



Andalusian
Research Institute in
**Data Science and
Computational Intelligence**

Practical Deep Learning: A quick glance

Image classification and object detection

Francisco Pérez Hernández
fperezhernandez@ugr.es
University of Granada

Outline

- Image classification
 - A brief summary
 - Galaxy classification
- Object detection
 - Introduction
 - State-of-the-art meta-architectures
 - Practical example of detection
- Practical: Classification of galaxies

Outline

- Image classification
 - A brief summary
 - Galaxy classification



- Object detection
 - Introduction
 - State-of-the-art meta-architectures
 - Practical example of detection
- Practical: Classification of galaxies

Image classification

Convolution layers are the eyes of a CNN → Extract features



Image classification

How CNNs extract features?

Each kernel extract some features → Convolution

More depth → More abstraction

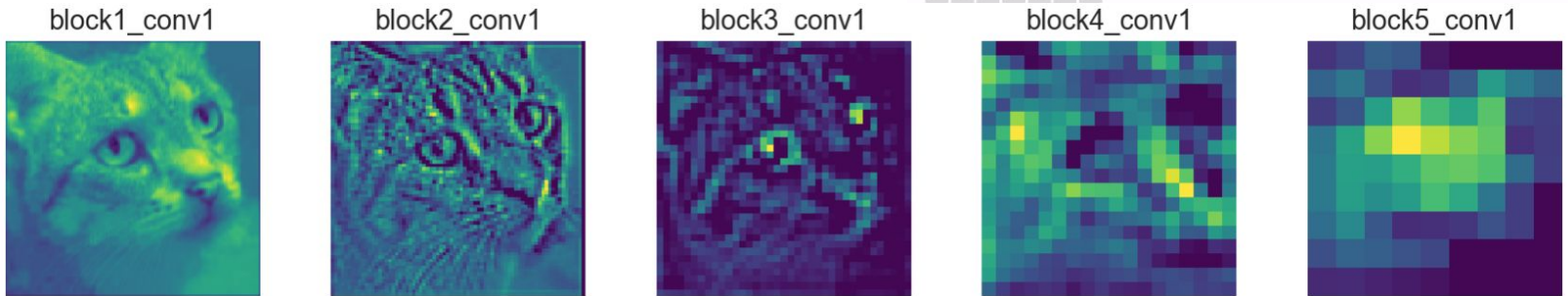
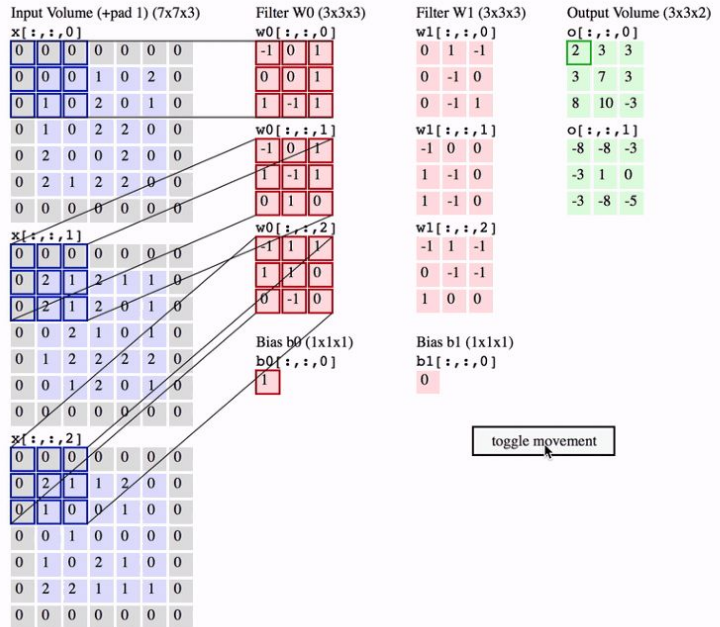


Image classification

Architecture of CNNs:

- Feature extraction
- Dimension reduction

State-of-the-art [models](#)

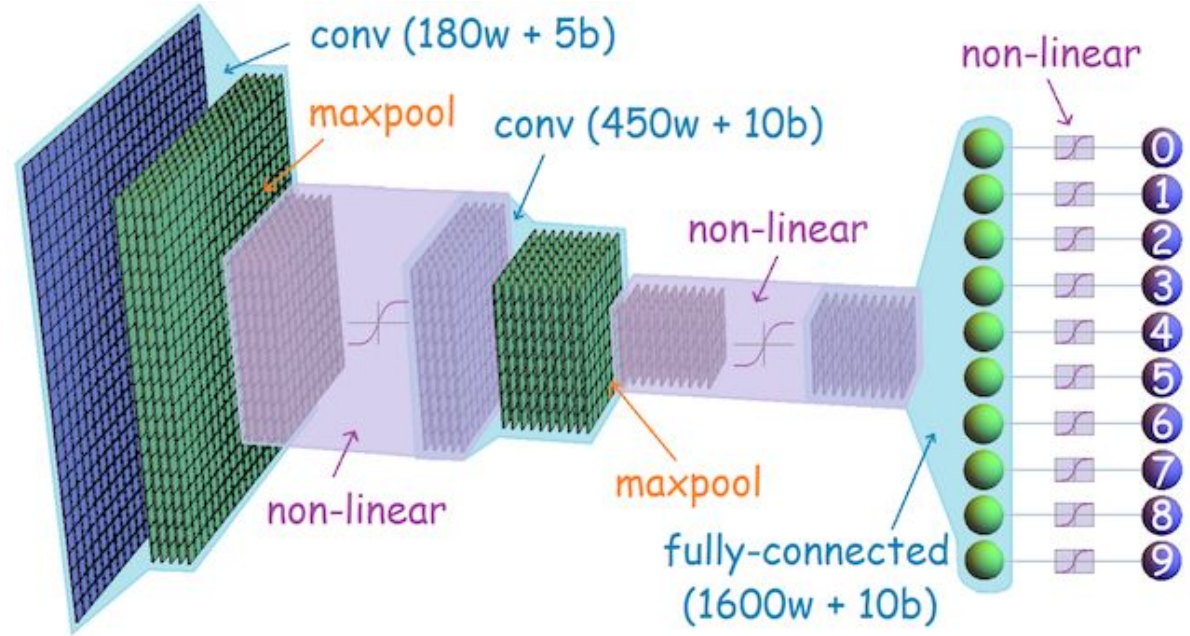


Image classification

Feature extraction is the input of the dense layer classification stage

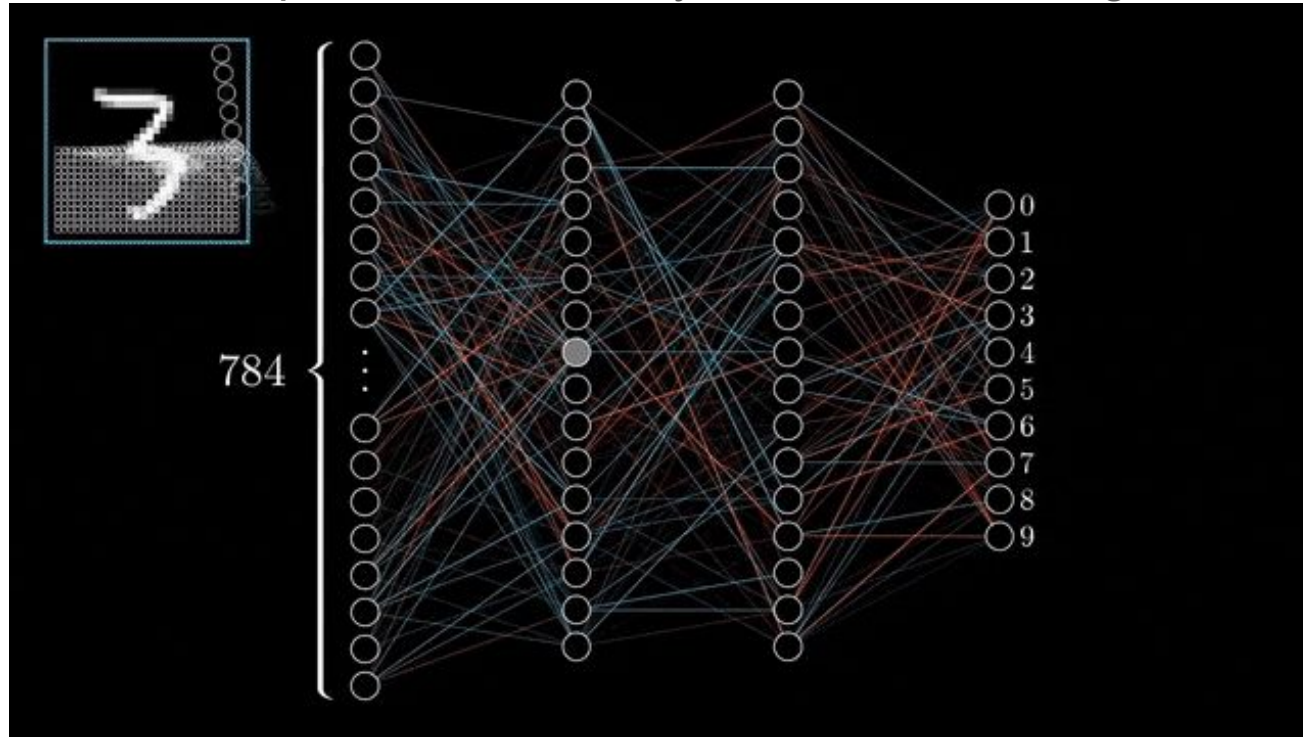


Image classification

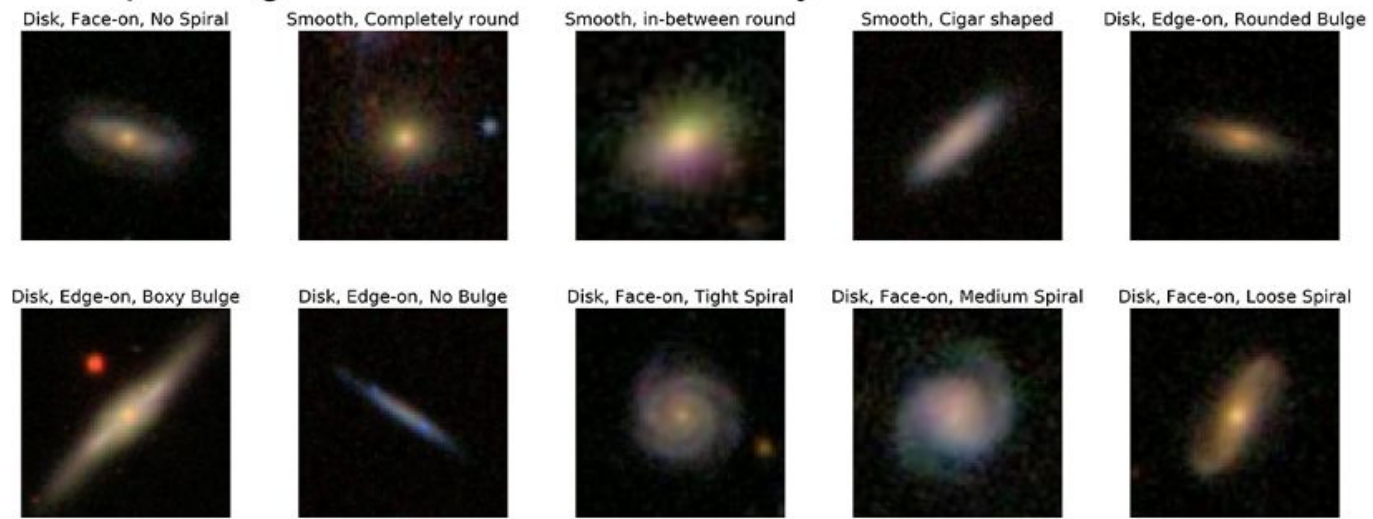
Galaxy10 toy dataset

([astroNN web](#))

Galaxy10 dataset (21785 images)

- Class 0 (3461 images): Disk, Face-on, No Spiral
- Class 1 (6997 images): Smooth, Completely round
- Class 2 (6292 images): Smooth, in-between round
- Class 3 (394 images): Smooth, Cigar shaped
- Class 4 (1534 images): Disk, Edge-on, Rounded Bulge
- Class 5 (17 images): Disk, Edge-on, Boxy Bulge
- Class 6 (589 images): Disk, Edge-on, No Bulge
- Class 7 (1121 images): Disk, Face-on, Tight Spiral
- Class 8 (906 images): Disk, Face-on, Medium Spiral
- Class 9 (519 images): Disk, Face-on, Loose Spiral

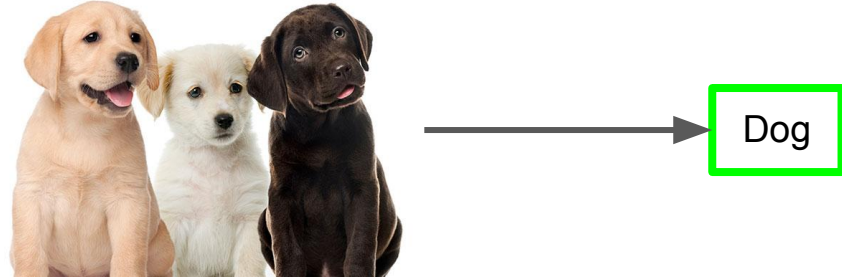
Example images of each class from Galaxy10 dataset



Outline

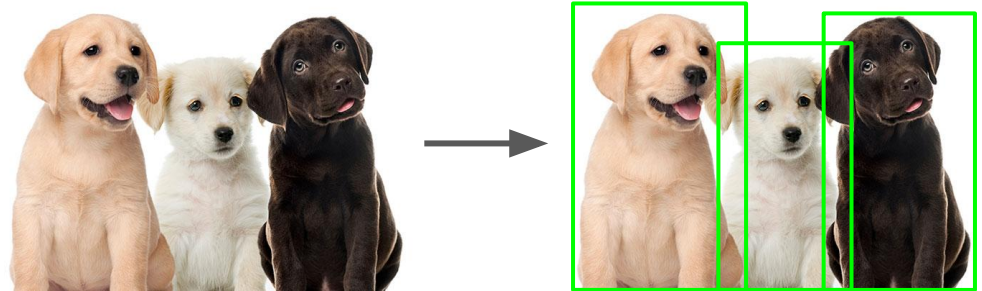
- Image classification

- A brief summary
- Galaxy classification



- Object detection

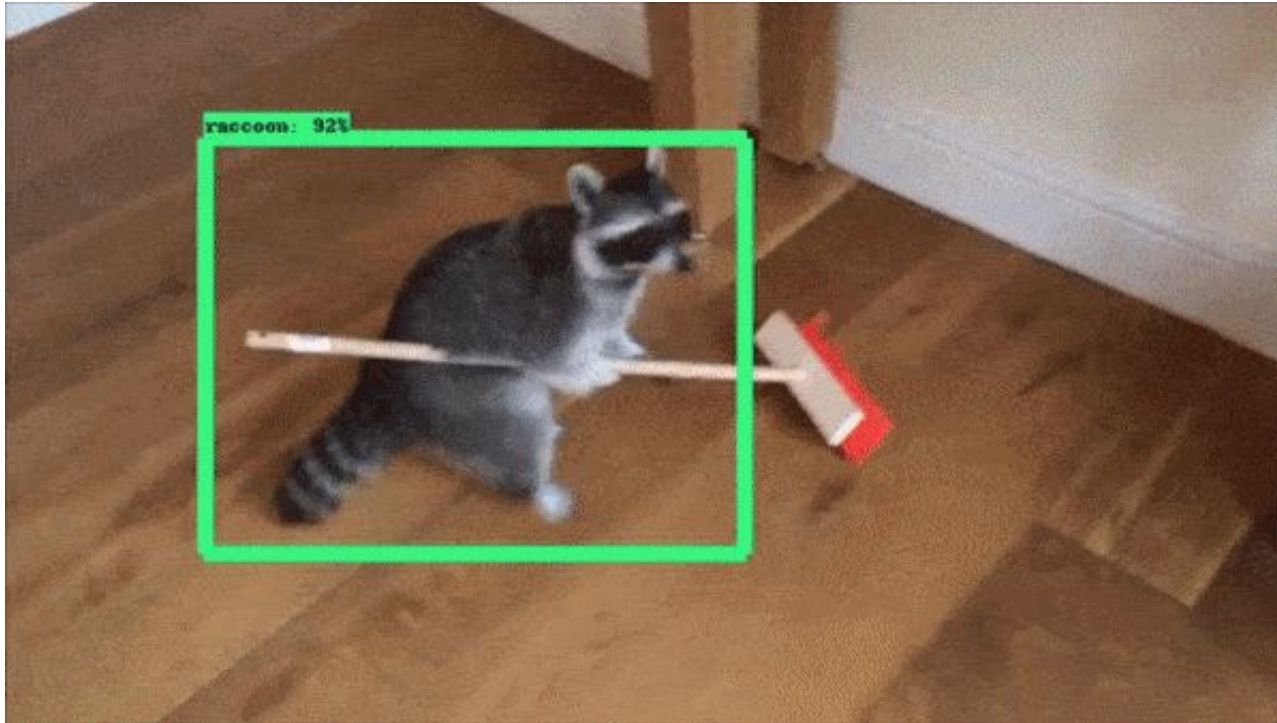
- Introduction
- State-of-the-art meta-architectures
- Practical example of detection



- Practical: Classification of galaxies

Object detection

Provide the object (class) and location (bounding box/region)



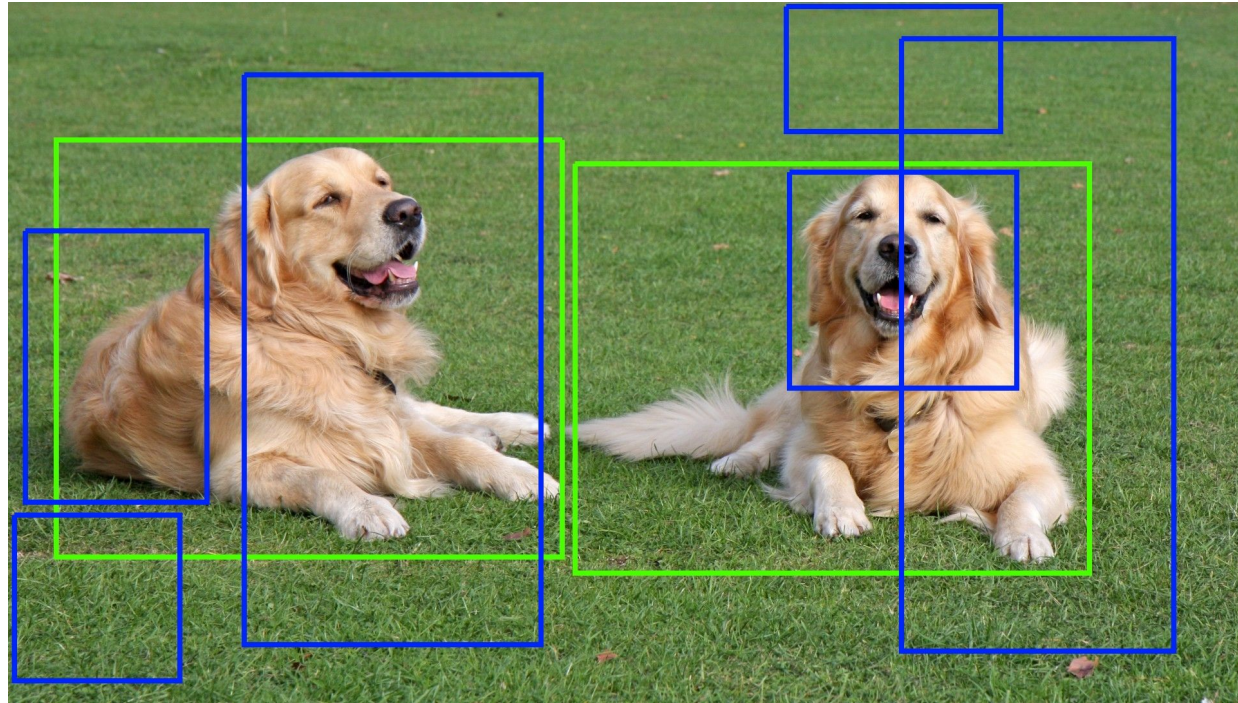
Object detection

Object detection models:

Simultaneously learn

- Region proposal
- Object classification

Output a set of candidates
regions-class and confidence

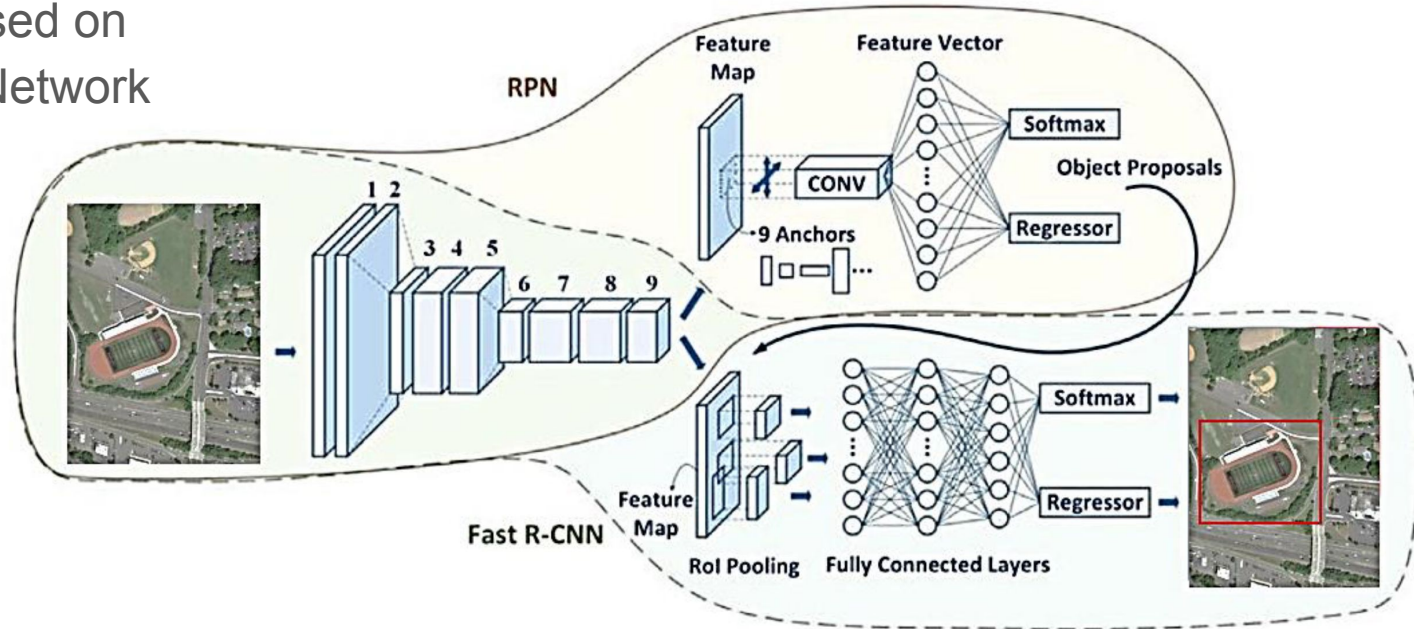


Object detection

Faster R-CNN architecture:

Region proposal based on a Region Proposal Network

- Slower
- More accurate
- Small objects



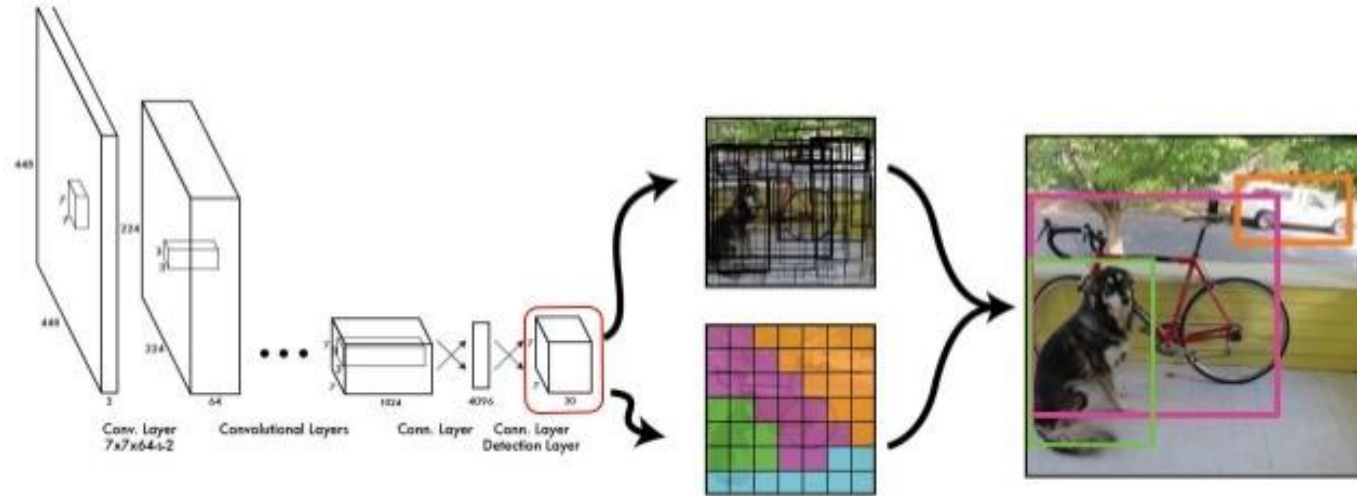
Object detection

SSD or YOLO

Region proposal based
on grid

YOLO: You Only Look Once

- Faster
- Less accurate
- No good with small objects



Object detection

Detect fiery looping rain on the Sun

[Video](#)

A recurrent pattern in a structured signal can be learned

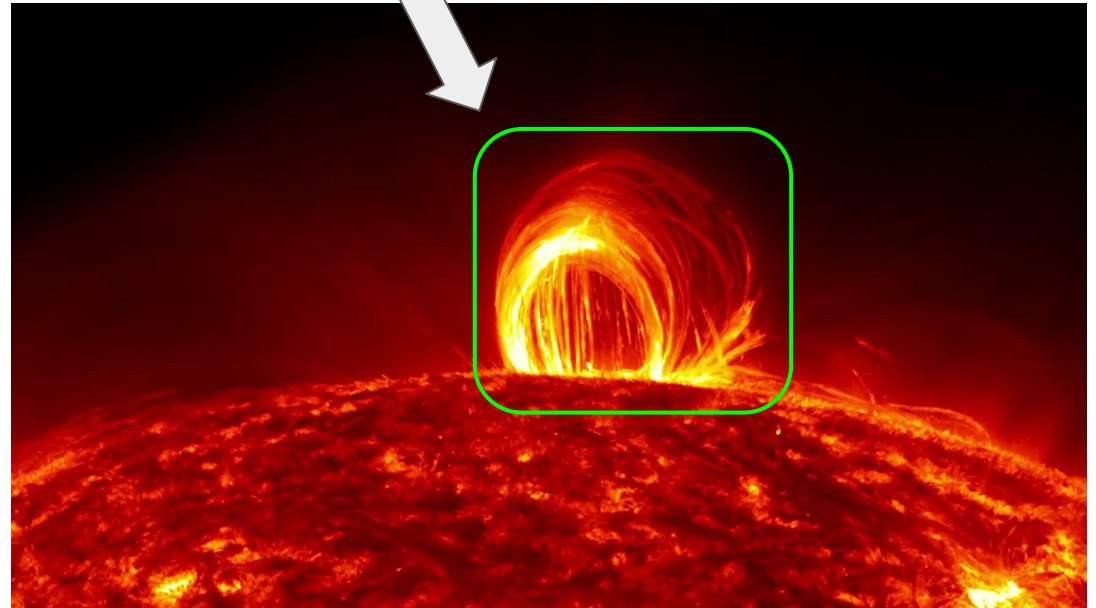
- RGB images
- Texture, color, shape

Experiment configuration

[Download](#)

(X, Y, W, H)

Location
Color properties
Dimension
Movement
...



Object detection

Detection annotation

- Image information
- Objects
 - Class
 - Region

```
1 <annotation>
2   <folder>train</folder>
3   <filename>FieryLoopingRainSun-187.jpg</filename>
4   <path>/home/alcasla/Escritorio/Astro/Solar-FieryLoop/train/FieryLoopingRainSun-187.jpg</path>
5   <source>
6     <database>Unknown</database>
7   </source>
8   <size>
9     <width>1280</width>
10    <height>720</height>
11    <depth>3</depth>
12  </size>
13  <segmented>0</segmented>
14  <object>
15    <name>fiery_loopw</name>
16    <pose>Unspecified</pose>
17    <truncated>0</truncated>
18    <difficult>0</difficult>
19    <bndbox>
20      <xmin>699</xmin>
21      <ymin>376</ymin>
22      <xmax>820</xmax>
23      <ymax>471</ymax>
24    </bndbox>
25  </object>
26 </annotation>
27
```

Object detection

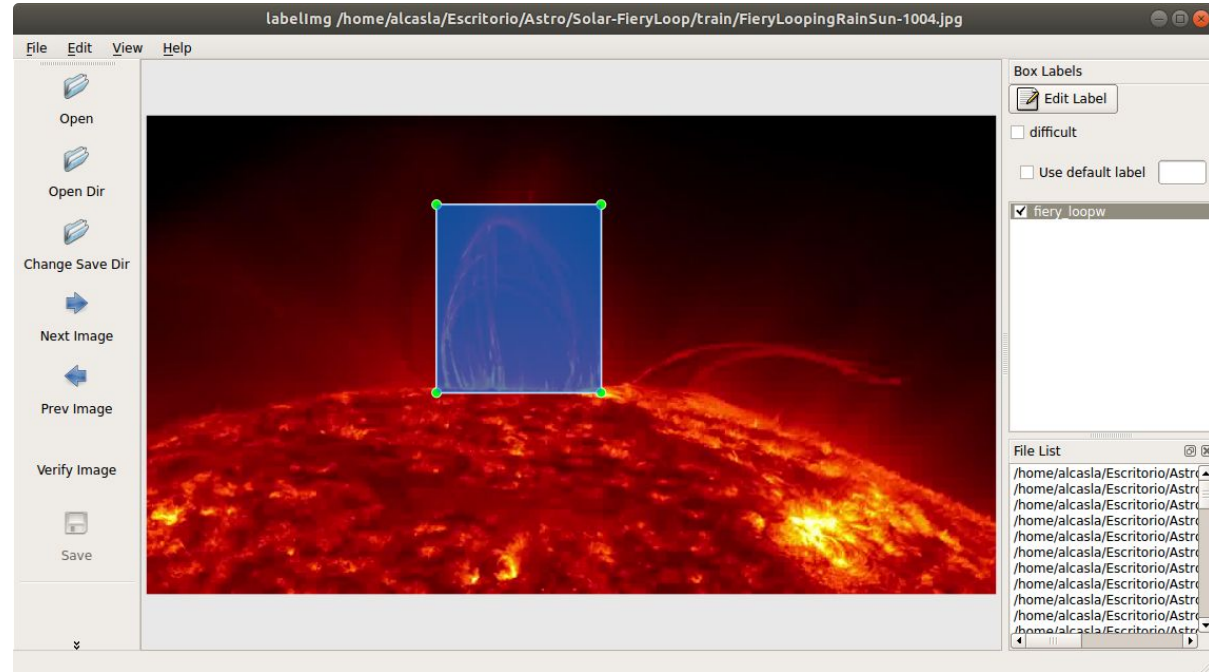
LabelImg

Repository

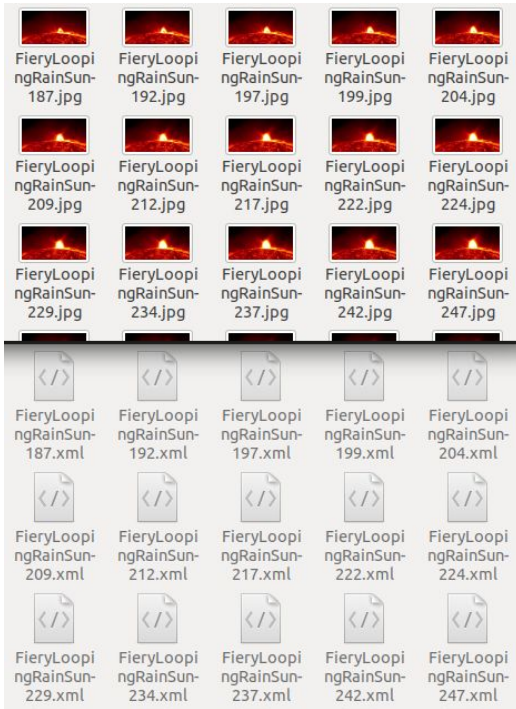
github.com/tzutalin/labelImg

Installer file

tzutalin.github.io/labelImg/



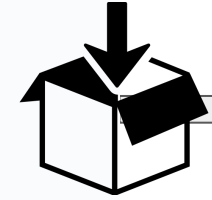
Object detection



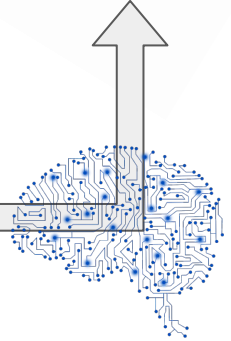
Data preparation



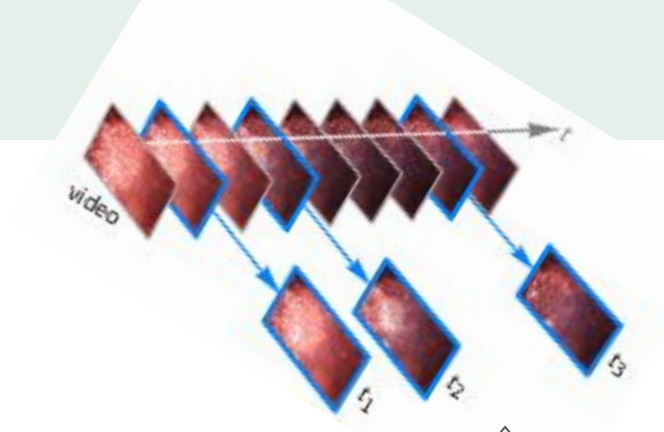
[Configuration and training]



[Packing]



[Inference]



Object detection

Object detection framework

github.com/tensorflow/models/tree/master/research/object_detection

Software configuration

github.com/spsrc/somachine2021

Build conda environment using .yml file

Model Zoo

We provide a large collection of models that are trained on several datasets in the [Model Zoo](#).

Guides

- [Configuring an object detection pipeline](#)
- [Preparing inputs](#)
- [Defining your own model architecture](#)
- [Bringing in your own dataset](#)
- [Supported object detection evaluation protocols](#)
- [TPU compatible detection pipelines](#)
- [Training and evaluation guide \(CPU, GPU, or TPU\)](#)

Extras:

- [Exporting a trained model for inference](#)
- [Exporting a trained model for TPU inference](#)
- [Inference and evaluation on the Open Images dataset](#)
- [Run an instance segmentation model](#)
- [Run the evaluation for the Open Images Challenge 2018/2019](#)
- [Running object detection on mobile devices with TensorFlow Lite](#)
- [Context R-CNN documentation for data preparation, training, and export](#)

Outline

- Image classification
 - A brief summary
 - Galaxy classification
- Object detection
 - Introduction
 - State-of-the-art meta-architectures
 - Practical example of detection
- Practical: Classification of galaxies