UNIVERSITÀ
DI TRENTO

Automatic
License Plate Detection

Course
Signal, Image and Video

Master Degree in AIS

Department of Information Engineering
and Computer Science

Academic Year 2021/2022

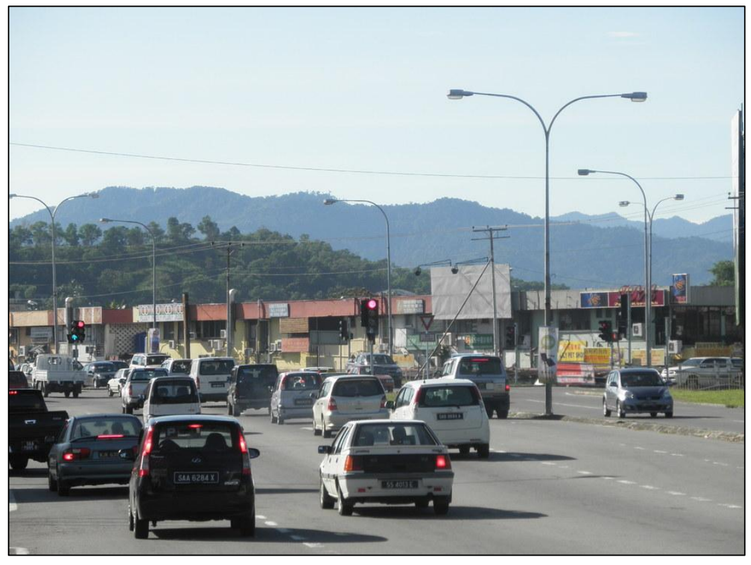
Students
Nicola Farina
Luca Zardini

Objective

Detect license plates in images under different conditions

No fixed camera

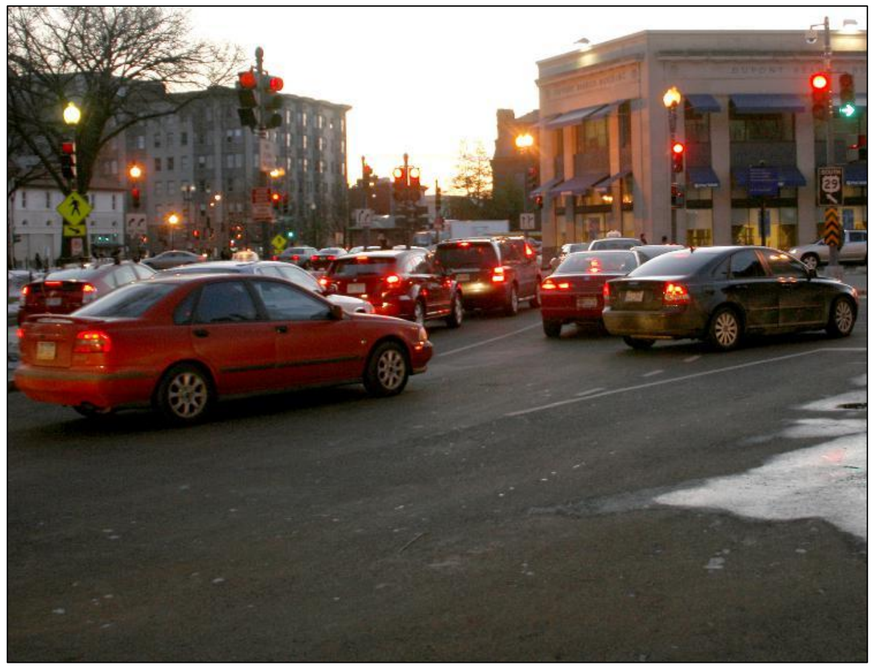
Light conditions



Multiple license plates

Datasets

OIDv4
- 6871 images
- “In the wild”
- GitHub



AOLP
- 2049 images
- “Easy dataset”
- GitHub

Methods

- **Implemented and delivered:**
  - Image Processing
  - Convolutional Neural Network

- **Not delivered:**
  - Image processing + text detector
  - Cascade classifiers with Haar features

Image processing

- **Preprocessing:**
  - Gray scale
  - Bilateral filter
  - Contrast enhancement

- **Canny edge algorithm**
  - Dilation
  - Closing

- **Contours and license plate candidates**

- **Sobel algorithm**

Starting image example

Image processing

- **Preprocessing**

Gray scale

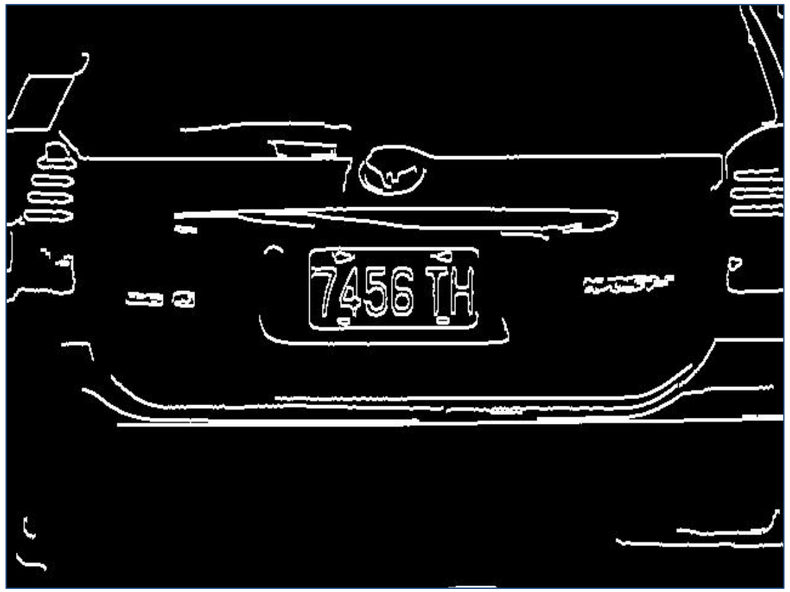
Bilateral filter

Contrast enhancement

Image processing

- **Canny edge algorithm**

Canny edges



Dilation

Closing

Image processing

- **Contours and license plate candidates**

All contours

Rectangular boxes around contours

Candidates after filtering out by criteria

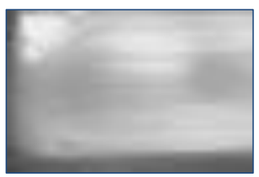
Image processing

- **Sobel algorithm**

Region 1

Region 2

Region 3



Region 4

Original image

Sobel on vertical direction

Removed 85th percentile

Image processing

- **Result**

Image processing - Assumptions

- License plates are **mostly white** (mean pixel value >= 100)
- License plate **area** falls **in a given range** (between 1605 and 35000 px^2)
- There is **enough contrast** between license plate and the rest of the car
- Only **one** license plate for image

Image processing - Results

- **Accuracy** of 52% on a simple dataset
- Sometimes bounding boxes are not accurate

Convolutional Neural Network

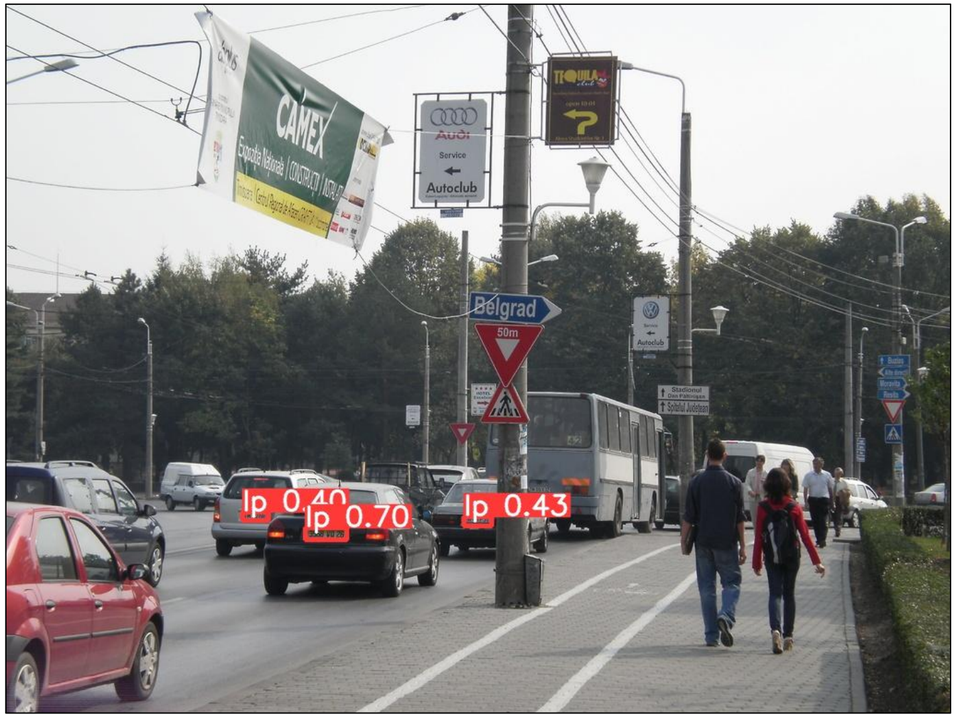
CNN

- **YOLOv5** family [1]
- **Pre-trained** YOLOv5 nano
- **Trained** it over **OIDv4** dataset
- Wrote a Python **interface** to use our model
- **Hardware limitations**
  - 100 epochs
  - 640x640 images resolution

[1] https://github.com/ultralytics/yolov5

CNN - Results

	Precision	Recall	mAP@.0.5	mAP@.5:.0.95
Train	0.94	0.88	0.92	0.68
Test	0.94	0.85	0.90	0.62



Comparison

Yolo

Image Processing

Other methods

Image processing + text detector (EAST detector)

- Apply image processing
- For each candidate use text detector
  - Solves detection of headlights, bumpers etc.

Why we didn’t upload the implementation:

- Text detector has lots of false positives
- Very small accuracy improvement

Other methods

Cascade classifiers with Haar features

- Used in some papers
- Tried some ready-made implementations found online

Why we didn’t upload the implementation:

- Performance was extremely poor
- **Future work:** experiment more with this method

Conclusion

Implemented two solutions

- Pure **image processing**
  - Accuracy of 52% on easy dataset
  - One plate detection for each image
- Convolutional **neural network**
  - mAP around 0.68
  - Multiple plates detection

Future work

- Experiment more with image processing techniques