Detection Suite: toolbox to evaluate object detection solutions based on NN

Sergio Paniego Blanco
MUVA, URJC, 3-3-2020

Universidad
Rey Juan Carlos

PRESENTATION

- URJC student.
- JdeRobot member.
- GSoC student developer (2018) and mentor (2019).
- AI learner.
- github.com/JdeRobot/DetectionSuite
- jderobot.github.io/DetectionSuite/

OBJECTIVES

- **Enumerate the vast amount of options in DL and the fast-paced development across the field.**
- **Introduce Detection Suite toolbox.**
- **Problems and solutions in a DL project.**
- **Provide real-world experiment.**

INDEX

INDEX

INTRODUCTION

- **Toolbox for object detection (classification + location).**
- **Open Source project**
- **Simplify development and testing of solutions based on object detection.**
- **Trial and error in deep learning networks development.**
- **Need for objective comparison between different approaches.**

INTRODUCTION

- 2 times accepted project in GSoC with JdeRobot:
  - Vinay Sharma (2018).
  - Jeevan Kumar (2019).
- Research paper:
  - Vinay Sharma .
  - José María Cañas.
- International collaboration.

INDEX

- INTRODUCTION
- STATE OF THE ART
- DETECTIONSUITE TOOLBOX
- EXPERIMENTS
- CONCLUSIONS

STATE OF THE ART

- **DATASETS**
- **FRAMEWORKS**
- **NETWORK MODELS**

DATASETS

- **IMAGENET.**
- **Dataset and annual competition.**
- **ImageNet Large Scale Visual Recognition Challenge (ILSVRC).**
- **1000 object classes (200 in object detection).**
- **1.5M images.**
- **Milestones: AlexNet (A. Krizhevsky, et al. 2012), ResNet (K. He, et al. 2015).**

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg & L. Fei-Fei . ImageNet Large Scale Visual Recognition Challenge. *IJCV, 115.* . Springer, 2015, pp. 211-252.

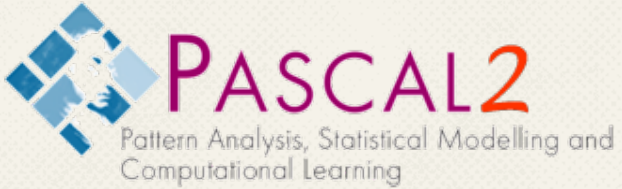
DATASETS

- **COCO DATASET (Tsung-Yi Lin et al. 2015).**
- **Designed for object detection and segmentation.**
- **~90 categories.**
- **Detection of objects in natural situations .**
- **~330.000 images. 2.500.000 labelled instances.**
- **Fewer categories, but more instances per category.**
- **Majority of non-iconic images.**
- **Flickr images**
- **minival**

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: Common objects in context. *European conference on computer vision*. Springer, 2014, pp. 740-755.

DATASETS

- **PASCAL VOC challenge.**
- **Dataset and annual competition.**
- **~20.000 images.**
- **20 object categories.**
- **~25.000 object instances.**
- **Flickr images.**



M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision, 88.* . Springer, 2010, pp. 303-338.

DATASETS

- **PRINCETON RGB DATASET.**
- **~10000 images.**
- **Focused on tracking.**

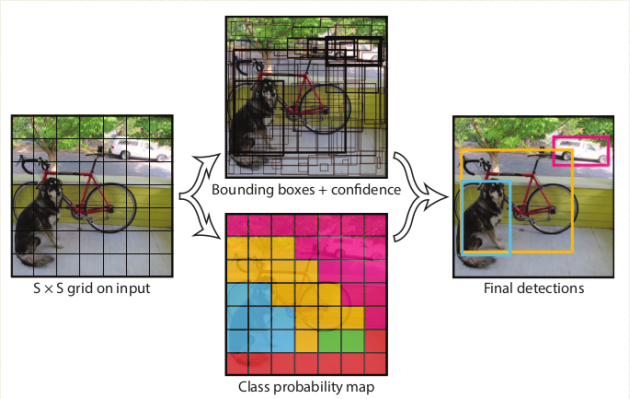
- **SPINELLO DATASET.**
- **~3000 images captured using Kinetic.**
- **Focused on person detection and tracking.**

FRAMEWORKS

- **Tensorflow.**
- **Keras.**
- **Darknet (J. Redmon).**
- **Pytorch.**
- **Caffe.**
- **…**

NETWORK MODELS

- Vast amount of network models. Different approaches to address the problem.
- FASTER REGIONAL-CNN (S. Ren et al. 2016).
  - Regional proposal network (RPN).
  - Improve over R-CNN and Fast R-CNN.
- SINGLE SHOT MULTIBOX DETECTOR (SSD) (Liu et al. 2016).
  - Detections can be generated in a single forward propagation.
- YOU ONLY LOOK ONCE (Redmon et al. 2016).
  - Sees the entire image.
  - YOLO v3.



INDEX

- INTRODUCTION
- STATE OF THE ART
- DETECTIONSUITE TOOLBOX
- EXPERIMENTS
- CONCLUSIONS

DETECTION SUITE

- **Toolbox to work with object detection NN and datasets.**
- **Pretrain your model, test it with Detection Suite.**
- **GUI or CLI, with configuration file.**
- **Broad framework and datasets support.**
- **Linux and macOS support.**
- **Written mainly in C++ and Python.**

IMAGENET

Keras

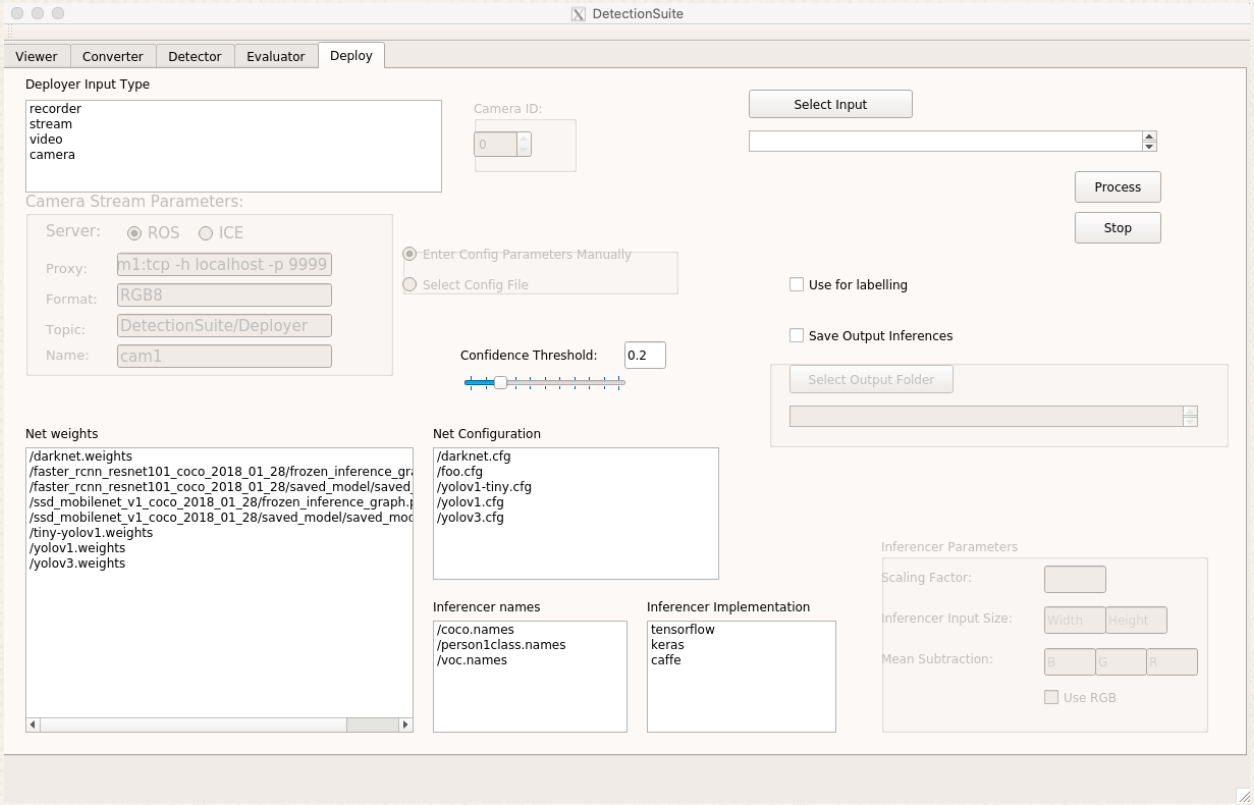
Caffe

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

TOOLBOX

- **VIEWER**
- **DETECTOR**
- **EVALUATOR**
- **DEPLOYER**
- **COMMAND LINE INTERFACE (AUTO EVALUATOR)**
- **CONVERTER**
- **LABELLING**
- **ROS NODE**

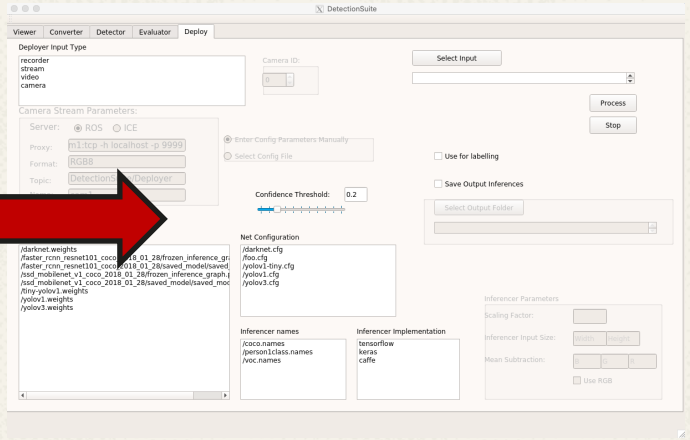
DETECTION SUITE



CONFIG FILE

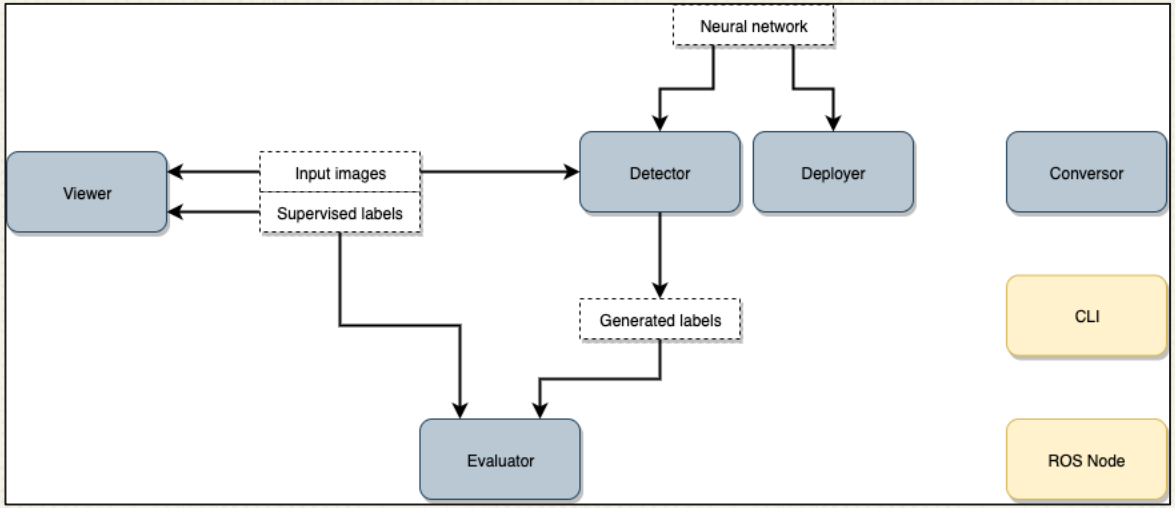
- **Two types of config files.**
- **Config file example (.yml):**

```
datasetPath: /opt/datasets/

evaluationsPath: /opt/datasets/eval

weightsPath: /opt/datasets/weights

netCfgPath: /opt/datasets/cfg

namesPath: /opt/datasets/names

inferencesPath: /opt/datasets
```



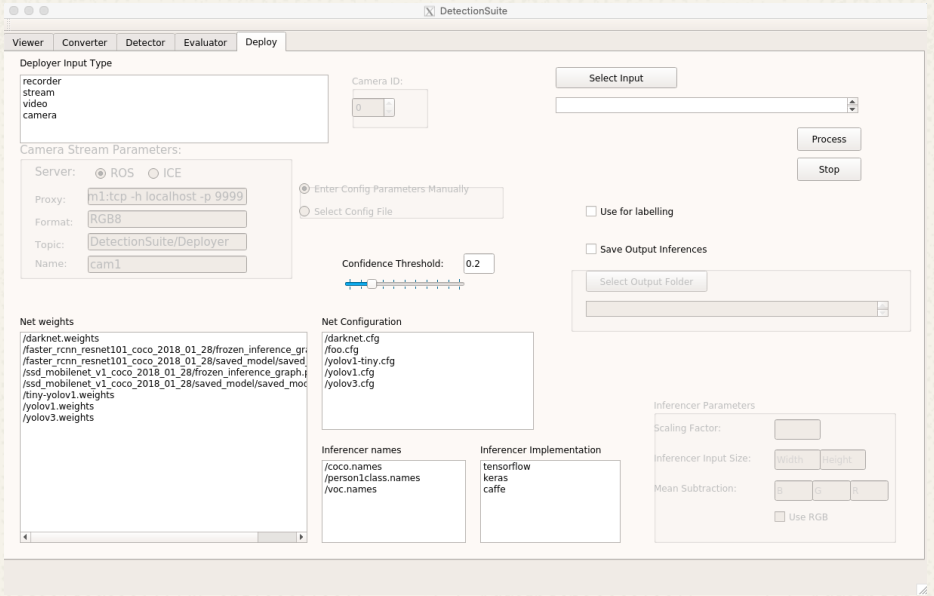
GENERAL ARCHITECTURE

- **Main (blue) and additional (yellow) modules.**



DETECTION SUITE

- Each tool has its own tab.
- Separated modules.
- GPU support.
- Set of resources available in the webpage:
  - Class names.
  - Model zoo.



FRAMEWORKS SUPPORT

- **Dependency with frameworks installed.**
- **Framework evaluators: connection with target DL framework.**
- **Each framework neural network format.**
- **Dependency with configuration files.**

DATASETS SUPPORT

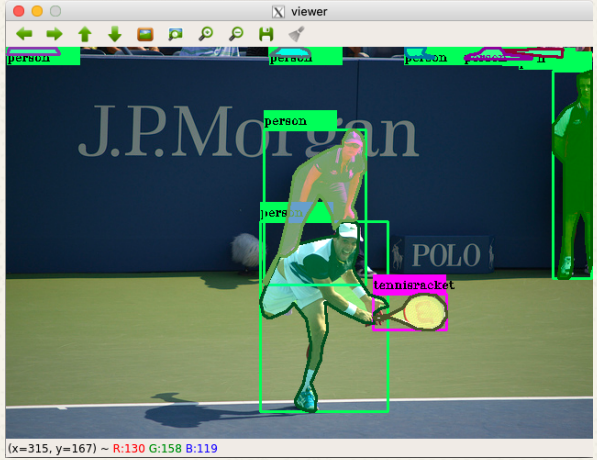
- **Dataset readers and writers for every option supported.**
- **Own dataset format.**
- **COCO dataset format vs. Pascal VOC dataset format.**
- **JSON file with annotations in COCO.**
- **Divided in:**
  - **Info.**
  - **Licenses.**
  - **Categories.**
  - **Images.**
  - **Annotations.**
- **XML file for each image in Pascal VOC.**

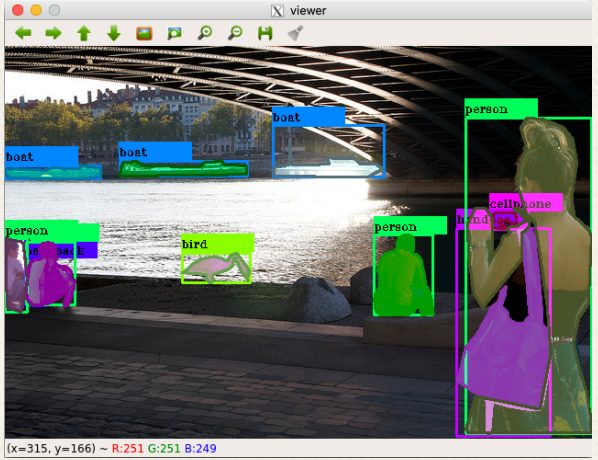
TOOLBOX

- **VIEWER**
- **DETECTOR**
- **EVALUATOR**
- **DEPLOYER**
- **COMMAND LINE INTERFACE (AUTOEVALUATOR)**
- **CONVERTER**
- **LABELLING**
- **ROS NODE**

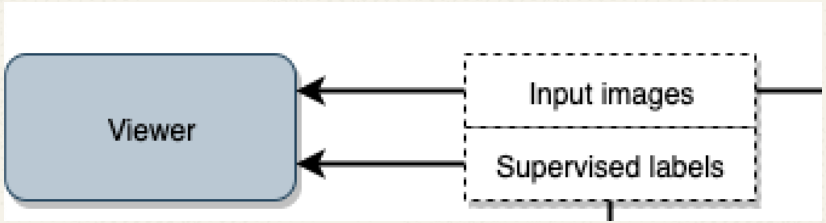
VIEWER

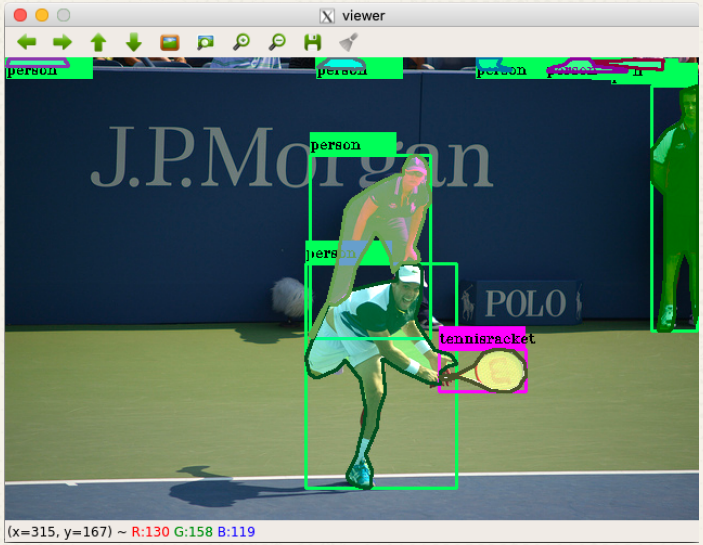
- Read annotations files and displays it.
- Provides general statistics about the dataset: number of elements/class, total images, total samples.
- Supports previously presented datasets.
- CLI available.

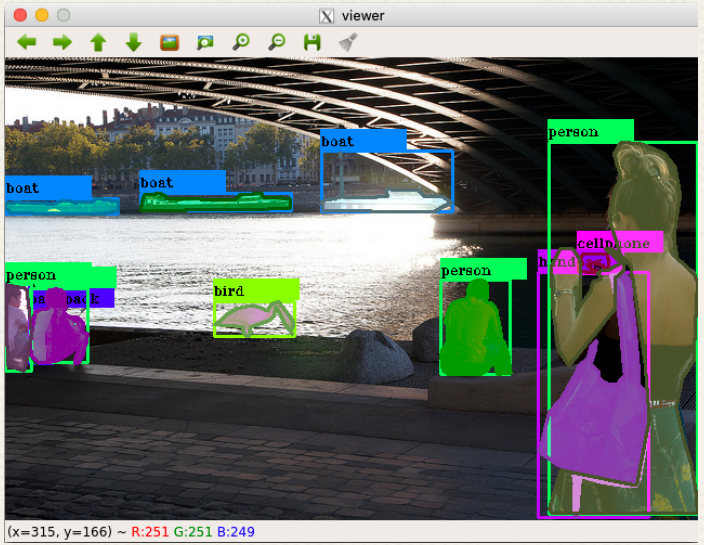




VIEWER

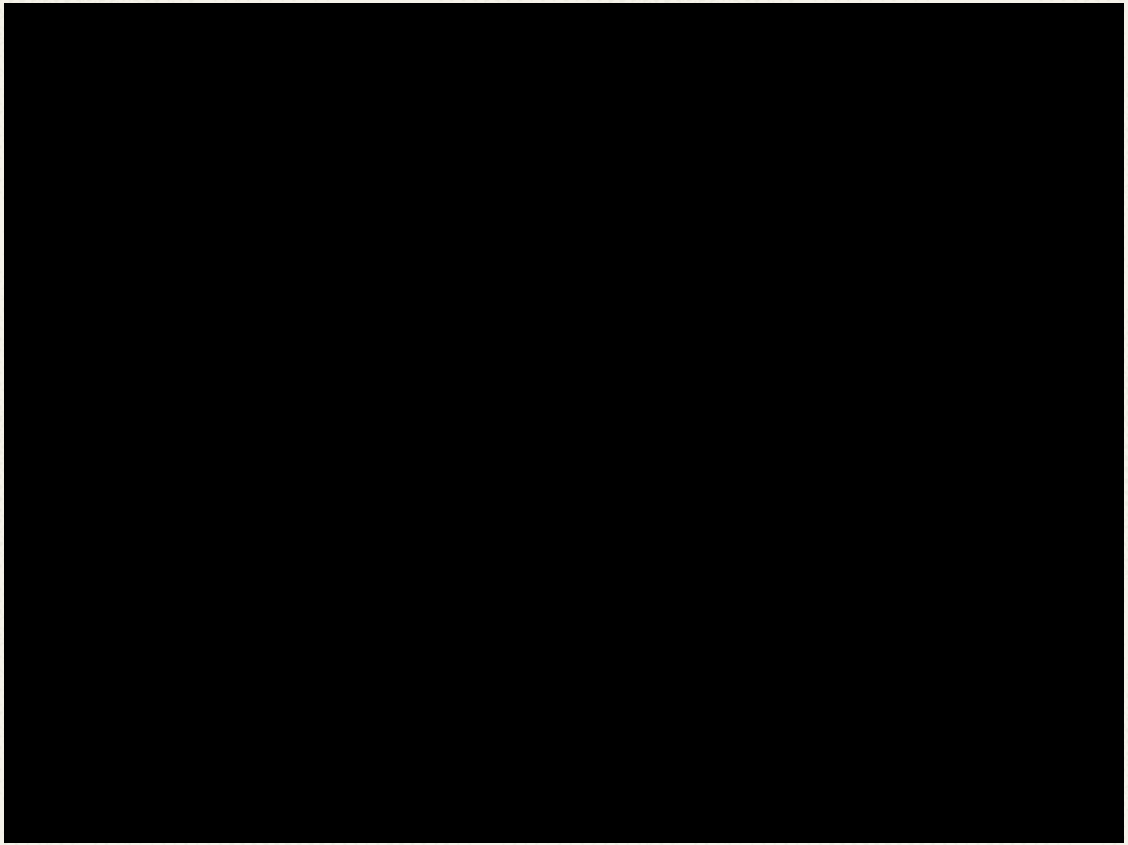

Viewer
Input images
Supervised labels


viewer
person
tennisracket
(x=315, y=167) ~ R:130 G:158 B:119


viewer
boat
person
bird
(x=315, y=166) ~ R:251 G:251 B:249

https://www.youtube.com/watch?v=VMd6ve8brTE

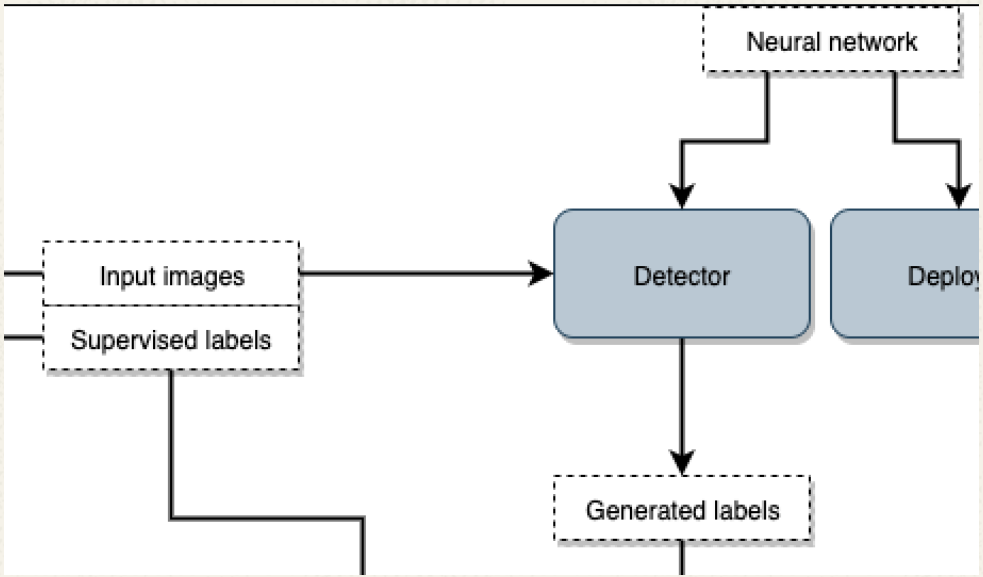
VIEWER



DETECTOR

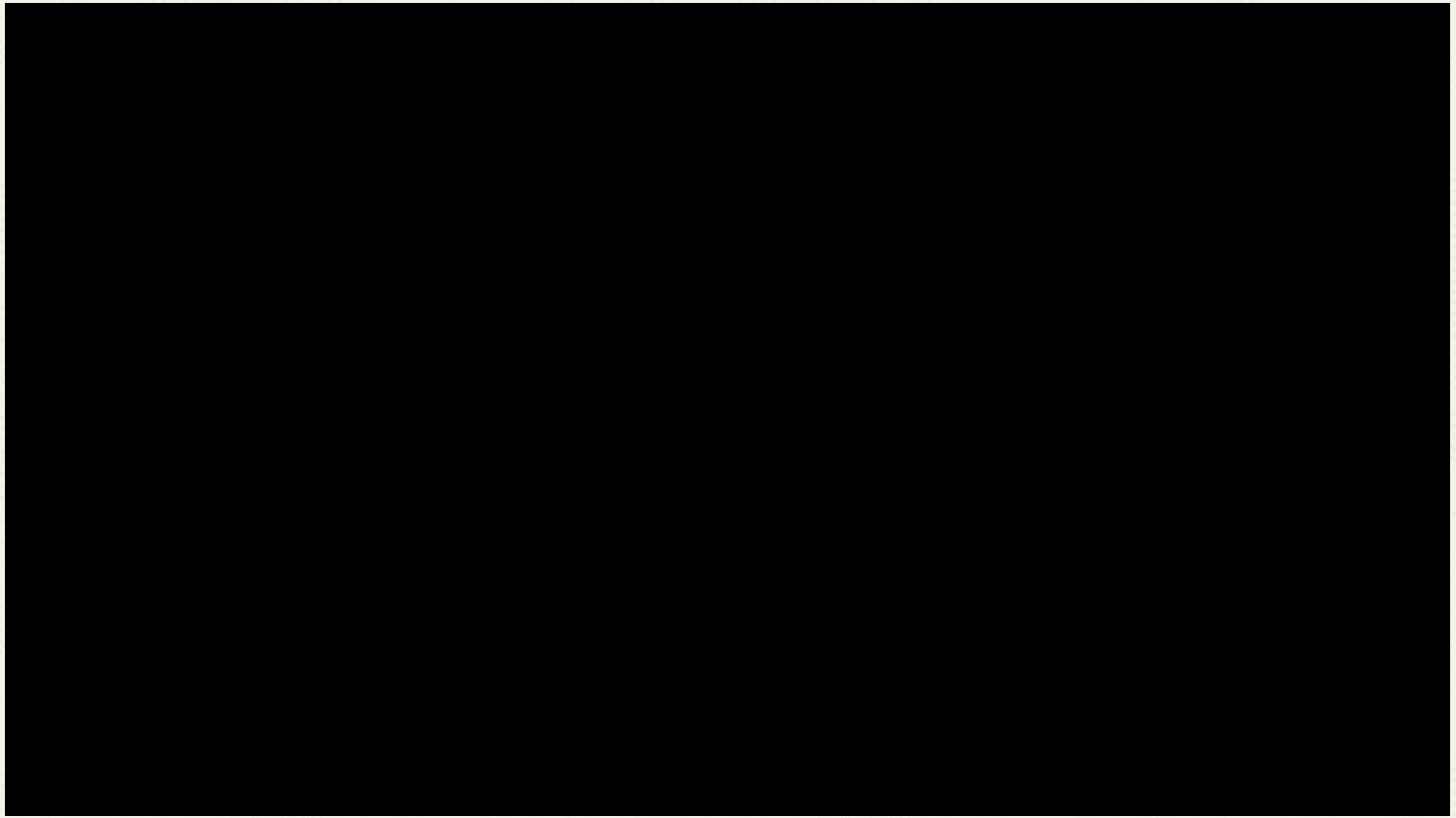
- Runs detections over an input dataset and generated labelled dataset.
- Needs a pretrained neural network.
- Command line available.
- GT view and detection view.

DETECTOR


Neural network
Input images
Supervised labels
Detector
Deploy
Generated labels

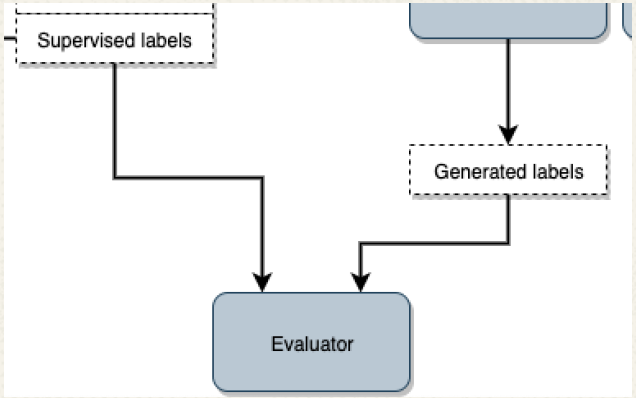
https://www.youtube.com/watch?v=DpK5gqwoSBc

DETECTOR



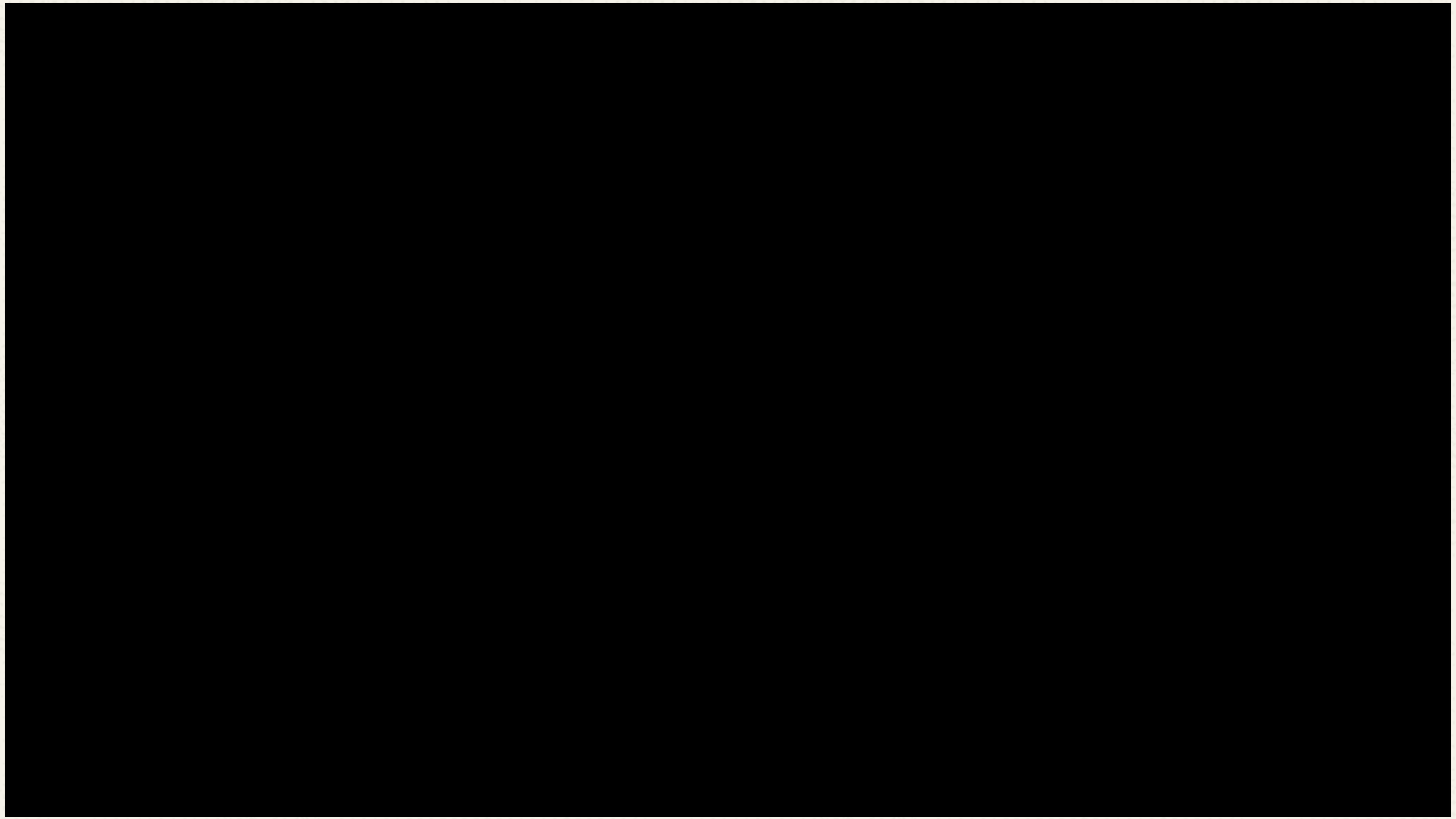
EVALUATOR

- 2 input datasets: ground truth and detections.
- Compares then using metrics.
- mAP, mAR.
- IoU = 0.5:0.95.
- Mean inference time.
- Generates a *.csv* file with the metrics for evaluated datasets.



https://www.youtube.com/watch?v=Vk6Hdv6mRZk

EVALUATOR



EVALUATOR

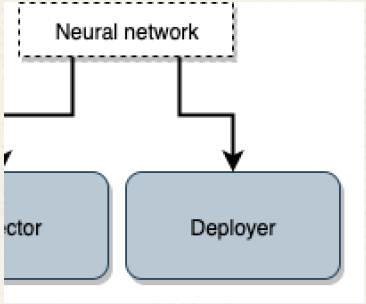
- **Example evaluation_results.csv over COCO minival using COCO pretrained SSD NN.**

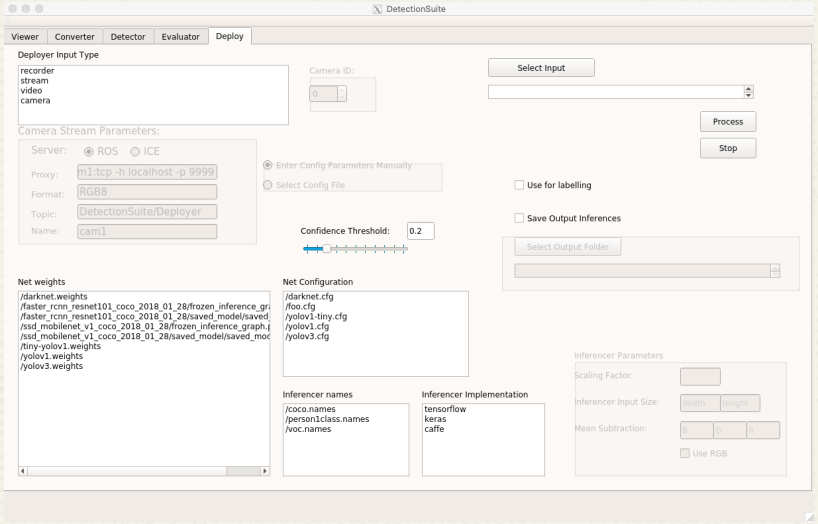
bed		diningtable		toilet	
mAP(IOU=0.5:0.95)	mAR(IOU=0.5:0.95)	mAP(IOU=0.5:0.95)	mAR(IOU=0.5:0.95)	mAP(IOU=0.5:0.95)	mAR(IOU=0.5:0.95)
0.411552	0.551534	0.207528	0.337986	0.631434	0.685475

mAP(Overall)(IOU=0.5:0.95)	mAR(Overall)(IOU=0.5:0.95)	Mean inference time(ms)	Time Taken in Evaluation (second)
0.374243	0.435117	0	14.8566

DEPLOYER

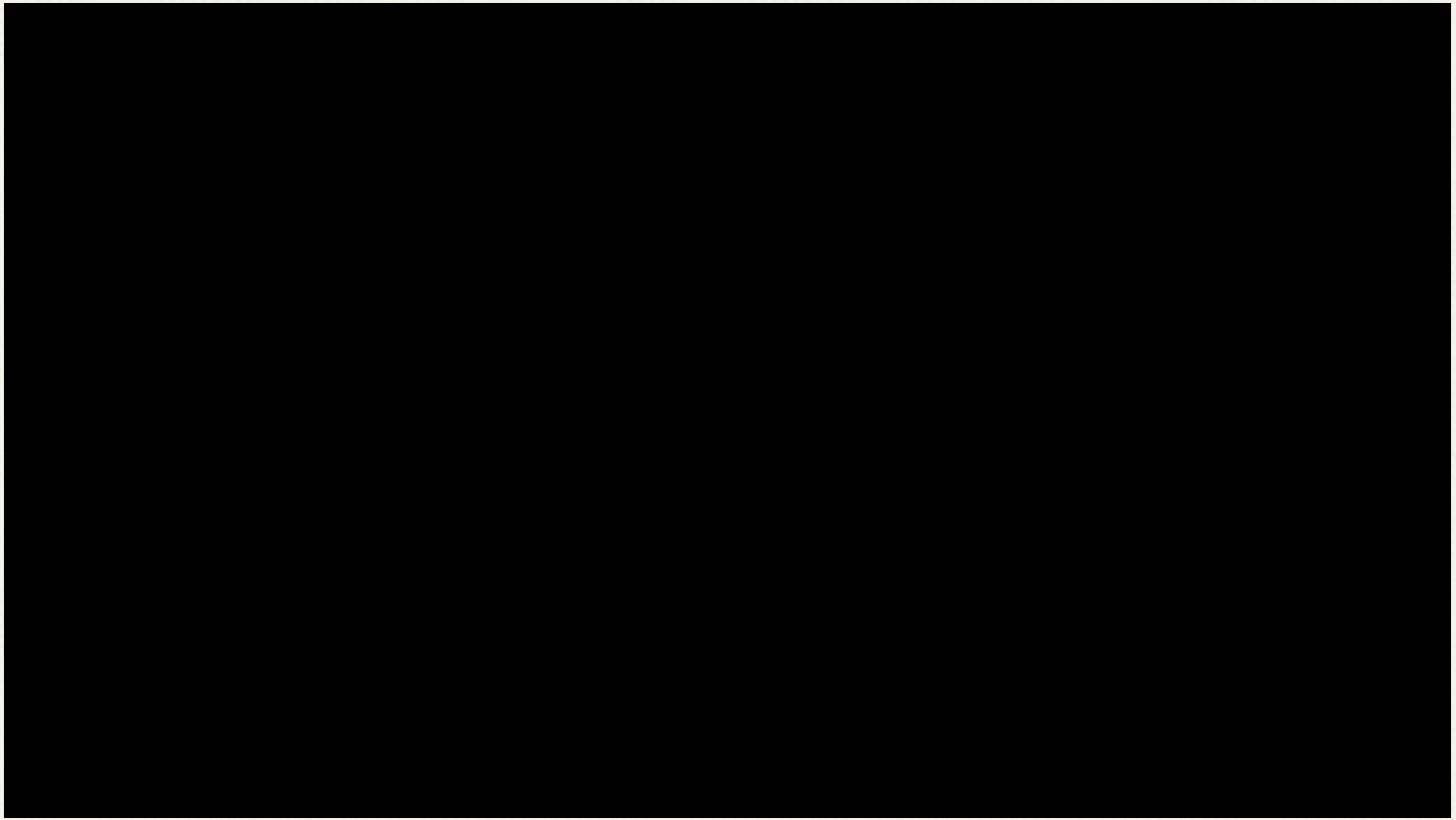
- Accepts different input sources:
  - Video file.
  - Camera stream.
- Neural network model needed.
- Class names.
- Real time processing.
- Only GUI.
- Detection threshold.
- Save inferences → create datasets.
- Labelling.





https://www.youtube.com/watch?v=xX2c_Trp9qY

DEPLOYER

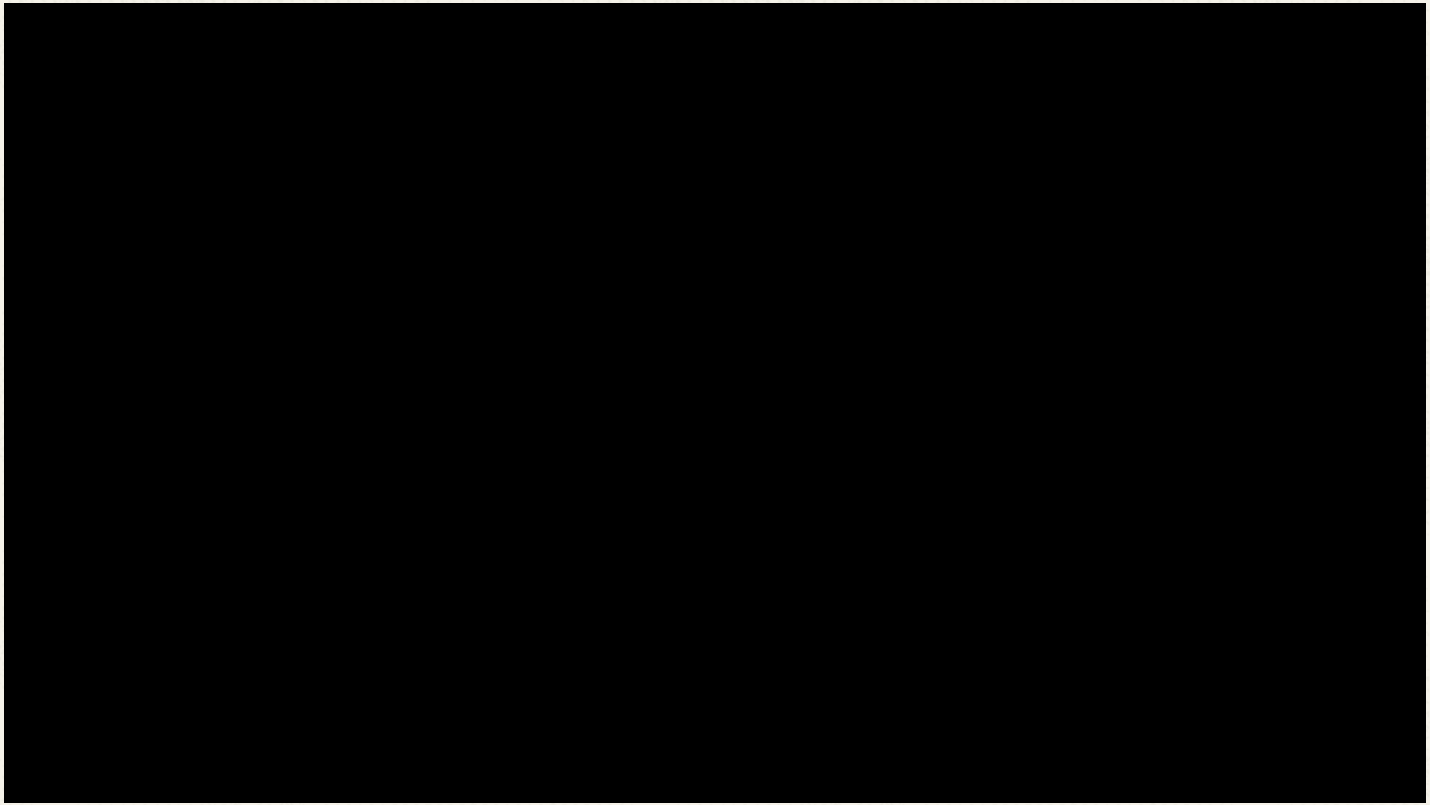


LABELLING

- **Playback functionality.**
- **Adjustable bounding boxes.**
- **Change class and add detections.**

https://www.youtube.com/watch?v=PPNjKML9vUw

LABELLING

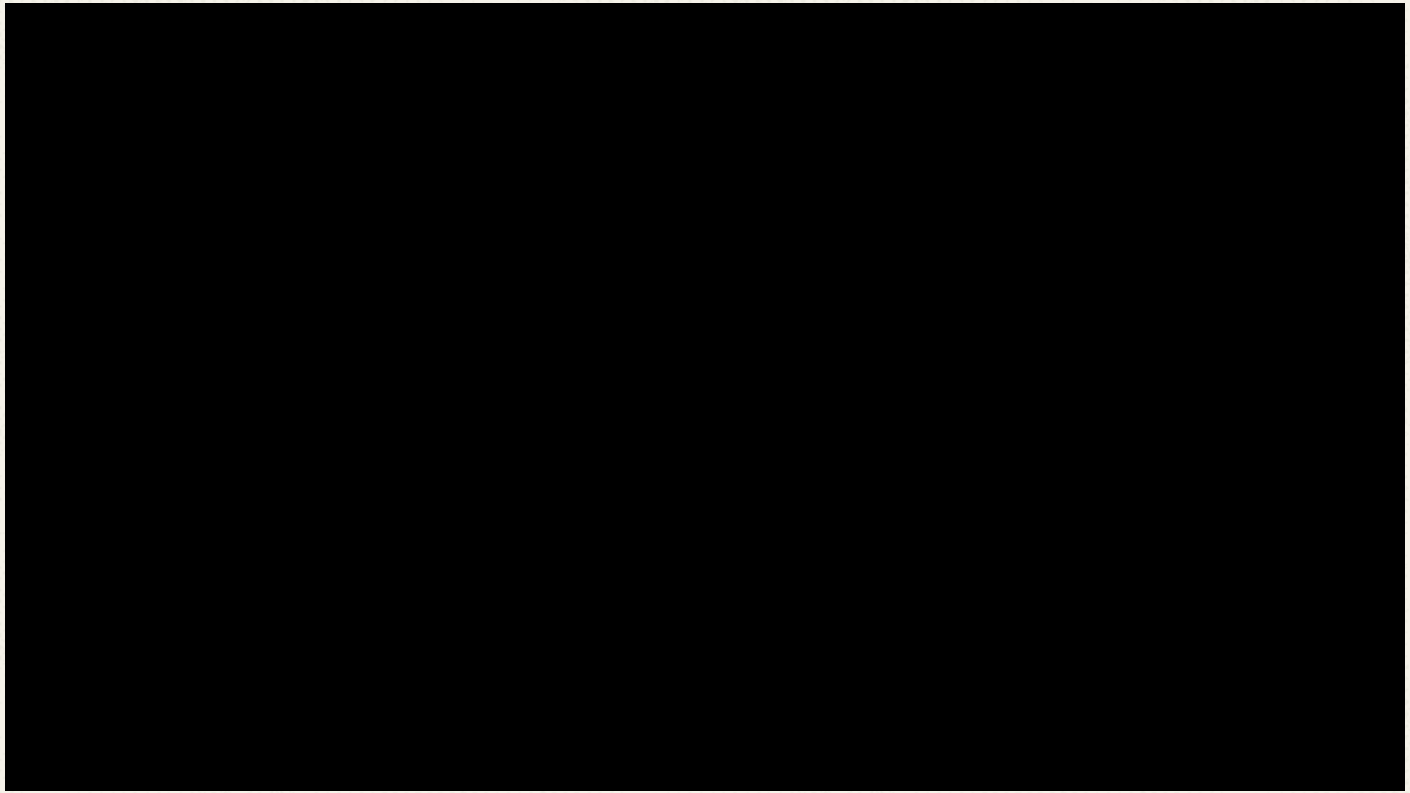


CONVERTER

- **Converts between dataset formats.**
- **CLI available.**
- **Map classes names between datasets.**
- **Split into train/test set.**
- **Own dataset.**

https://www.youtube.com/watch?v=bOjt0v_h640

CONVERTER



CLI

- Many of the different tools are available as command line applications.
- Only a config file is needed.
- AutoEvaluator tool.
- Multiple datasets and frameworks evaluation at the same time.
- Generates a csv with the results.

```
./autoEvaluator -c config.yml
```

CLI

```
Datasets:

-
    inputPath: /opt/datasets/coco/annotations/instances_train2014.json
    readerImplementation: COCO
    readerNames: /opt/datasets/names/coco.names


Inferencers:

-
    inferencerWeights:          /opt/datasets/weights/ssd_mobilenet_v2_coco_2018_03_29/frozen_inference_graph.pb
    inferencerConfig:           /opt/datasets/cfg/foo.cfg               # TensorFlow doesn't need any config file, hence any
    inferencerImplementation:   tensorflow                              # empty foo.cfg file
    inferencerNames:            /opt/datasets/names/coco.names

-
    inferencerWeights:          /opt/datasets/weights/VGG_VOC0712_SSD_512x512_iter_120000.h5
    inferencerConfig:           /opt/datasets/cfg/foo.cfg               # New version Keras also doesn't need any file, all the
    inferencerImplementation:   keras                                   # data is stored in the HDF5 file including model
    inferencerNames:            /opt/datasets/names/voc.names           # weights, configuration and optimizer state, hence we
                                                                       # are using an empty foo.cfg file
-
    inferencerWeights:          /opt/datasets/weights/VGG_VOC0712_SSD_512x512_iter_240000.h5
    inferencerConfig:           /opt/datasets/cfg/foo.cfg
    inferencerImplementation:   keras
    inferencerNames:            /opt/datasets/names/voc.names



outputCSVPath: /opt/datasets/output
```

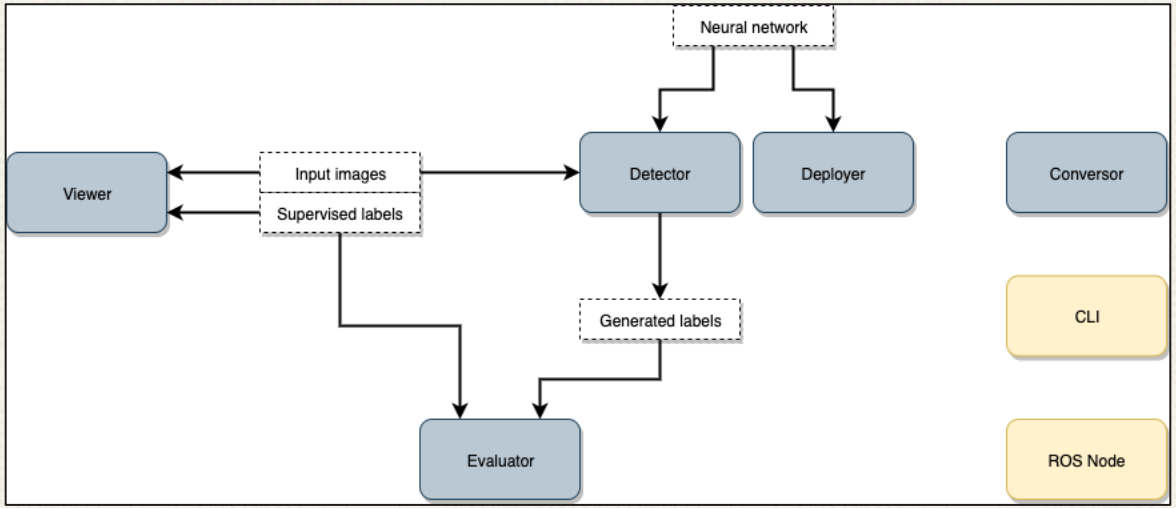
ROS NODE

- **Real time inference over an image stream.**
- **Subscribe to a ROS topic.**
- **Configuration file.**

https://www.youtube.com/watch?v=xvPaFZD5EoY

ROS NODE

DETECTION SUITE WORKFLOW



minival

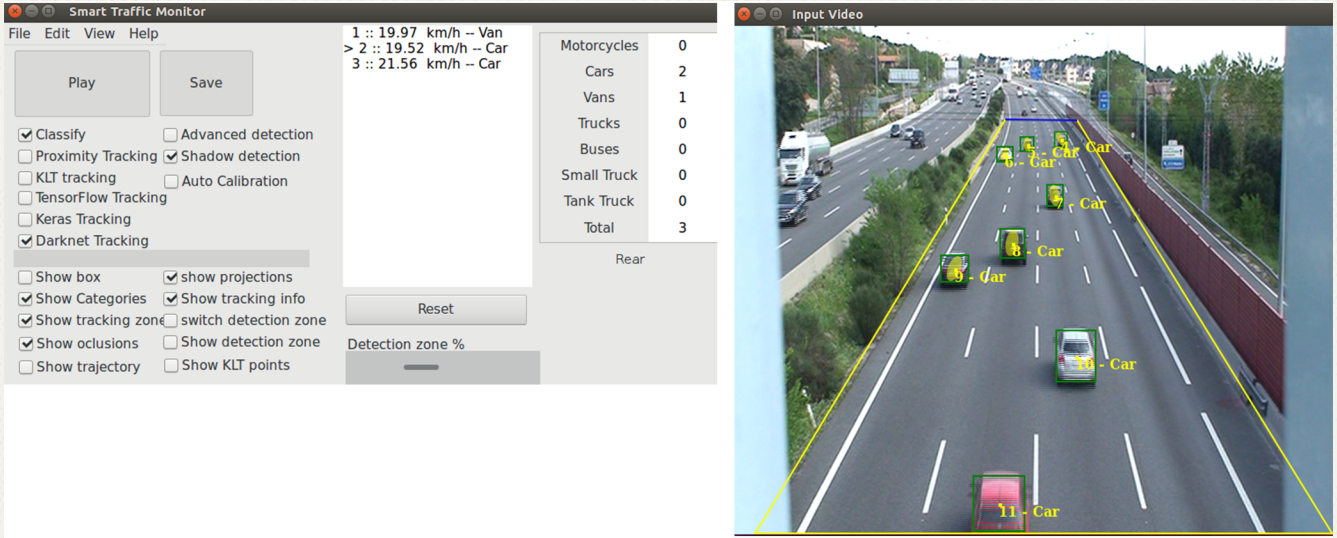
evaluation_results.csv

INDEX

- INTRODUCTION
- STATE OF THE ART
- DETECTIONSUITE TOOLBOX
- EXPERIMENTS
- CONCLUSIONS

SMART TRAFFIC SENSOR

- **Monitors road traffic using computer vision.**



SMART TRAFFIC SENSOR

- **Evolution of Traffic Monitor project (classical CV techniques).**
- **New version with state of the art deep learning techniques.**
- **Several neural networks trained (different frameworks).**
- **Compared using Detection Suite.**
- **Detection and classification of road traffic.**
- **Own created dataset.**
- **Pretrained network vs. pretrained + trained using the dataset.**
- **The best performing network is YOLOv3.**
- **Better performance and higher mean inference time (more complex and deeper network).**

J.F. Martínez. Monitorización Visual de Tráfico Rodado usando Deep Learning. Master's thesis. Universidad Rey Juan Carlos, Madrid, Spain. 2019.

SMART TRAFFIC SENSOR

- **Evaluation over created STS dataset**

Network	mAP (Overall)(IoU=0.5:0.95)	mAR (Overall)(IoU=0.5:0.95)	Mean inference time (ms)
Keras SSD VVG-16	0	0	0
Yolo v3	0	0	14162
TF SSD MobileNet v2	0.0035	0.0373	142
Keras SSD VVG-16 STS trained	0.6709	0.7082	3194
Yolo v3 STS trained	0.8641	0.9385	16894
TF SSD MobileNet v2 STS trained	0.3283	0.4231	76

SMART TRAFFIC SENSOR

- **Comparison of systems in high-quality videos using Detection Suite.**

System type	mAP	mAR
Smart Traffic Sensor	0.8926	0.9009
Traffic Monitor	0.4374	0.5940
Best performing NN (YOLO v3)	0.8316	0.8966

INDEX

- INTRODUCTION
- STATE OF THE ART
- DETECTIONSUITE TOOLBOX
- EXPERIMENTS
- CONCLUSIONS

CONCLUSIONS

- **Set of tools. Modulable.**
- **CLI and GUI.**
- **Real life application.**
- **Extensible**
- **In development (pytorch support, YOLOv3...).**
- **Accepting new contributors to projects and org!**

QUESTIONS?

@sergiopaniego
sergiopaniegoblanco@gmail.com

https://github.com/JdeRobot/DetectionSuite
https://jderobot.github.io/DetectionSuite/

Presentation template: slidescarnival.com