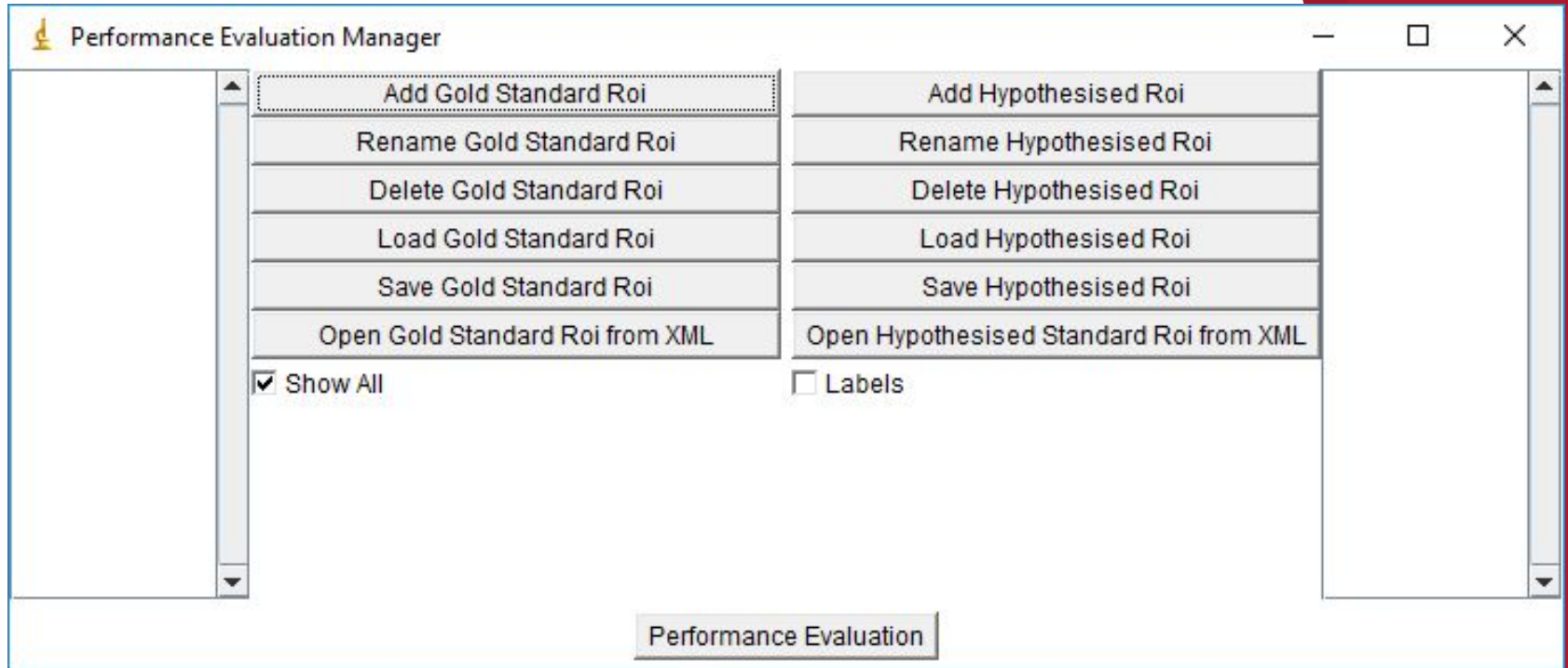
# List of measures

- ▶ Accuracy
- ▶ Precision
- ▶ Recall
- ▶ Fallout
- ▶ Sensitivity
- ▶ Negative predictive value
- ▶ LR+
- ▶ LR-
- ▶ Specificity
- ▶ False negative rate
- ▶ False discovery rate
- ▶ F-measures (0.5, 1, 2)

# List of measures

- ▶ Intersection over union
- ▶ Fowlkes Mallows index
- ▶ Diagnostic odds ratio
- ▶ Balanced accuracy
- ▶ Error rate
- ▶ Youden's J statistic
- ▶ Markedness
- ▶ Matthews correlation coefficient
- ▶ ROC space

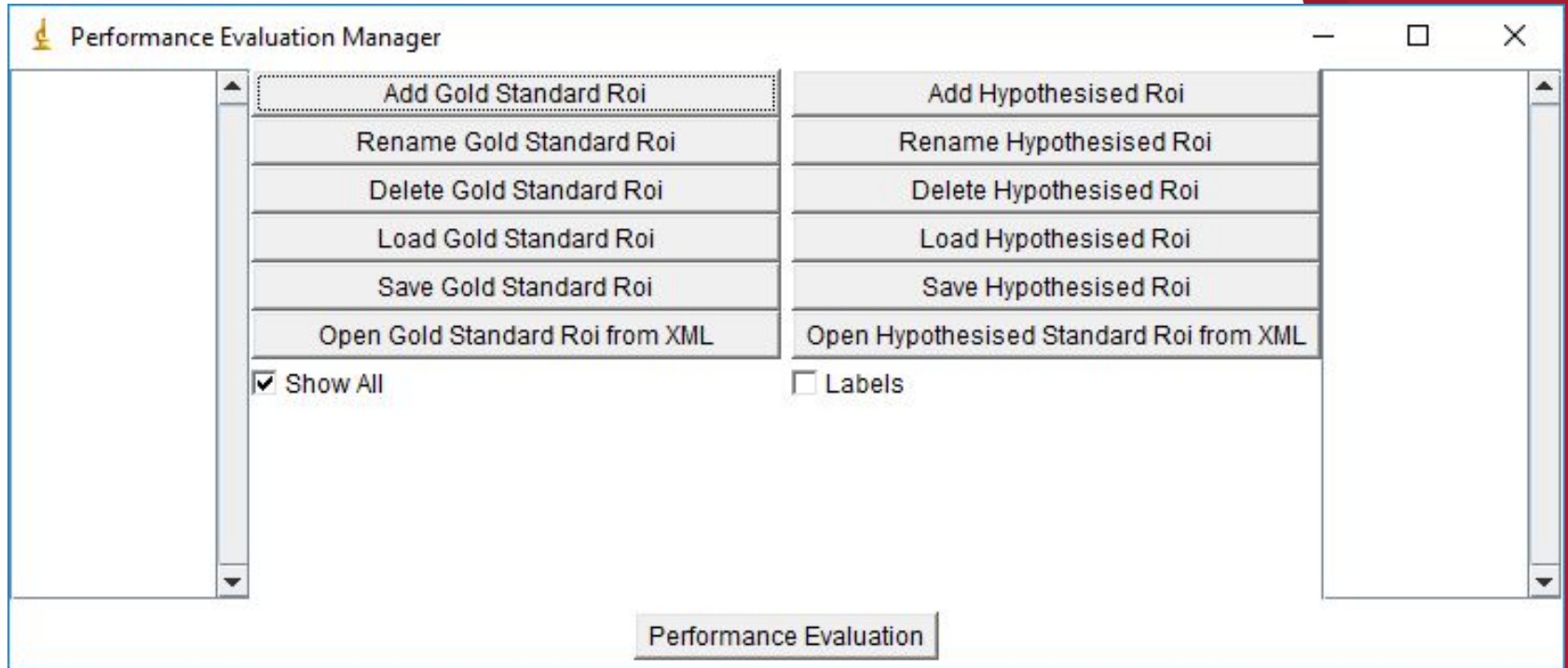
# Example



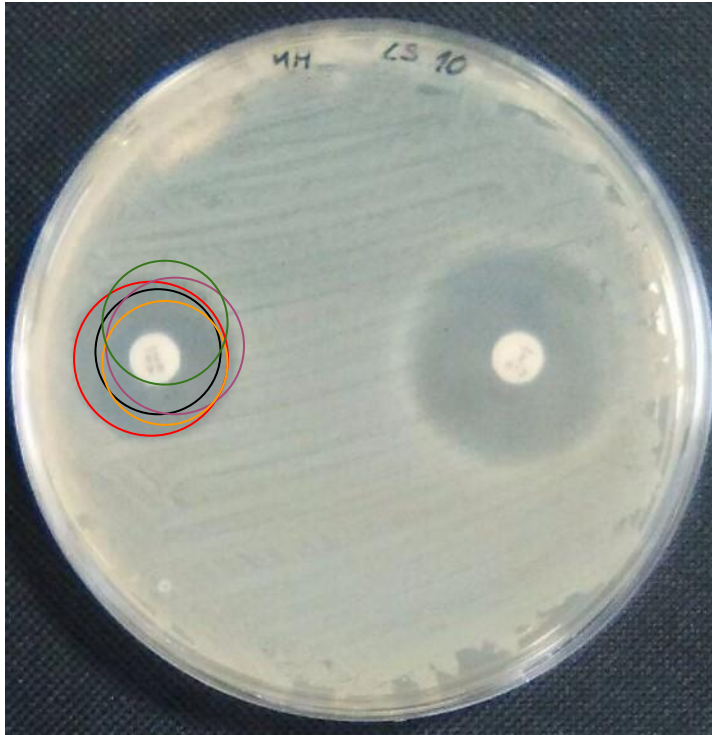
# Example



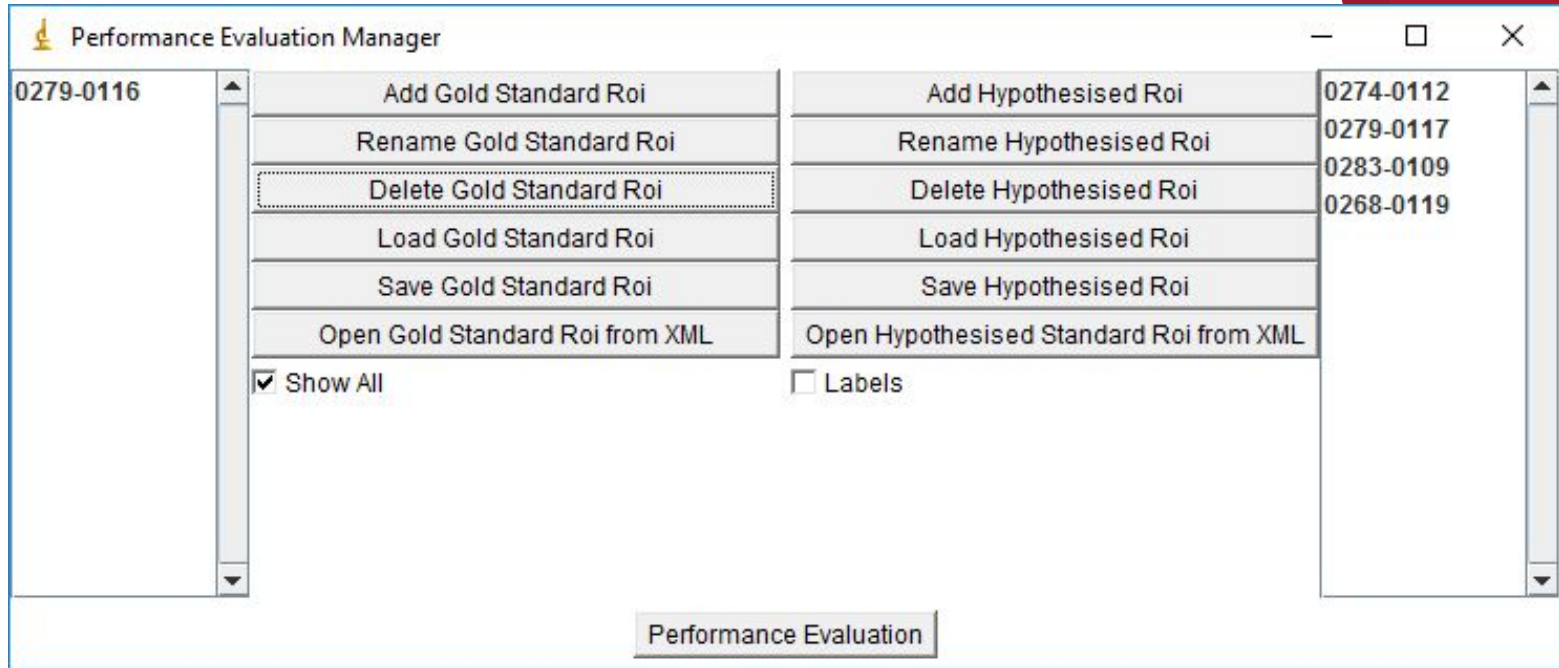
# Example



# Example



# Example





# Example

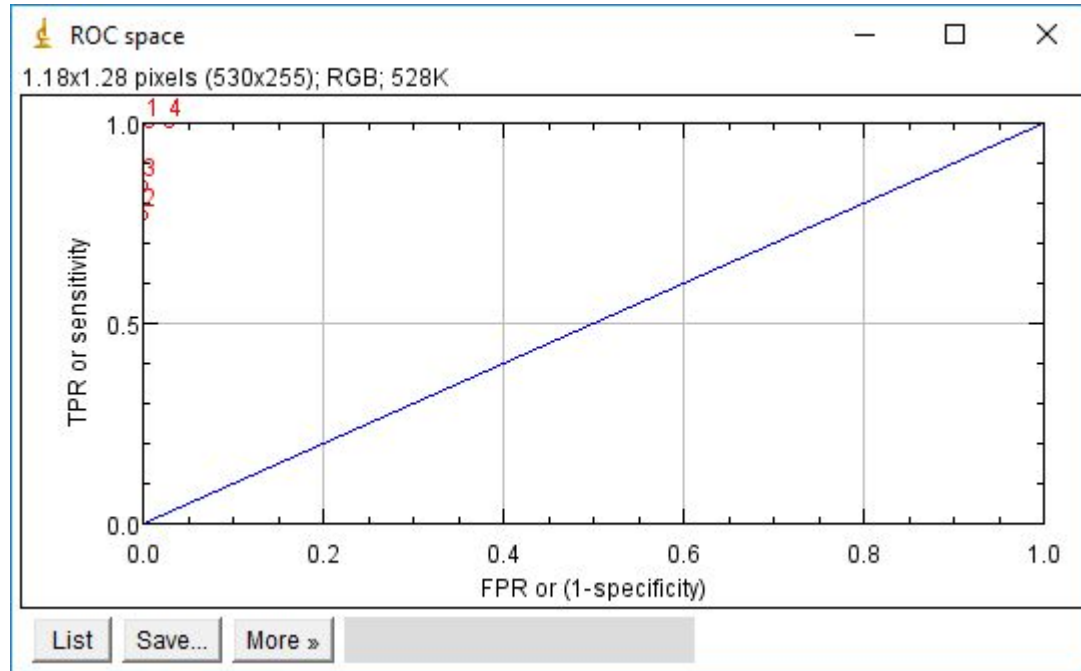
True ROI		Hypothesised ROI
0279-0116	<->	0274-0112
0279-0116	<->	0279-0117
0279-0116	<->	0283-0109
0279-0116	<->	0268-0119

Match regions

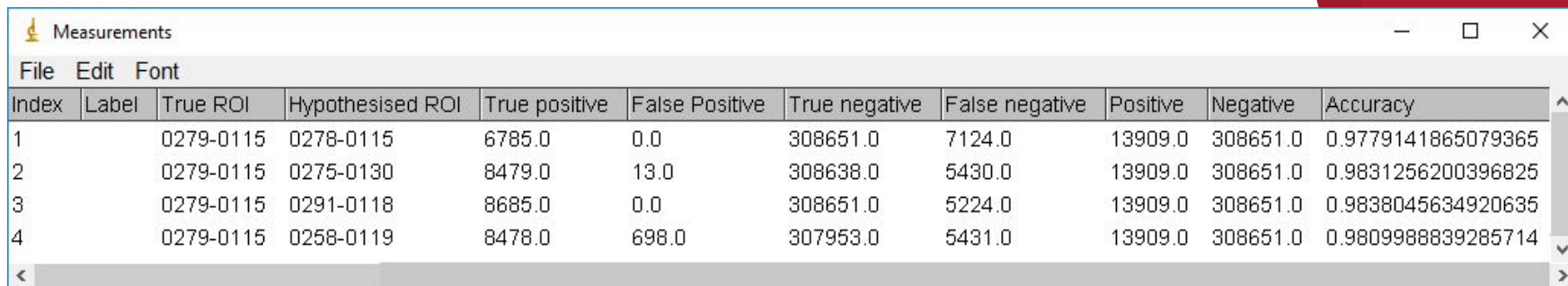
Add results to previous data

Cancel Evaluate

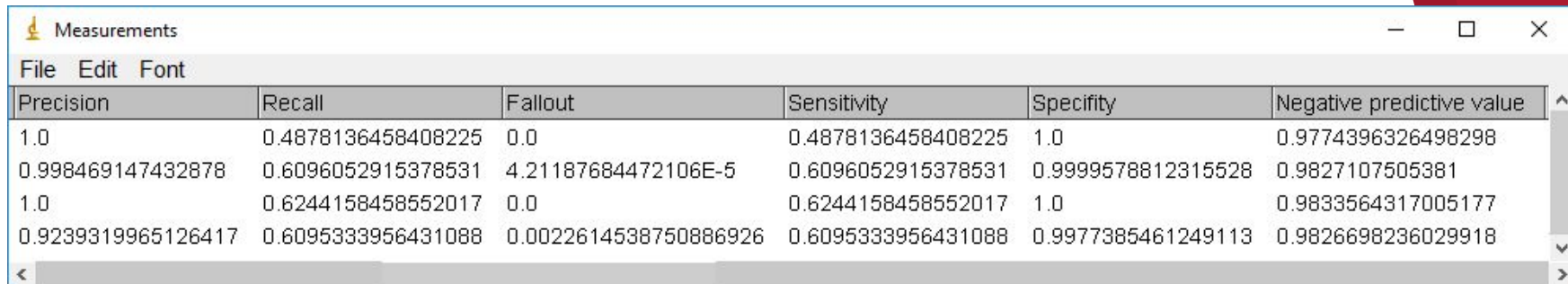
# Example



# Example



Index	Label	True ROI	Hypothesised ROI	True positive	False Positive	True negative	False negative	Positive	Negative	Accuracy
1		0279-0115	0278-0115	6785.0	0.0	308651.0	7124.0	13909.0	308651.0	0.9779141865079365
2		0279-0115	0275-0130	8479.0	13.0	308638.0	5430.0	13909.0	308651.0	0.9831256200396825
3		0279-0115	0291-0118	8685.0	0.0	308651.0	5224.0	13909.0	308651.0	0.9838045634920635
4		0279-0115	0258-0119	8478.0	698.0	307953.0	5431.0	13909.0	308651.0	0.9809988839285714



Precision	Recall	Fallout	Sensitivity	Specifity	Negative predictive value
1.0	0.4878136458408225	0.0	0.4878136458408225	1.0	0.9774396326498298
0.998469147432878	0.6096052915378531	4.21187684472106E-5	0.6096052915378531	0.9999578812315528	0.9827107505381
1.0	0.6244158458552017	0.0	0.6244158458552017	1.0	0.9833564317005177
0.9239319965126417	0.6095333956431088	0.0022614538750886926	0.6095333956431088	0.9977385461249113	0.9826698236029918

# Conclusions

Measuring the performance of object detection algorithms is a common problem in computer vision

DetectionEvaluationJ is an open source ImageJ plugin that solves this problem

# Where can we find this plugin?

DetectionEvaluationJ is freely available at [joheras.github.io/DetectionEvaluationJ/](https://joheras.github.io/DetectionEvaluationJ/)

**Questions?**