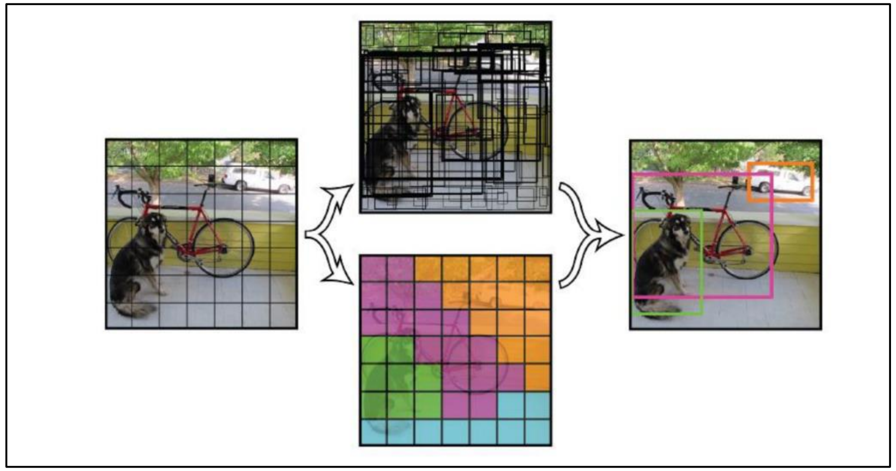
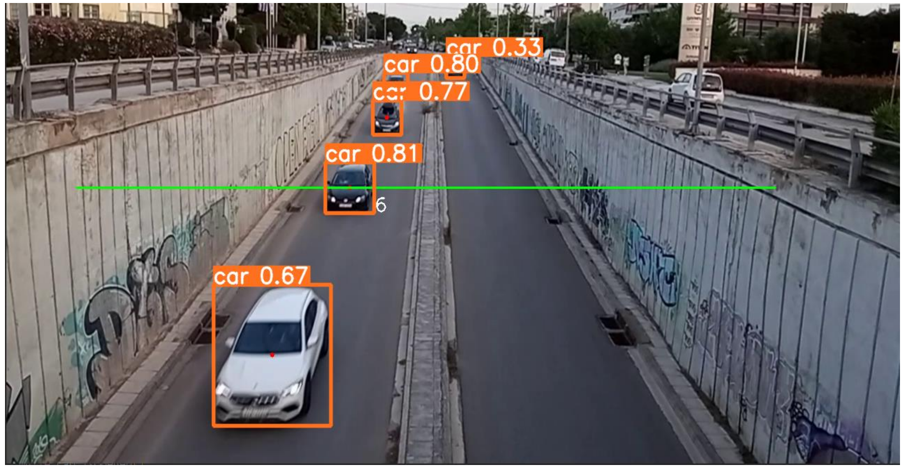
NCSR Demokritos – University of Piraeus

MSc in Artificial Intelligence

Assignment presentation for Machine Learning on Multimedia data course

Vehicle detection and counting from a static point using a camera





Pavlos Patsonis
3 July 2023

Outline

- Problem statement
- YOLO algorithm
- Energy of audio signal
- Implementation
- Results - Conclusions
- Future improvements

Problem statement

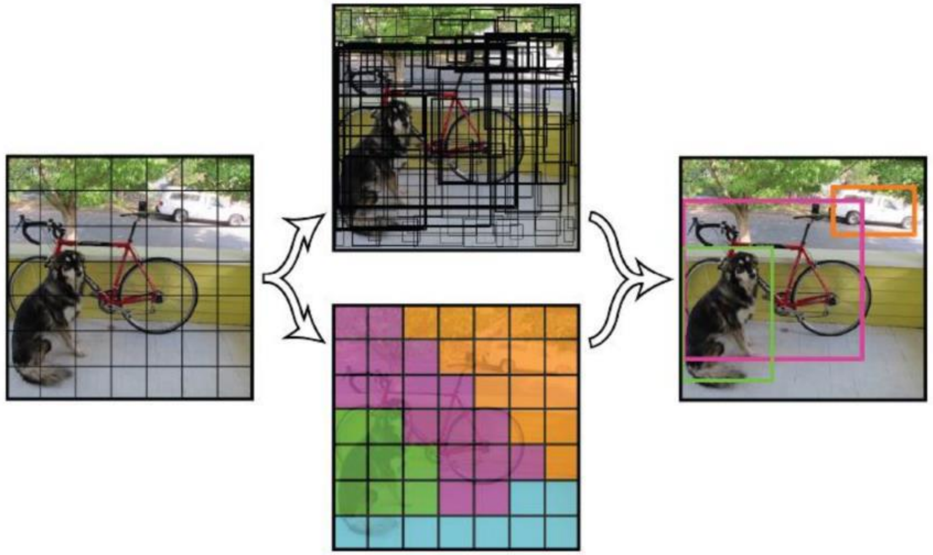
Basic goal: Detect and Count vehicles in static videos

Methodology: Combine audio and visual information in real-time

Evaluation: Computation of mean absolute error of 10 videos

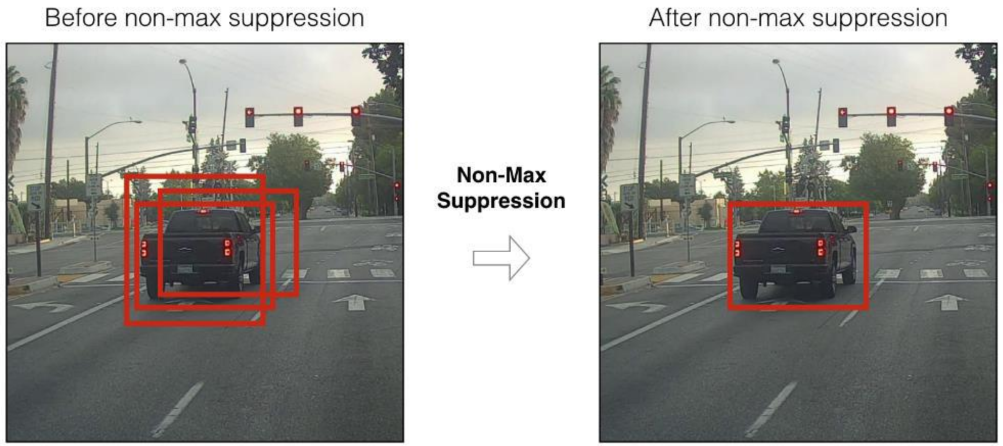
What is the YOLO algorithm?

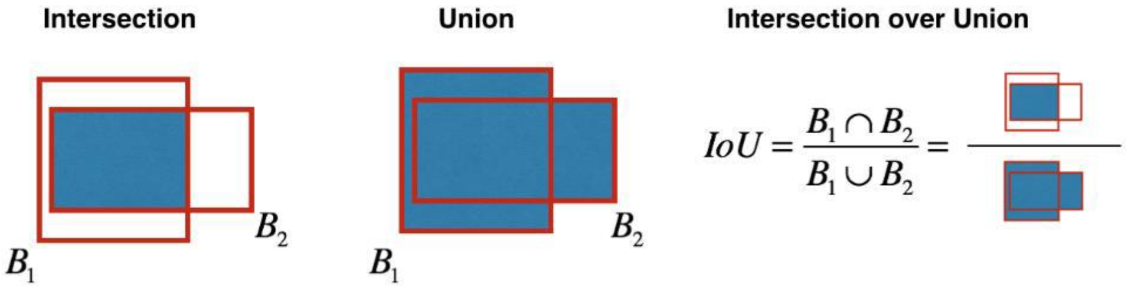
- YOLO stands for *You Look Only Once*
- A pre-trained CNN suitable for object detection
- Break image into cells
- Detect objects with highest probs in each cell
- Combine detected objects from different cells to detect objects in image
- Ideal for real-time detections



Non-max suppression (NMS)

Intersection over union (IoU)

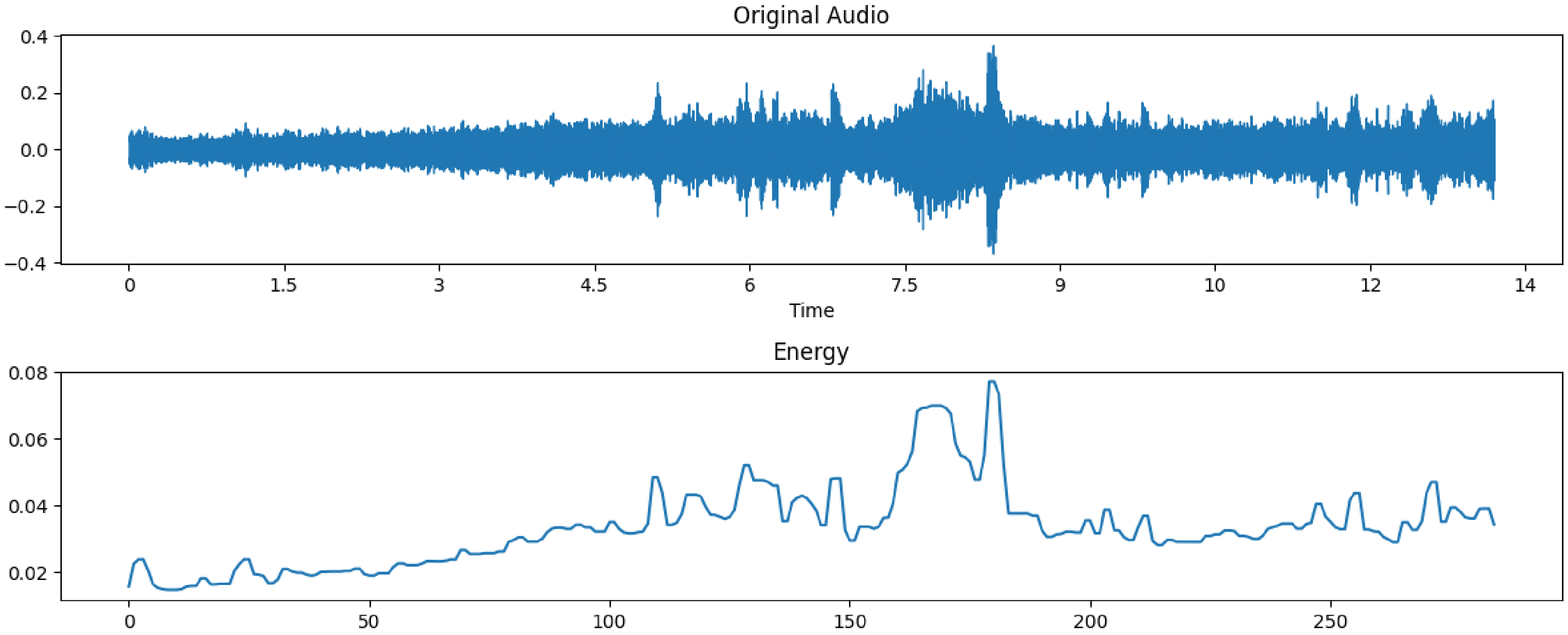




- *NMS* removes redundant bounding box predictions
- Use of *IoU* measurement to achive filterings

Energy of audio signal

Root Mean Square



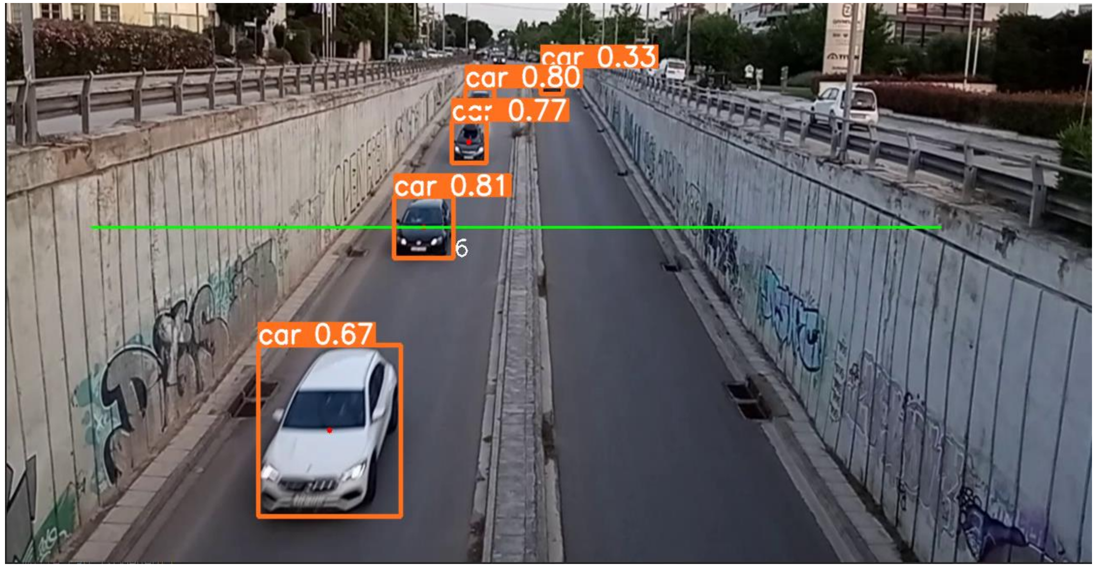
$$RMS = \sqrt{\frac{1}{N}\sum_{i}^{N} {x_i}^2}$$

where x_i: individual audio sample
N: Number of total sample

Implementation

Basic Idea: Use a trigger line on video to determine if a vehicle is passing by.

1. Setting up experiment
    a. Find optimal location
    b. Shoot video data
    c. Split to 10 evaluation videos
    d. Calculate actual counts of each video
2. Gather useful information
    a. Frame resolution – drawing decision line
    b. Choosing the vehicle classes from YOLO
    c. Computing audio frame from video FPS
    d. Define Energy Threshold value for audio signal



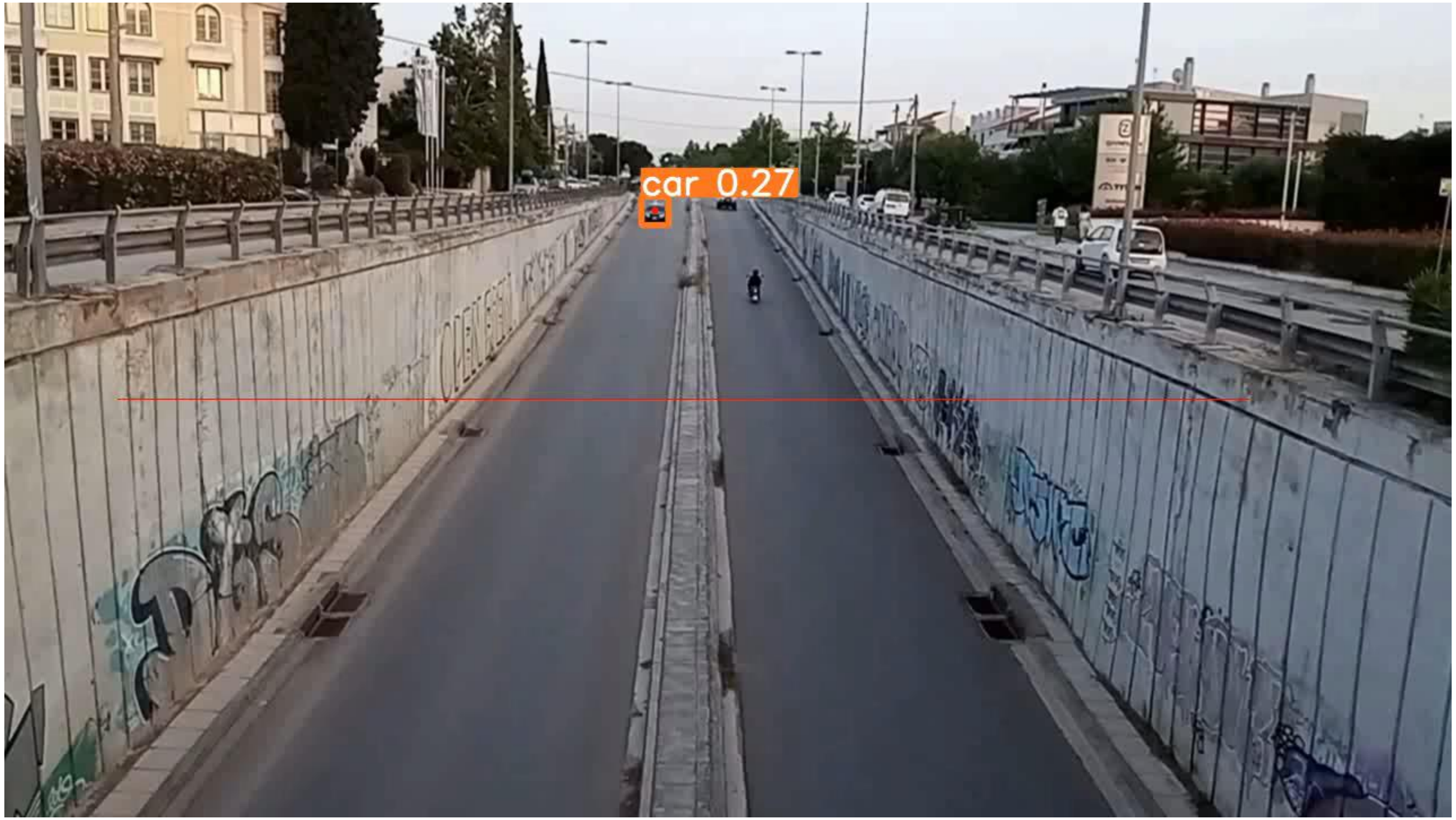
Implementation

For each of the 10 evaluation videos:

1. Locate and track objects considered vehicles by YOLO **frame by frame**
2. Place center point in each bounding box
3. Calculate E_{RMS} of **respective audio frame**
4. Combine YOLO and audio information to trigger counter
5. Calculate and store absolute error

Calculate mean absolute error (MAE) of all evaluation videos

Example



Results

video index	1	2	3	4	5	6	7	8	9	10
model_counts	6	3	10	7	2	5	8	3	5	4
actual_counts	5	4	10	7	4	6	7	5	5	4

- ● model_counts > actual_counts
- ● model_counts = actual_counts
- ● model_counts < actual_counts

Comments:

- Not large deviation from actual counts
- Both false positives & negatives
- Counter in a way is “conservative” (false negatives > false positives)

$$MAE = \frac{\sum_{i=1}^{\mathrm{N}} |model\_counts_i - actual\_counts_i\,|}{\mathrm{N}} = 0.8 \text{ vehicles}$$

where, MAE: mean absolute error

N: Number of evaluation videos = 10

Future improvements

1. Reduce False Positive & Negative counts
    - Use camera with more FPS
2. Experiment Redesign
    - Choose different location
    - Use of directional microphones
3. Better utilization of audio information
    - Explore more sophisticated audio features (spectral analysis)
    - Develop model for vehicle detection trained on vehicle audio dataset

Thank you for your patience!