Object Detection with Faster RCNN

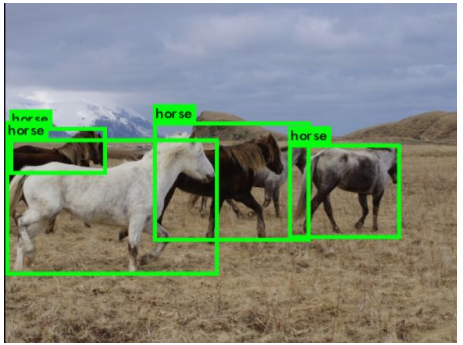
Núria Marzo, JuanJo Nieto, Rafel Palliser and Sergi Sánchez

Outline

1. Introduction
2. Motivation
3. Faster RCNN
4. First approach: Keras + Traffic lights dataset.
5. Final approach: TensorFlow API
6. Video detection approach
7. Conclusions

Introduction

- Undergraduate students.
- Not much experience with ML, DL or CV.
- We wanted to use:
  a. Intuitive software: Keras.
  b. Defined problem: Object detection.
- We have faced multiple problems.
- As trainings are very slow we have not been able to correct the mistakes in a fast way.



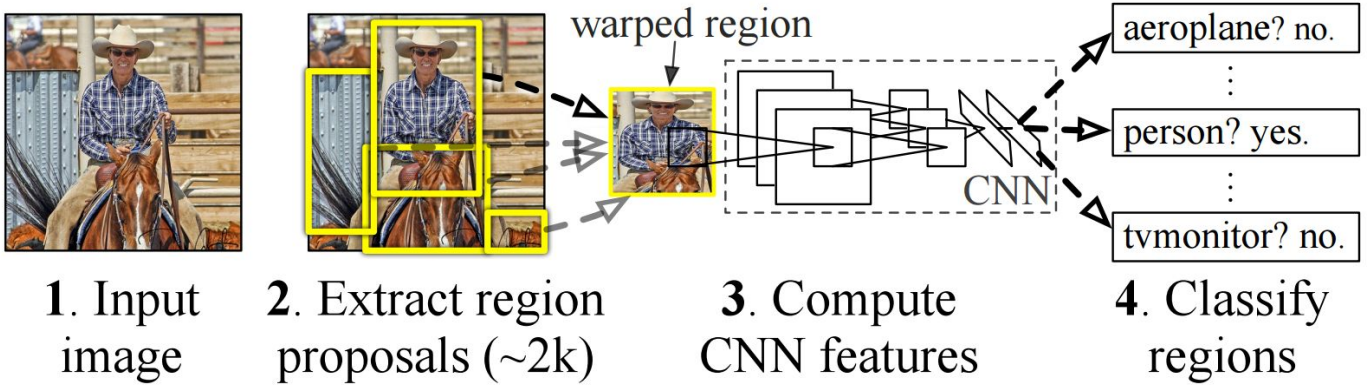
Motivation

- Detect traffic lights. Detect the state: **GO STOP WARNING**.
- We found a good dataset with 4 GB of sequences with images for testing and training. 15K train (with +35K traffic lights) and 10K test images.
- Good annotations: Bounding boxes and state.

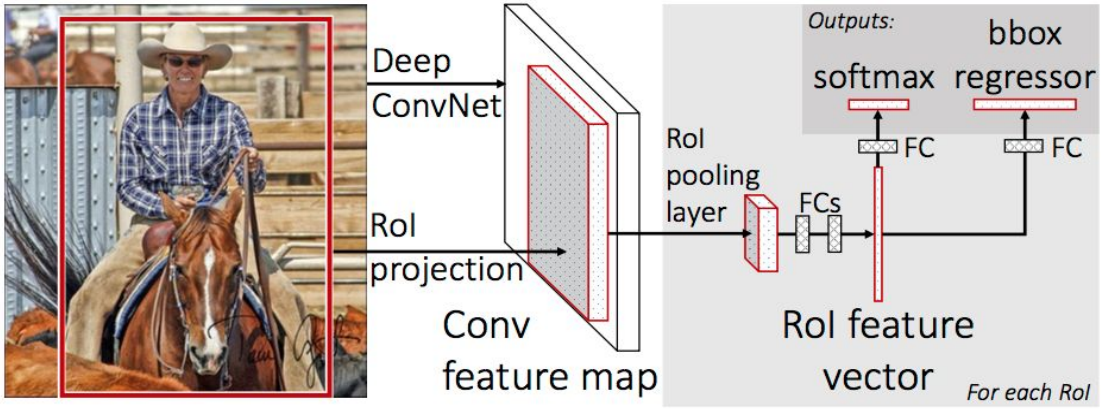



Faster RCNN

- R-CNN:

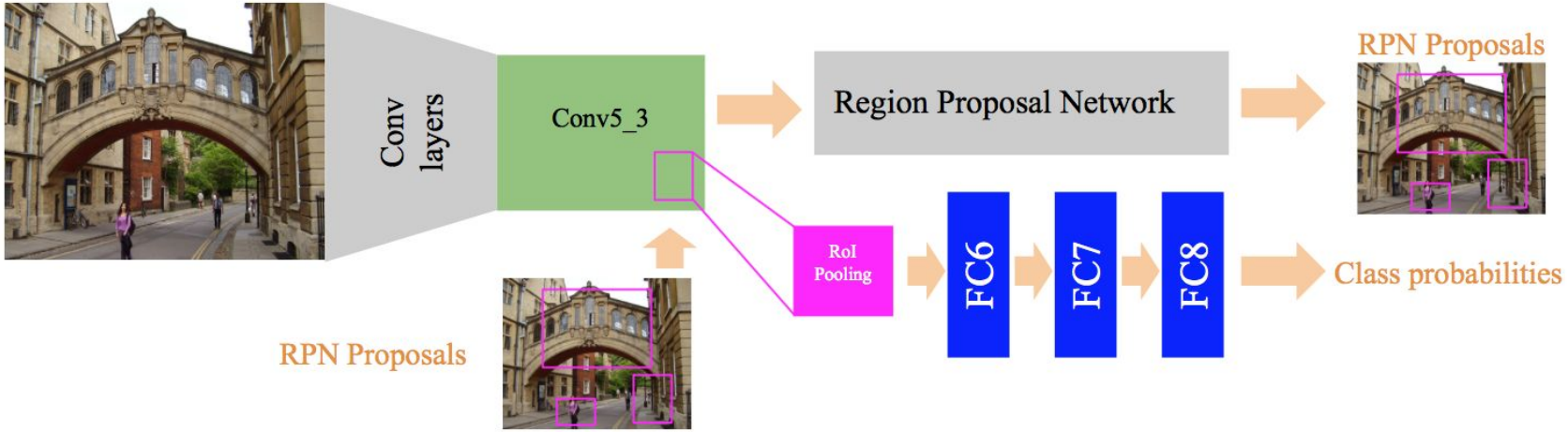


- Fast R-CNN:



Faster RCNN

- Faster RCNN:



First Approach: Keras + Traffic Lights

- Found a GitHub repo with the Fast R-CNN implementation:

  Keras Fast RCNN GitHub Repo 

- There was no pretrained model to use.
- We started to train the model with the dataset found in Kaggle:

  LISA Traffic Light Dataset - Kaggle kaggle

First Approach: Keras + Traffic Lights

- We obtained some results:

Model trained with 2 images to overfit.

Final Approach: TensorFlow API

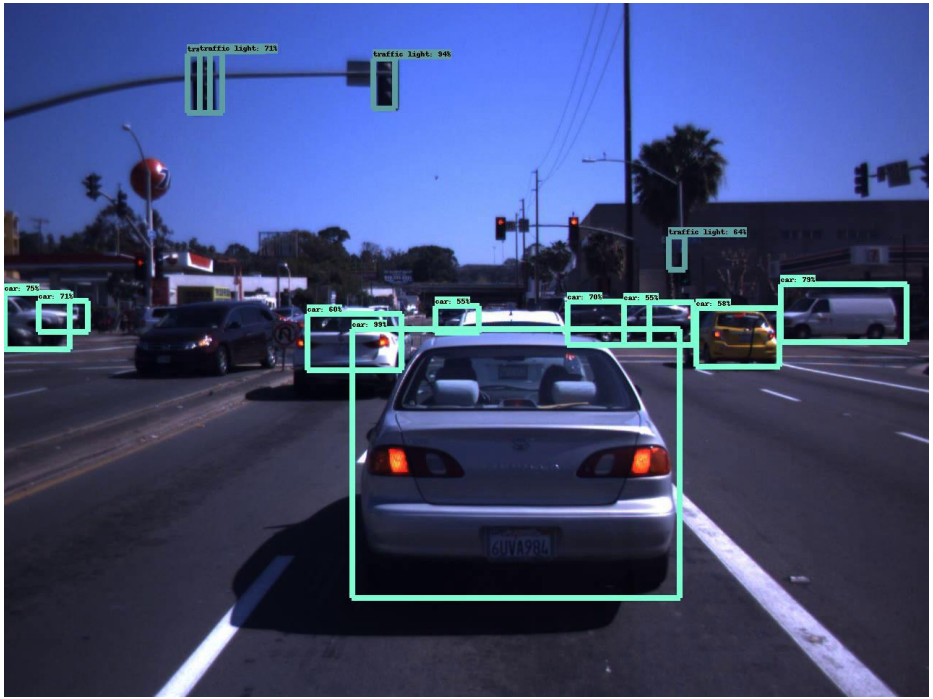
- Github OFFICIAL TensorFlow repo.

Object Detection TensorFlow GitHub Repo

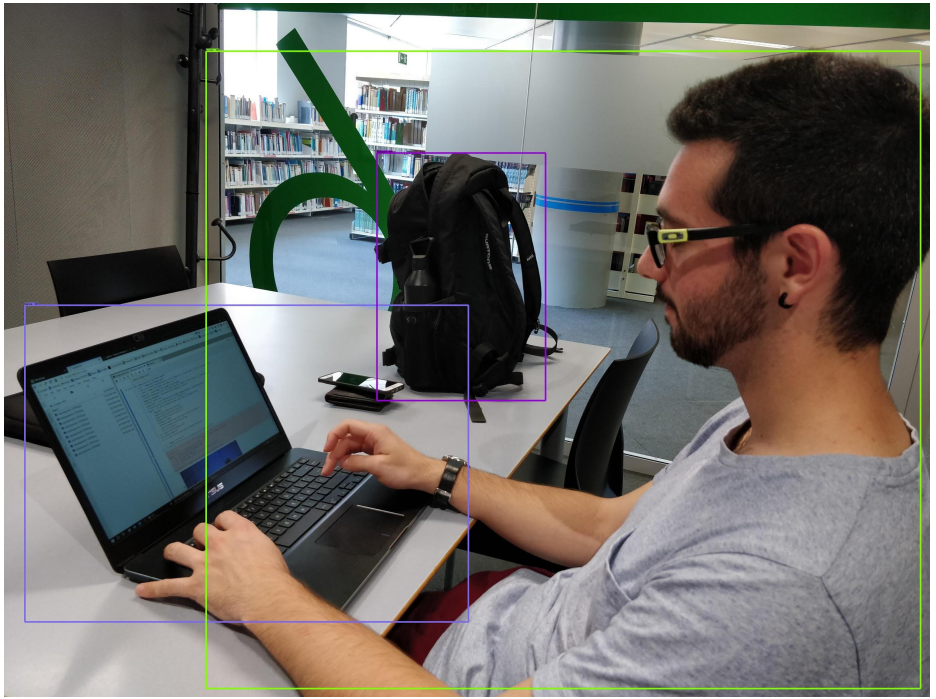
- Installed everything and prepared the server to try to detect objects with an already trained model.
- 90 classes for objects such as: car, traffic light, person, backpack, laptop.

Final Approach: TensorFlow API - We obtained results:

Detected: traffic lights and cars.



Detected: person, laptop and backpack.



Final Approach: TensorFlow API

- Detector works! New goal → Train with our own dataset
- We used the pre-trained weights: "**Faster R-CNN Inception v2**"
- Steps:
  a. Prepare the data to be in the correct format. CSV to TF-Records
  b. Train a model with our dataset and **9 classes**
  c. Test the model.

Final Approach: TensorFlow API

- First train: 55K steps, ~4h

Final Approach: TensorFlow API

- Second train: From checkpoint, raise limit to 150K steps, ~8h

Video detection approach

Conclusions

- TensorFlow API codes.
  - Work with our dataset of traffic lights
- Results although not so good
- First time playing with NNs and frameworks. We have learned a lot

Thank you!

Questions?