7th UCAAT

User Conference on Advanced Automated Testing

ETSI

Bordeaux, 22-24 October 2019

USING TENSORFLOW AND COMPUTER VISION TO TEST GENERIC WEB SERVICE AVAILABILITY

Presented by Enrico La Vela

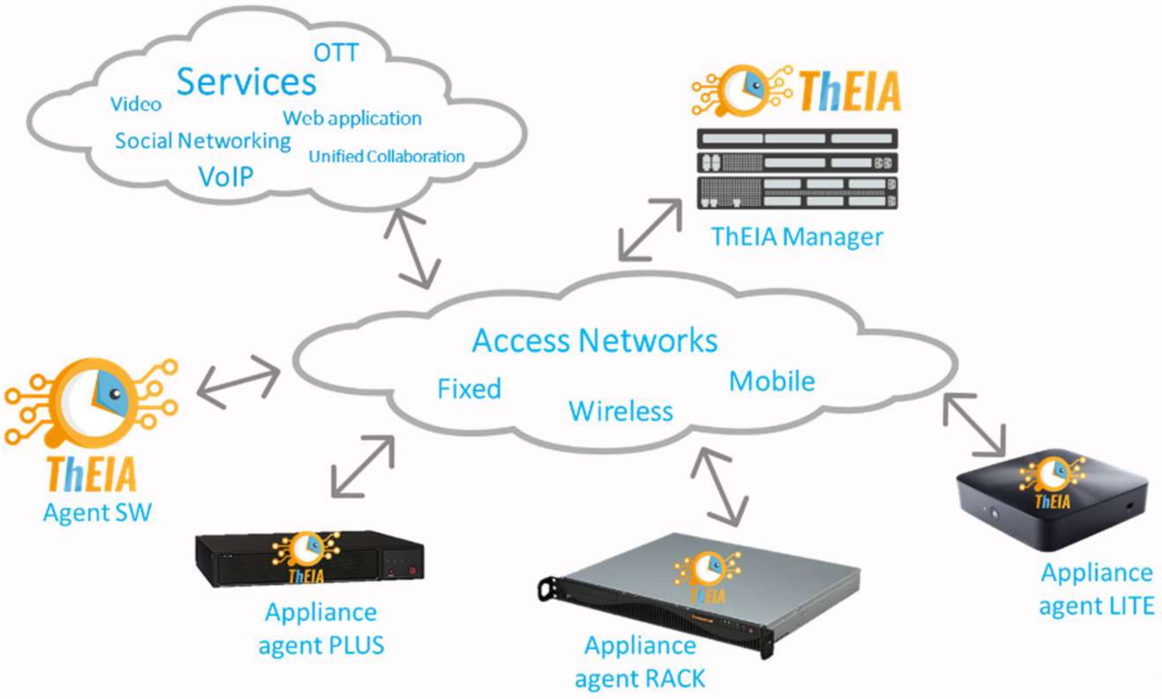
NetResults
Building the digital society

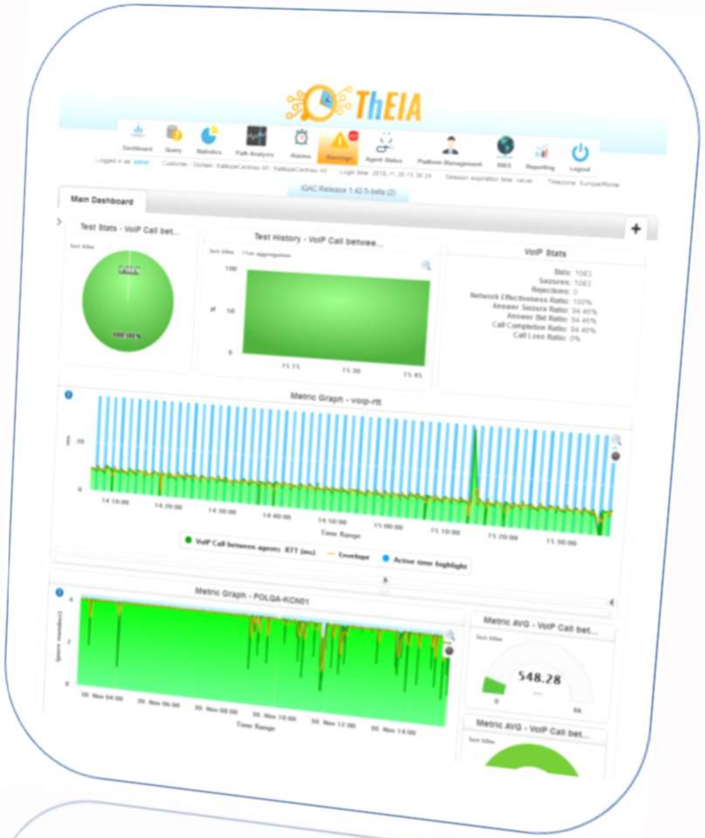
Agenda

- Brief intro to our test platform: ThEIA
- Use case: Computer Vision for testing web services
- Sikuli – brief explaination and problem faced
- Solution: Tensorflow – brief explaination
- Our Custom Object Detector applied to *Login forms*
- Results and final considerations
- Next steps

ThEIA Platform

Testing Environment for Internet Application





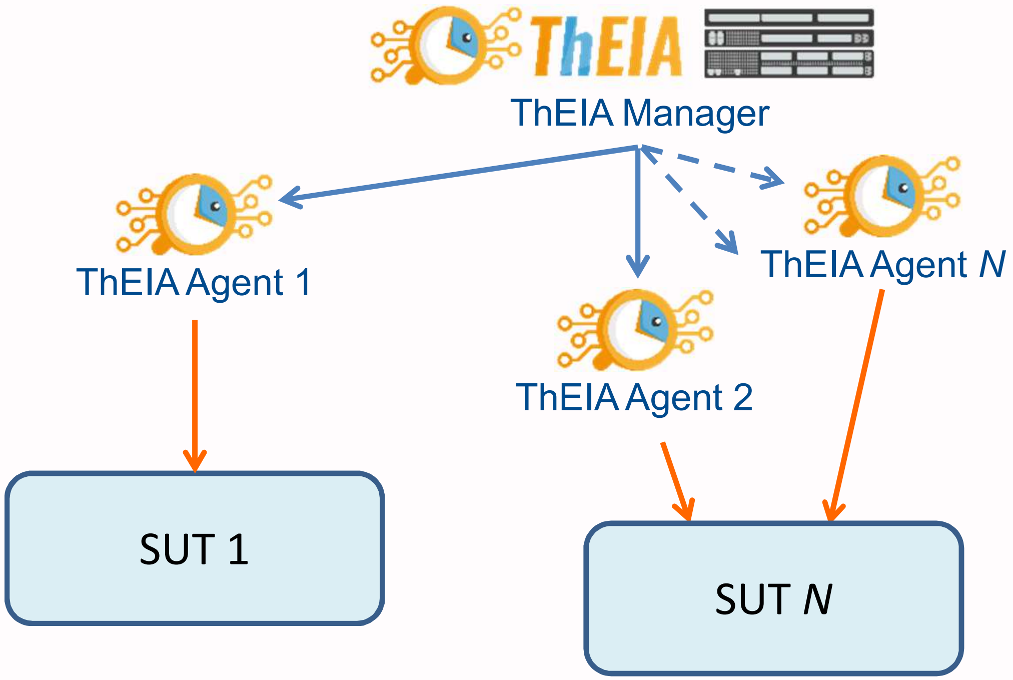
VoIP Stats

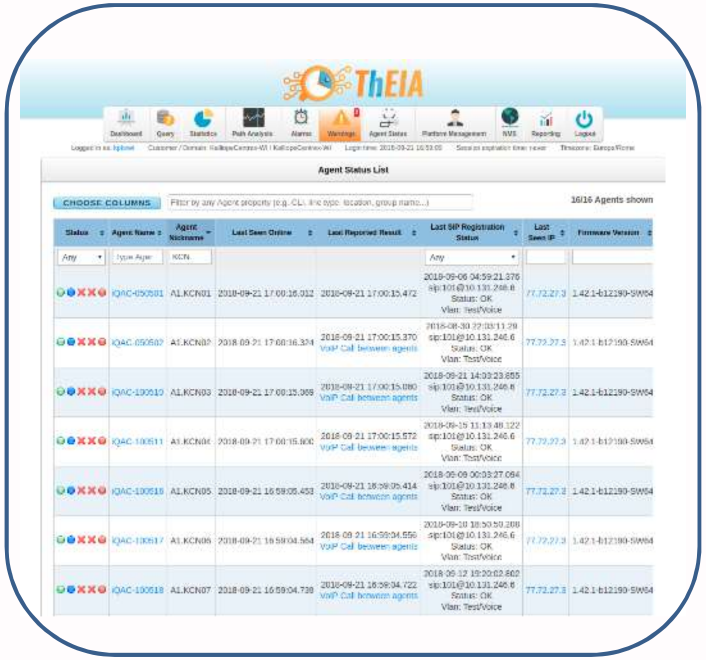
Bids: 1063
Seizures: 1063
Rejections: 0
Network Effectiveness Ratio: 100%
Answer Seizure Ratio: 94.40%
Answer Bid Ratio: 94.40%
Call Completion Ratio: 94.40%
Call Loss Ratio: 0%

ThEIA Platform

Testing Environment for Internet Application

- Example of architecture:
  - Independent or coordinated tests managed with *Testplans*





Logged in as: kpinel Customer / Domain: KalliopeCentrex-WI / KalliopeCentrex-WI Login time: 2018-09-21 16:53:05 Session expiration time: never Timezone: Europe/Rome

Agent Status List

Filter by any Agent property (e.g. CLI, line type, location, group name...)

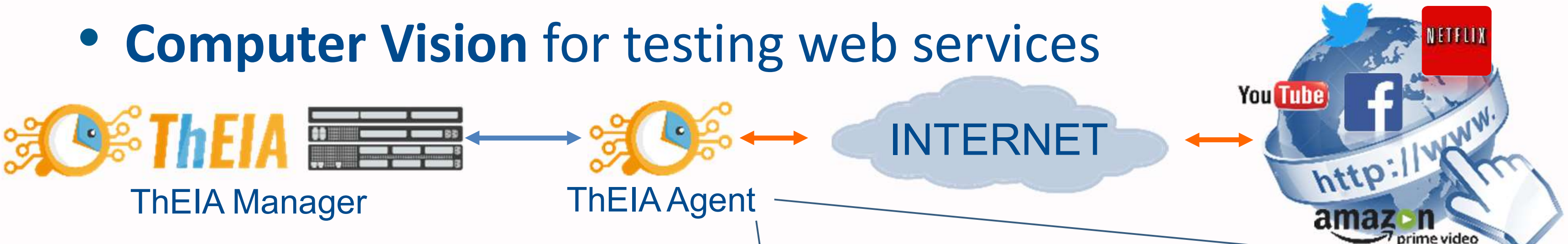
16/16 Agents shown

Status	Agent Name	Agent Nickname	Last Seen Online	Last Reported Result	Last SIP Registration Status	Last Seen IP	Firmware Version
Any	type Agen	KCN			Any		
●●✕✕●	IQAC-050501	A1.KCN01	2018-09-21 17:00:16.012	2018-09-21 17:00:15.472	2018-09-06 04:59:21.376 sip:101@10.131.246.6 Status: OK Vlan: Test/Voice	77.72.27.3	1.42.1-b12190-SW64
●●✕✕●	IQAC-050502	A1.KCN02	2018-09-21 17:00:16.324	2018-09-21 17:00:15.370 VoIP Call between agents	2018-08-30 22:03:11.29 sip:101@10.131.246.6 Status: OK Vlan: Test/Voice	77.72.27.3	1.42.1-b12190-SW64
●●✕✕●	IQAC-100510	A1.KCN03	2018-09-21 17:00:15.368	2018-09-21 17:00:15.080 VoIP Call between agents	2018-09-21 14:03:23.855 sip:101@10.131.246.6 Status: OK Vlan: Test/Voice	77.72.27.3	1.42.1-b12190-SW64
●●✕✕●	IQAC-100511	A1.KCN04	2018-09-21 17:00:15.800	2018-09-21 17:00:15.572 VoIP Call between agents	2018-09-15 11:13:48.122 sip:101@10.131.246.6 Status: OK Vlan: Test/Voice	77.72.27.3	1.42.1-b12190-SW64
●●✕✕●	IQAC-100516	A1.KCN05	2018-09-21 16:59:05.453	2018-09-21 16:59:05.414 VoIP Call between agents	2018-09-09 00:03:27.064 sip:101@10.131.246.6 Status: OK Vlan: Test/Voice	77.72.27.3	1.42.1-b12190-SW64
●●✕✕●	IQAC-100517	A1.KCN06	2018-09-21 16:59:04.564	2018-09-21 16:59:04.556 VoIP Call between agents	2018-09-10 18:50:50.208 sip:101@10.131.246.6 Status: OK Vlan: Test/Voice	77.72.27.3	1.42.1-b12190-SW64
●●✕✕●	IQAC-100518	A1.KCN07	2018-09-21 16:59:04.738	2018-09-21 16:59:04.722 VoIP Call between agents	2018-09-12 19:20:02.802 sip:101@10.131.246.6 Status: OK Vlan: Test/Voice	77.72.27.3	1.42.1-b12190-SW64

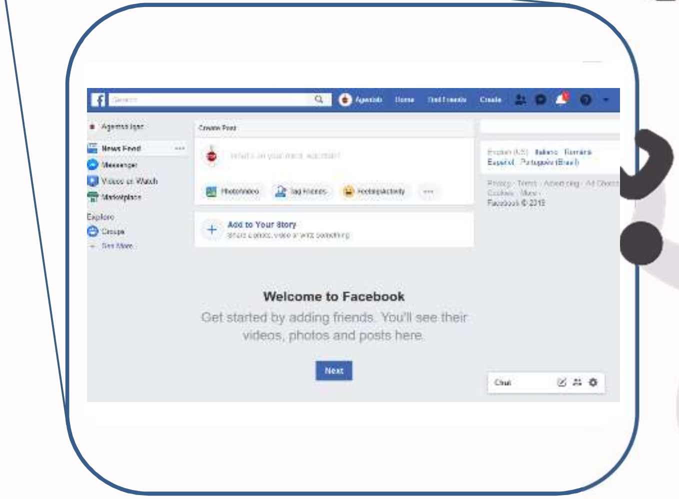
ThEIA Platform

Testing Environment for Internet Application

- One of ThEIA use cases:
  - **Computer Vision** for testing web services



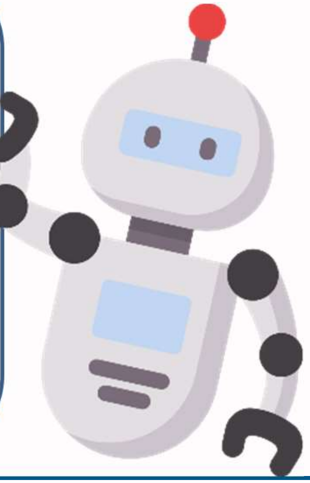
  - Agent acts as an user that access to web services and performs several actions
  - Agent measures service availability, web pages responsiveness time, etc



Welcome to Facebook

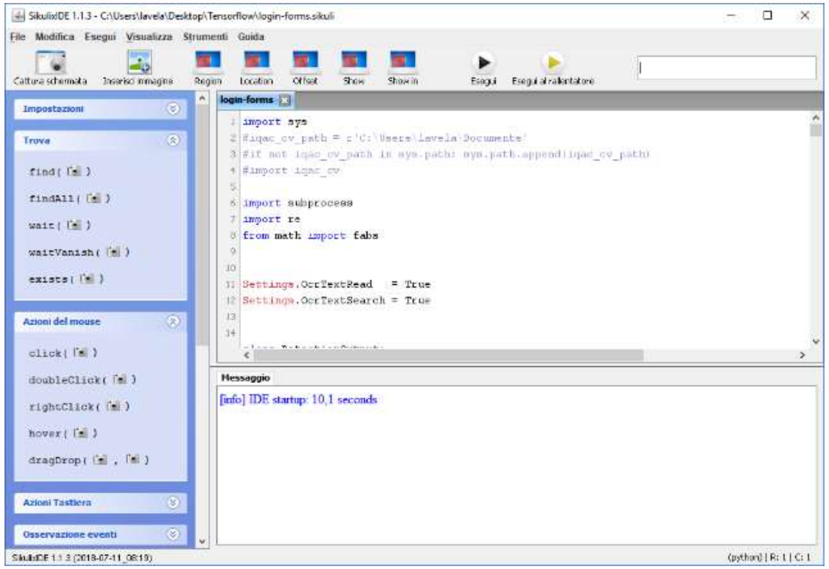
Get started by adding friends. You'll see their videos, photos and posts here.

Next



How do we implement Computer Vision?

sikuli



```
find( )
findAll( )
wait( )
waitVanish( )
exists( )
```

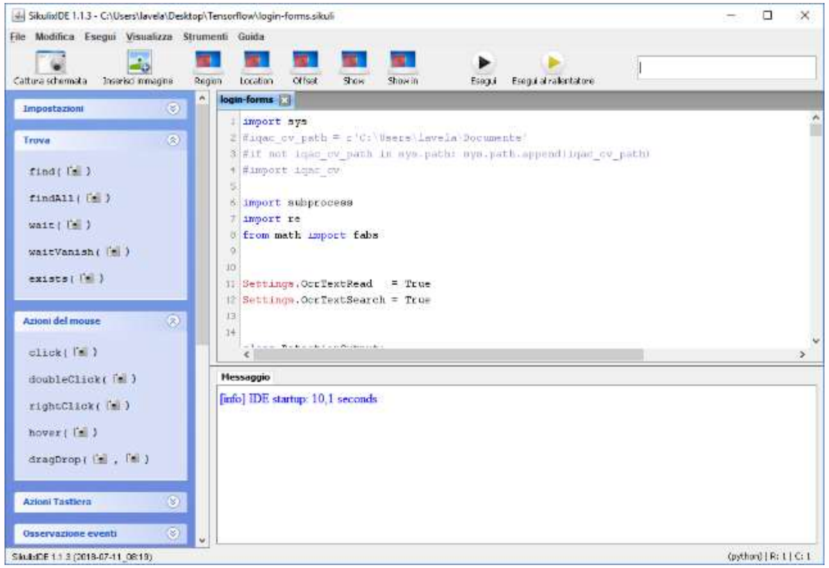
```
click( )
doubleClick( )
rightClick( )
hover( )
dragDrop( , )
```

```
import sys
#iqac_cv_path = r'C:\Users\lavela\Documente'
#if not iqac_cv_path in sys.path: sys.path.append(iqac_cv_path)
#import iqac_cv

import subprocess
import re
from math import fabs


Settings.OcrTextRead   = True
Settings.OcrTextSearch = True
```

```
[info] IDE startup: 10,1 seconds
```

What’s Sikuli?

- **SikuliX** automates anything you see on the screen of your desktop computer.
- Sikuli uses image recognition powered by OpenCV to identify and control GUI components.



```
import sys
#iqac_cv_path = r'C:\Users\lavela\Documents'
#if not iqac_cv_path in sys.path: sys.path.append(iqac_cv_path)
#import iqac_cv

import subprocess
import re
from math import fabs


Settings.OcrTextRead   = True
Settings.OcrTextSearch = True
```

```
[info] IDE startup: 10,1 seconds
```

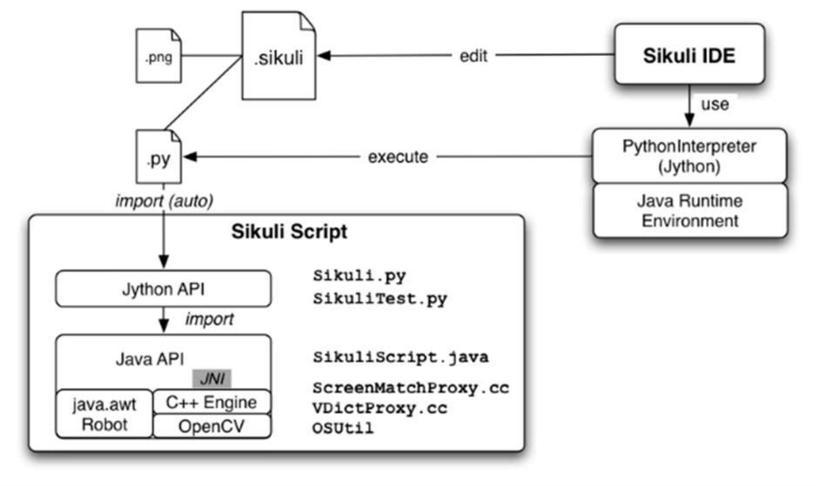
SikulixIDE 1.1.3 (2018-07-11_08:19) (python) | R: 1 | C: 1

Why Sikuli?

- Sikuli is powerful in cases when there is no easy access to a GUI's internals or the source code of the application or web page you want to act on or when you want effectively test what the user sees on the screen -> **Quality of Experience (QoE)**
- It uses pixel detection in order to automate things
- We use Sikuli for testing QoE of web services and their availability through our ThEIA platform

How Sikuli works?

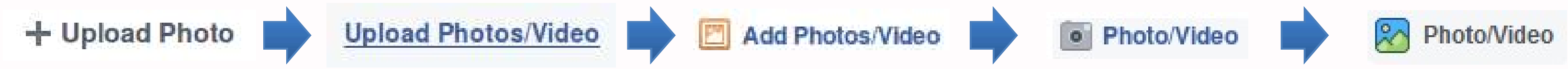
- It works under Java (> 8) environment and makes use of python scripts (2.7) python powered



Limitations

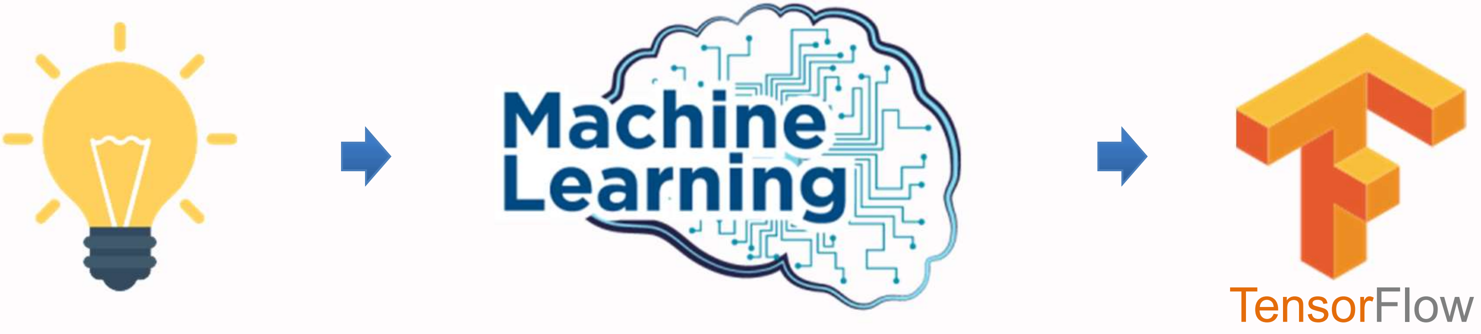
- Sikuli is based on pixel detection
- When webpages’ GUI changes, Sikuli script fails -> test fails -> false negative results
  - Continuous changes in webpage layout, icons, button style

Example:



- You need to perform changes on script code and image capture

How to solve it?



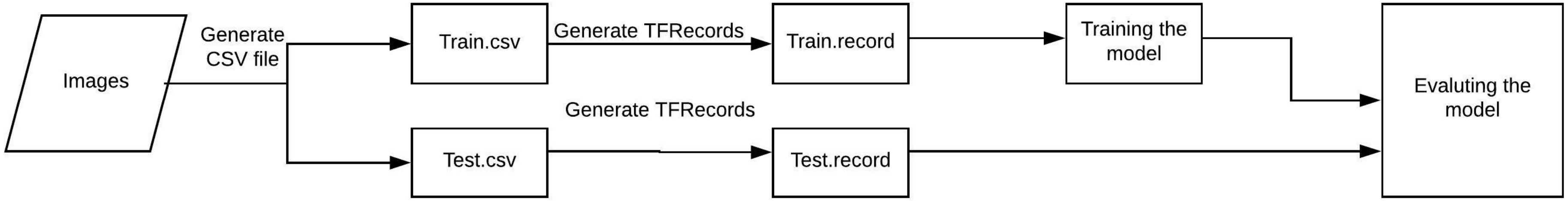
What is TensorFlow?

- TensorFlow is an open source computational framework designed by the Google team, used to build **Machine Learning** models
- It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called ***tensors***
- TensorFlow provides stable **Python** and **C++** APIs
- There are a certain number of different models

How do we used TensorFlow?

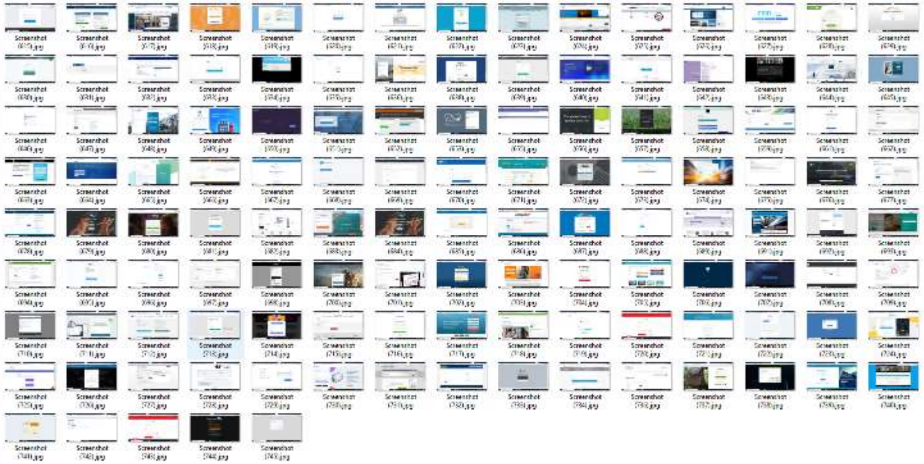
- We used the TensorFlow (Custom) **Object Detection API** (based on Python 3.6), invoked by our Sikuli scripts (based on Python 2.7)
- We have customized the model to recognize login forms, by **Training and Evaluating a Custom Object Detector**

Basic flow diagram



Our custom *Login forms* Object Detector STEP 1

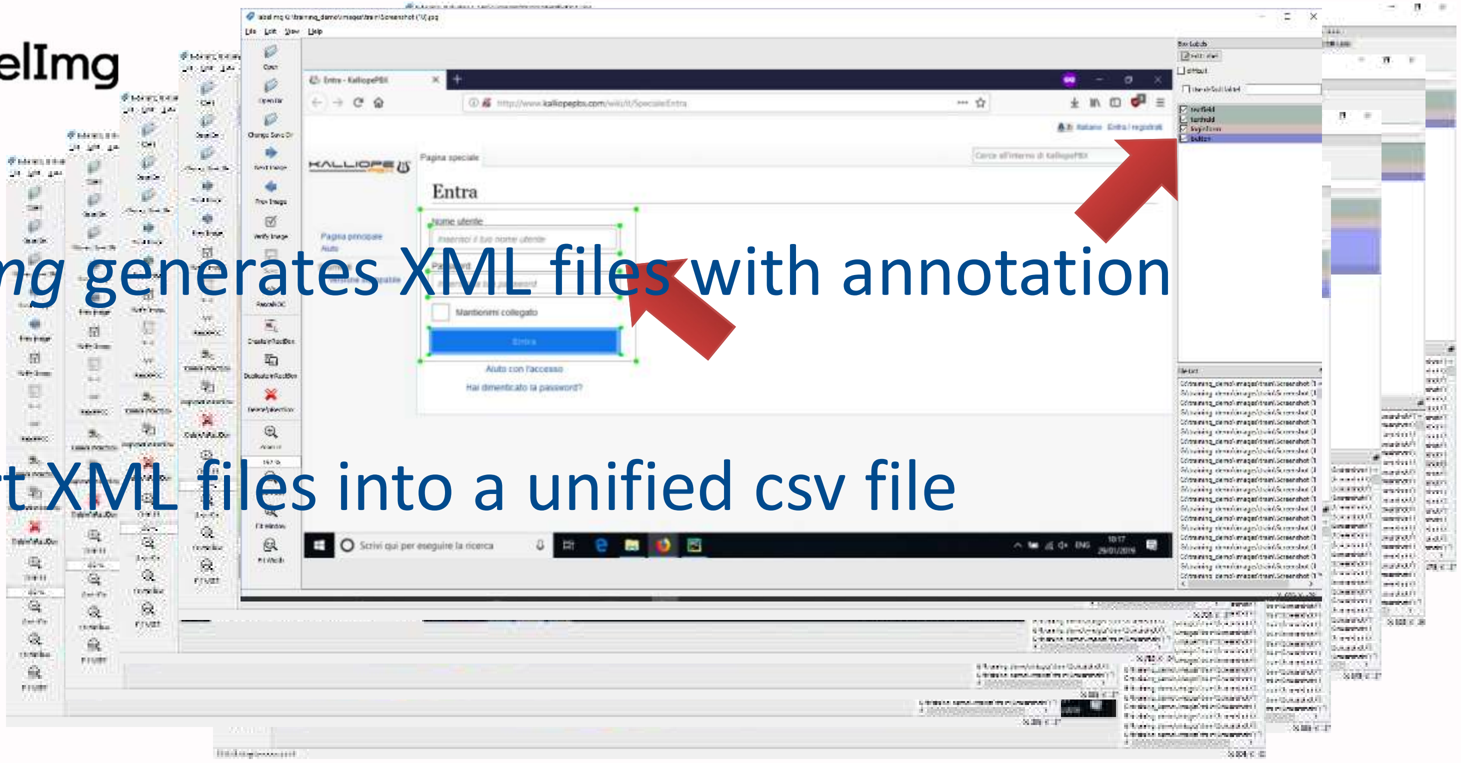
- Create a dataset:
  - Go into several web pages with login forms and take screenshots (we used 700+ references for training phase)
- Random web page research (different domains)
- Language: English, Italian



Our custom *Login forms* Object Detector STEP 2

- Label each image using a graphic image annotation tool

LabelImg



- *LabelImg* generates XML files with annotation

- Convert XML files into a unified csv file

Our custom *Login forms* Object Detector STEP 3

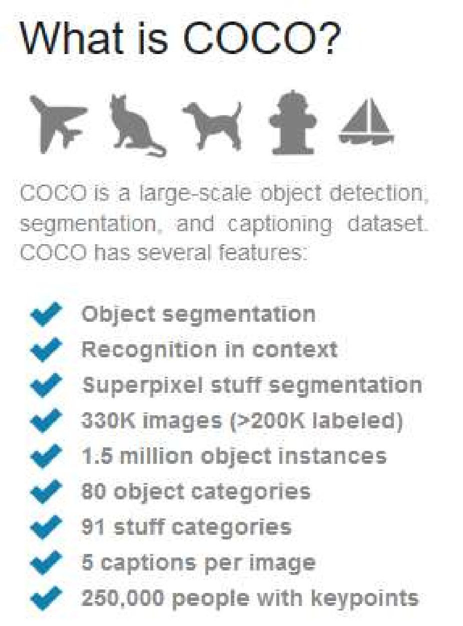
- Generate **TFRecords**

- TensorFlow object detection API doesn’t take csv files as an input, but it needs *tf record files* to train the model

- Use of a python scripts to generate *record file* from *CSV file*

Our custom *Login forms* Object Detector STEP 4 - [1/2]

- Training the Model
- Use of a pre-trained model as starting point:
  - **faster_rcnn_inception_v2_coco**
  - i.e. Faster R-CNN with Inception v2 algorithm for MSCOCO Dataset
- **What are these acronyms?**
  - They are related to algoritm and dataset used
  - R-CNN: **R**egion-**C**onvolutional **N**eural **N**etwork



What is COCO?

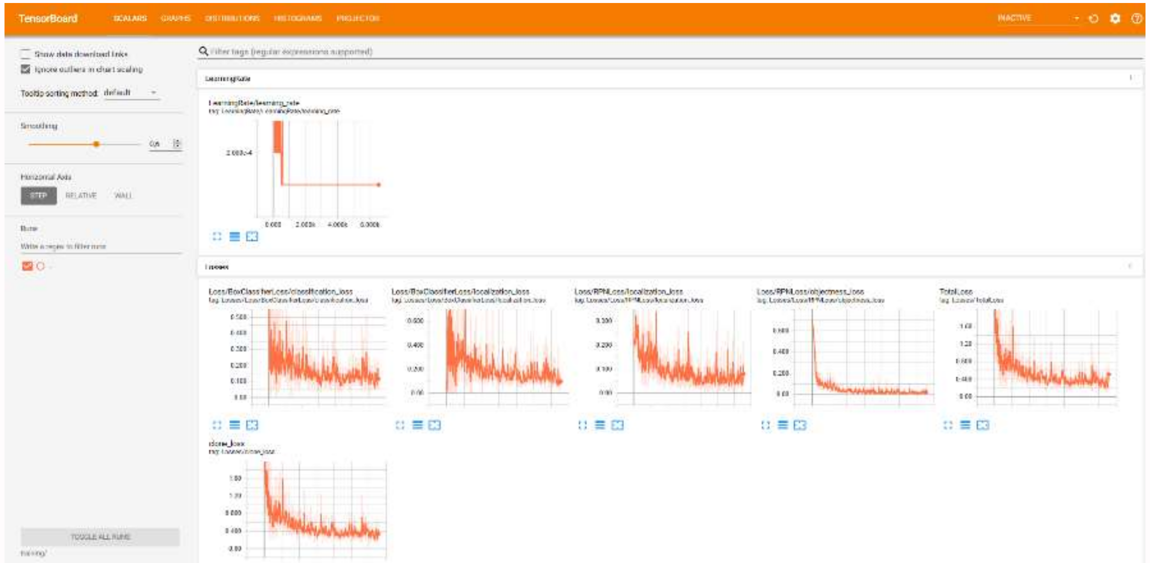
COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

Our custom *Login forms* Object Detector STEP 4 - [2/2]

- Training the Model
- With a normal laptop: spent 1 week in computation
- ~41K training steps

PLEASE BE PATIENT



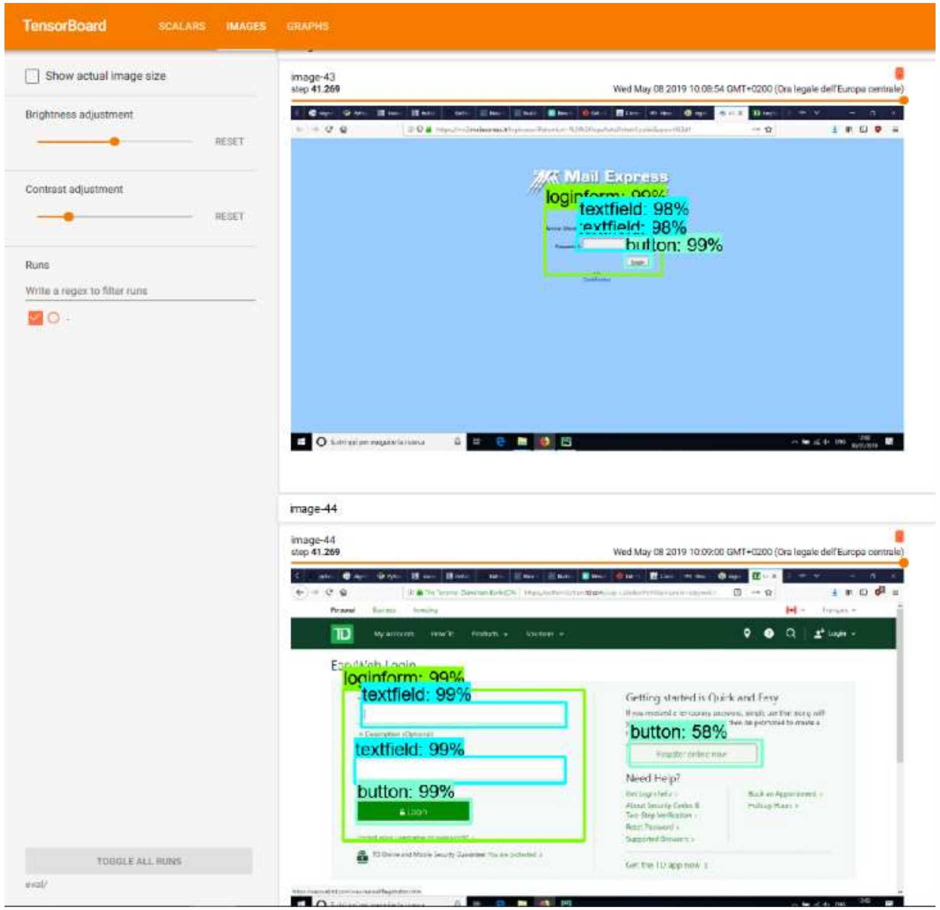
TensorBoard SCALARS GRAPHS DISTRIBUTIONS HISTOGRAMS PROJECTOR INACTIVE

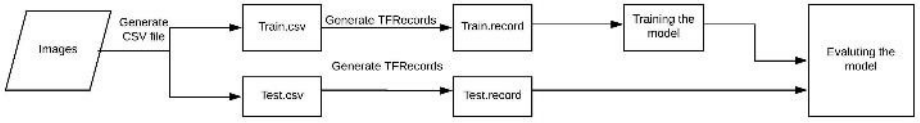
□ Show data download links
☑ Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing 0.6
Horizontal Axis: STEP RELATIVE WALL
Runs
Write a regex to filter runs
TOGGLE ALL RUNS
training/

Filter tags (regular expressions supported)
LearningRate
LearningRate/learning_rate
tag: LearningRate/LearningRate/learning_rate
2.000e-4
0.000
2.000k
4.000k
6.000k
Losses
Loss/BoxClassifierLoss/classification_loss
tag: Losses/Loss/BoxClassifierLoss/classification_loss
0.500
0.400
0.300
0.200
0.100
0.00
Loss/BoxClassifierLoss/localization_loss
tag: Losses/Loss/BoxClassifierLoss/localization_loss
0.600
0.400
0.200
0.00
Loss/RPNLoss/localization_loss
tag: Losses/Loss/RPNLoss/localization_loss
0.300
0.200
0.100
0.00
Loss/RPNLoss/objectness_loss
tag: Losses/Loss/RPNLoss/objectness_loss
0.600
0.400
0.200
0.00
TotalLoss
tag: Losses/TotalLoss
1.60
1.20
0.800
0.400
0.00
clone_loss
tag: Losses/clone_loss
1.60
1.20
0.800
0.400
0.00

Our custom *Login forms* Object Detector STEP 5

- Evaluate the Model
- Use of a different data set (not the 700+ screenshots, but others…~70)
- Some iteration in training phase (STEP 4)





Our custom *Login forms* Object Detector STEP 6

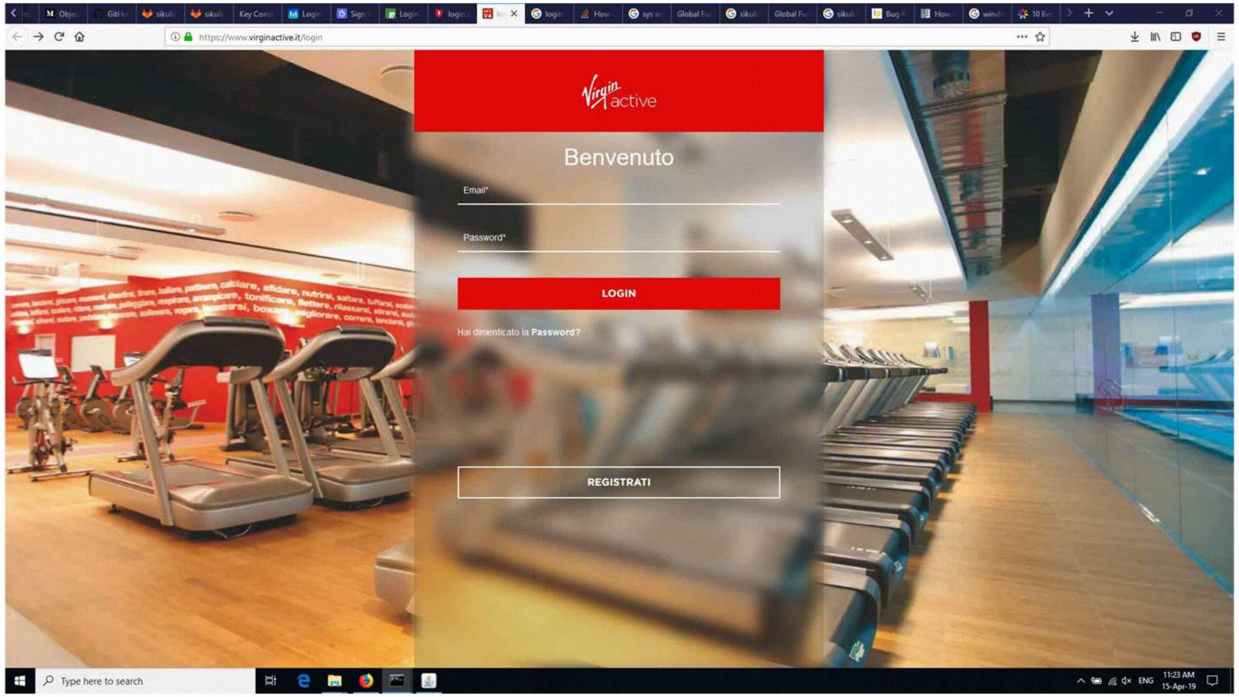
- The model is well trained and results are acceptable
  - **Training phase is complete**

MISSION ACCOMPLISHED

- It is possible to use the model database (frozen inference graph *.pb* file) to perform the *login forms* object detection with your py scripts
  - In our PoC case the .pb file is ~60MB

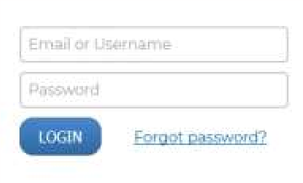
Results [1/2]

- Our scripts are able to perform login access to almost any web page (below a real example)

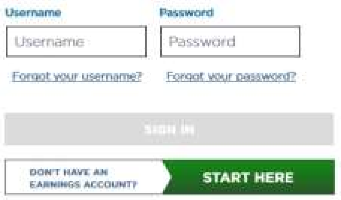


Results [2/2]

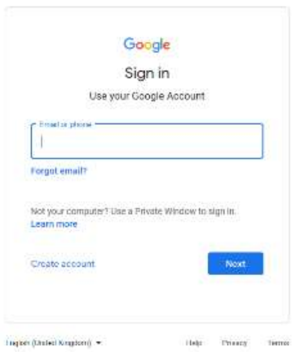
- We have optimized our Sikuli scripts, making use of some logic (application layer) that helps to detect typical position of text fields, two steps form and *Login button* (use of OCR)


Email or Username
Password
LOGIN
Forgot password?

vs


Username
Password
Username
Password
Forgot your username?
Forgot your password?
SIGN IN
DON'T HAVE AN EARNINGS ACCOUNT?
START HERE

vs


Google
Sign in
Use your Google Account
Email or phone
Forgot email?
Not your computer? Use a Private Window to sign in.
Learn more
Create account
Next
English (United Kingdom)
Help
Privacy
Terms

Final considerations

- Lightweight py scripts executed at client side that avoid script rework when web pages changes
- Limitations: it doesn't work with captcha

- It is a Proof of Concept (PoC), still limited to perform generic login action

Next steps

- Apply the same concept (Sikuli + TensorFlow API) to other kind of web page common actions (post a comment, upload a photo, …)
- Apply TensorFlow API to evaluate QoE of video content (i.e. MOS evaluation)

Q&A

- Authors Contacts
  - Enrico La Vela e.lavela@netresults.it [SPEAKER]
  - Silvia Vistoli s.vistoli@netresults.it
  - Sergio Borghese s.borghese@netresults.it
  - Francesco Oppedisano f.oppedisano@netresults.it