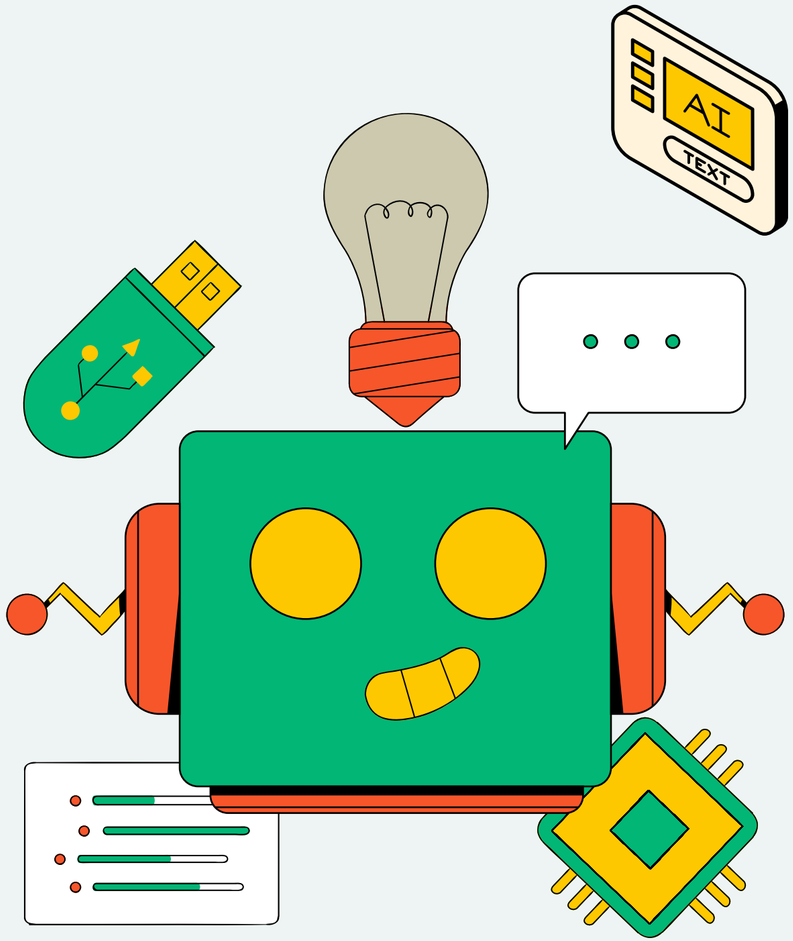
universidade de aveiro

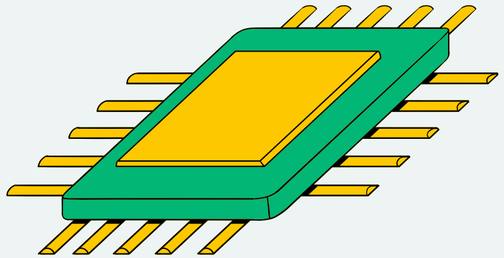


POTHOLE OBJECT DETECTION

PRESENTED BY:

GONÇALO SILVA, 103244, MECT

SAMUEL TEIXEIRA, 103325, MECT



PRESENTATION OUTLINE

- Statistics – potholes
- Introduction
- Related Work/ State of Art
- Dataset
- Training, Validation and Testing
- Results
- Future Work

STATISTICS - POTHOLES

UNITED KINGDOM (UK)

- Number has increased to around **1 million**, with about **6** per **mile** (1.6 km)
- Incidents involving them, rose from **21,725** in 2020, to **29,333** in 2024

UNITED KINGDOM (UK)

- Pothole-related incidents cause about **1%** of all **road accidents**
- In a survey, **31%** of **VRU's** had been involved in **accidents** or near misses due to poor road surfaces, including **potholes**

UNITED STATES (US)

- Is estimated that potholes cost drivers **$3 billion** annually in vehicle **repairs**.
- In **NYC**, potholes and road defects cost **$138 million** in **settlements** for pedestrian injuries and vehicle damage

INTRODUCTION

Potholes are a widespread problem in urban and rural road networks, posing serious risks to transportation infrastructure, vehicle safety, and human life.

This project aims to develop a robust system for the **real-time** detection and classification of potholes, with a focus on **accuracy**, **speed**, and **practicality**

RELATED WORK/ STATE OF ART

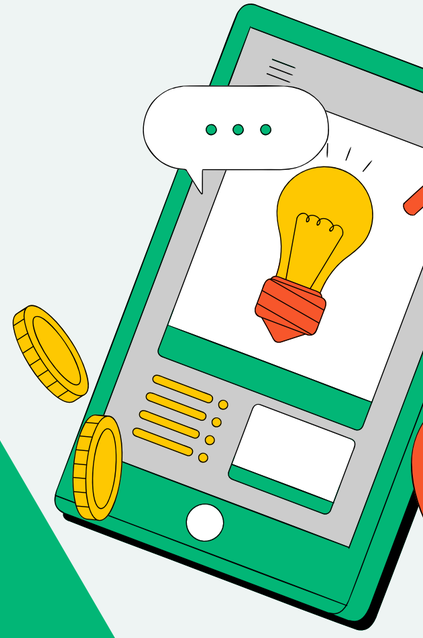
Traditional Machine Learning

Object Detection

CNN

Deep Learning

YOLO



TRADITIONAL MACHINE LEARNING

01

GENERALIZED HOUGH TRANSFORM

Used geometric patterns for detection but struggled with changes in size, rotation, and grayscale values.

02

HARRIS CORNER DETECTION

Identified points with significant intensity variation but was sensitive to transformations.

03

SIFT (SCALE-INVARIANT FEATURE TRANSFORM)

Addressed rotation and scale changes but required costly implementation for different use cases

DEEP LEARNING

Transition from Traditional Methods:

- No more extensive feature engineering!
- Enabled processing of large datasets with improved accuracy and efficiency.

Key Milestones:

- Introduction of Region-Based Convolutional Neural Networks (R-CNN).
- Rapid advancements in techniques like Fast R-CNN, Faster R-CNN, and Mask R-CNN.

Advantages:

- Unified frameworks, scalability, and the ability to learn features directly from data.

CONVOLUTIONAL NEURAL NETWORKS (CNNS)

R-CNN (2014)

High accuracy but computationally intensive

FAST R-CNN (2015)

Enabled single forward-pass processing for efficiency

FASTER R-CNN (2015)

Optimized for real-time performance w/ less computational load

---

SELECTIVE SEARCH

Identify Regions of Interest

EXTRACT

Features from each region

CLASSIFY

Each region and refine bounding boxes

YOLO (YOU ONLY LOOK ONCE)

WHAT

- **Unified** detection pipeline simplifies detection into a **single-shot** process
- **Directly** predicts bounding boxes and class probabilities from image pixels.

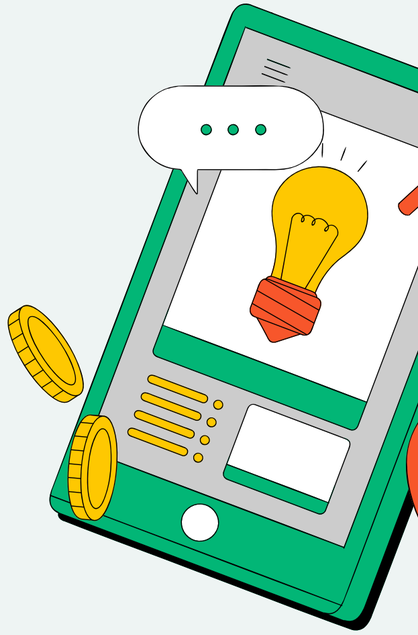
ADVANTAGES

- **Real-time** performance/ speec (no multi-stages)
- Only 1 CNN is used, **simplifying** the processs

LIMITATIONS

- **Struggles** with **small** and **overlapping** objects in the same grid.
- Accuracy **lower** than region-based networks like **Faster R-CNN**.

DATASET



Data augmentation

Dataset

Variety

Small vs. Full Dataset

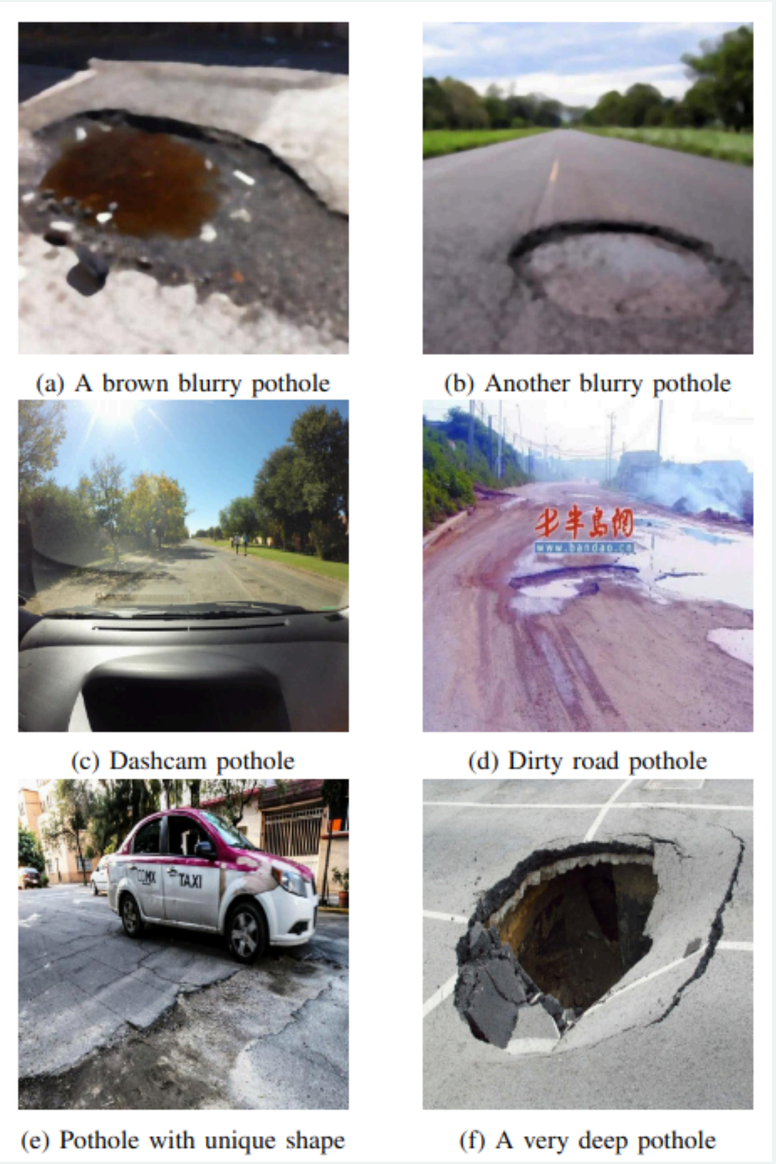
DATASET

Roboflow dataset, composed of **11068** images for training (80%), **1808** images for validation (7%) and **891** images for testing (13%).

Big variety in the data, prevents **Data leakage.**

Some data **augmentation** was used.

Since computational resources were limited, we tested the impacts of a reduction in the dataset (~80%), composed of **2400** images for training (80%), and **300** for validation and testing, each (10%)



(a) A brown blurry pothole

(b) Another blurry pothole

(c) Dashcam pothole

(d) Dirty road pothole

(e) Pothole with unique shape

(f) A very deep pothole

TRAINING, VALIDATION AND TESTING

Batch Size

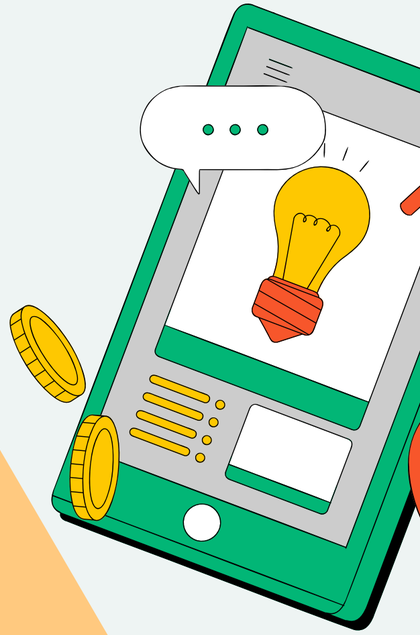
Epochs

Reduced vs. Full dataset

Patience

Optimizer

YOLO Versions



BATCH SIZE

Number of training examples processed per iteration

Low batch size

Reduces overfitting

Higher Variance in Gradient Estimates

High batch size

Smooth Convergence

Careful choice of LR (less updates -> less generalization)

No relevance in our case

EPOCHS

Number of times the model goes through all the dataset

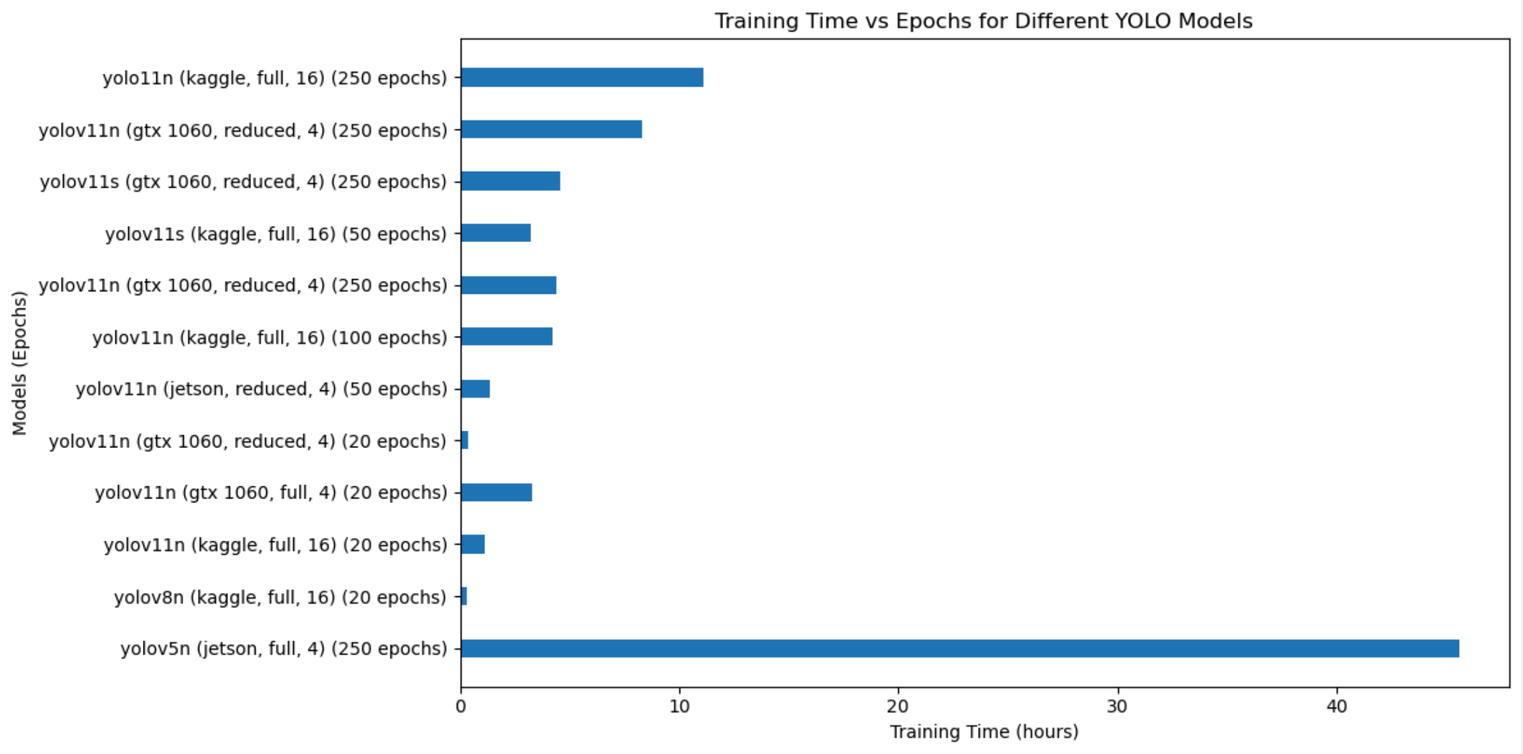
$$\text{Iterations per Epoch} = \left\lceil \frac{\text{Total Images}}{\text{Batch Size}} \right\rceil$$

Few epochs–

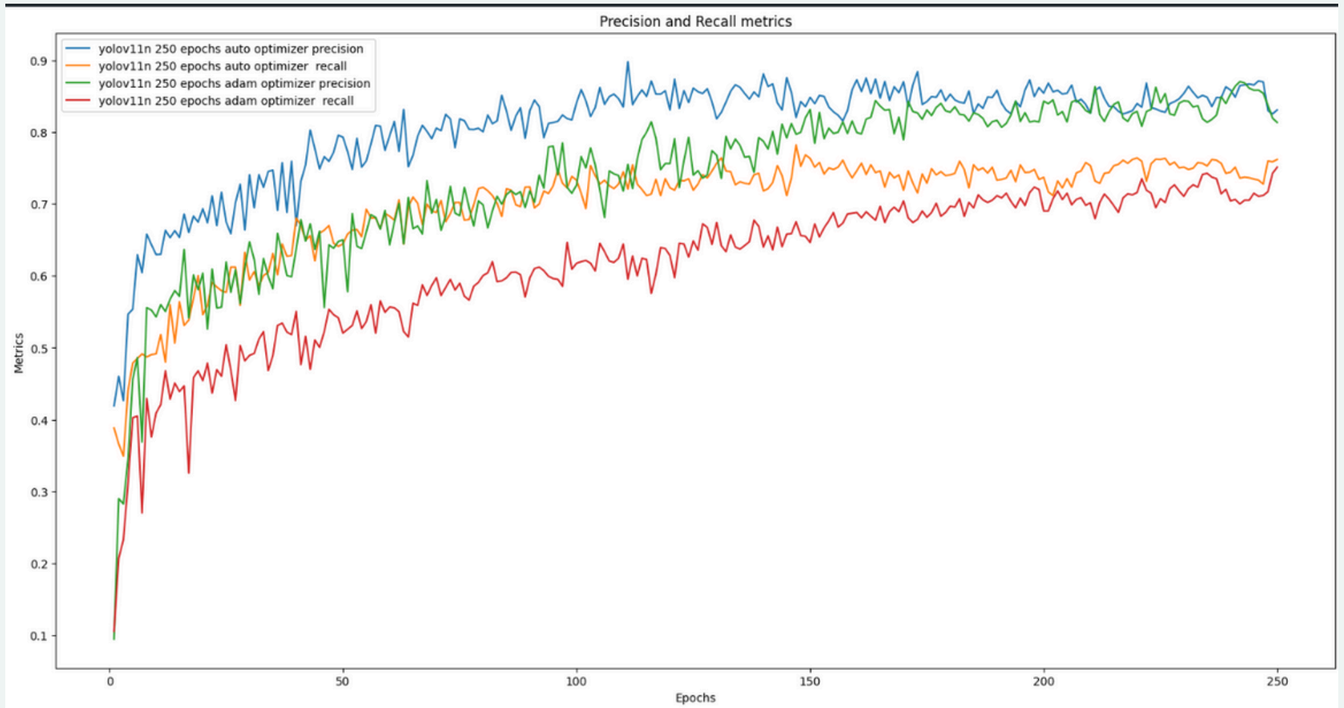
Risks underfitting

Too much epochs–

Risks overfitting



OPTIMIZER

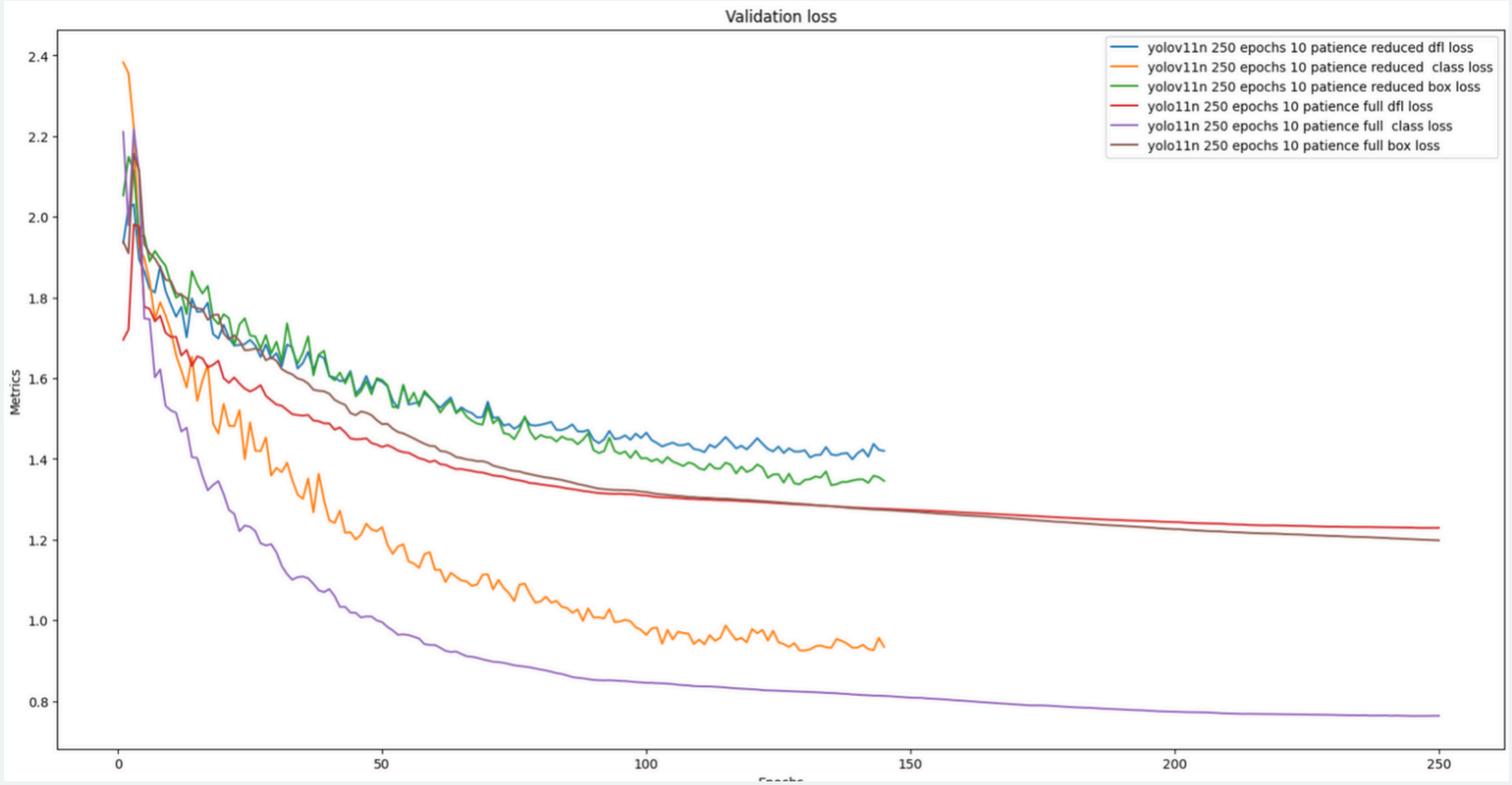


- Helps in minimizing cost function
- Stochastic Gradient Descent
- RMSprop
- Adaptative Moment Estimation
- Auto vs Adam

REDUCED VS COMPLETE DATASET

Lack of computational power → **Reduction of the dataset**

Bigger datasets tend to be better for generalization

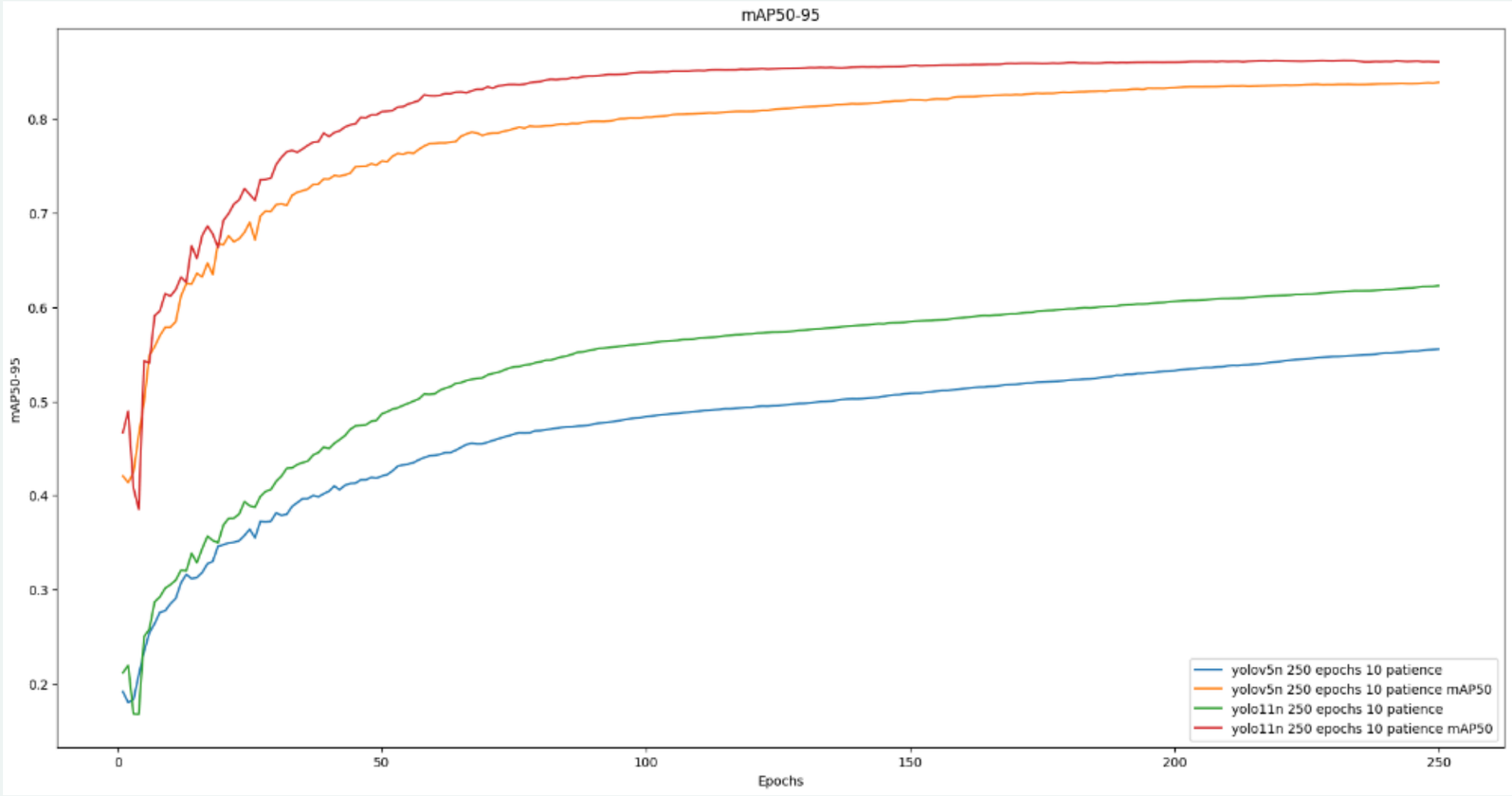


YOLO VERSIONS

Testing **YOLOv5nu** vs **YOLOv11n**

YOLOv5nu: suitable for lower power defices

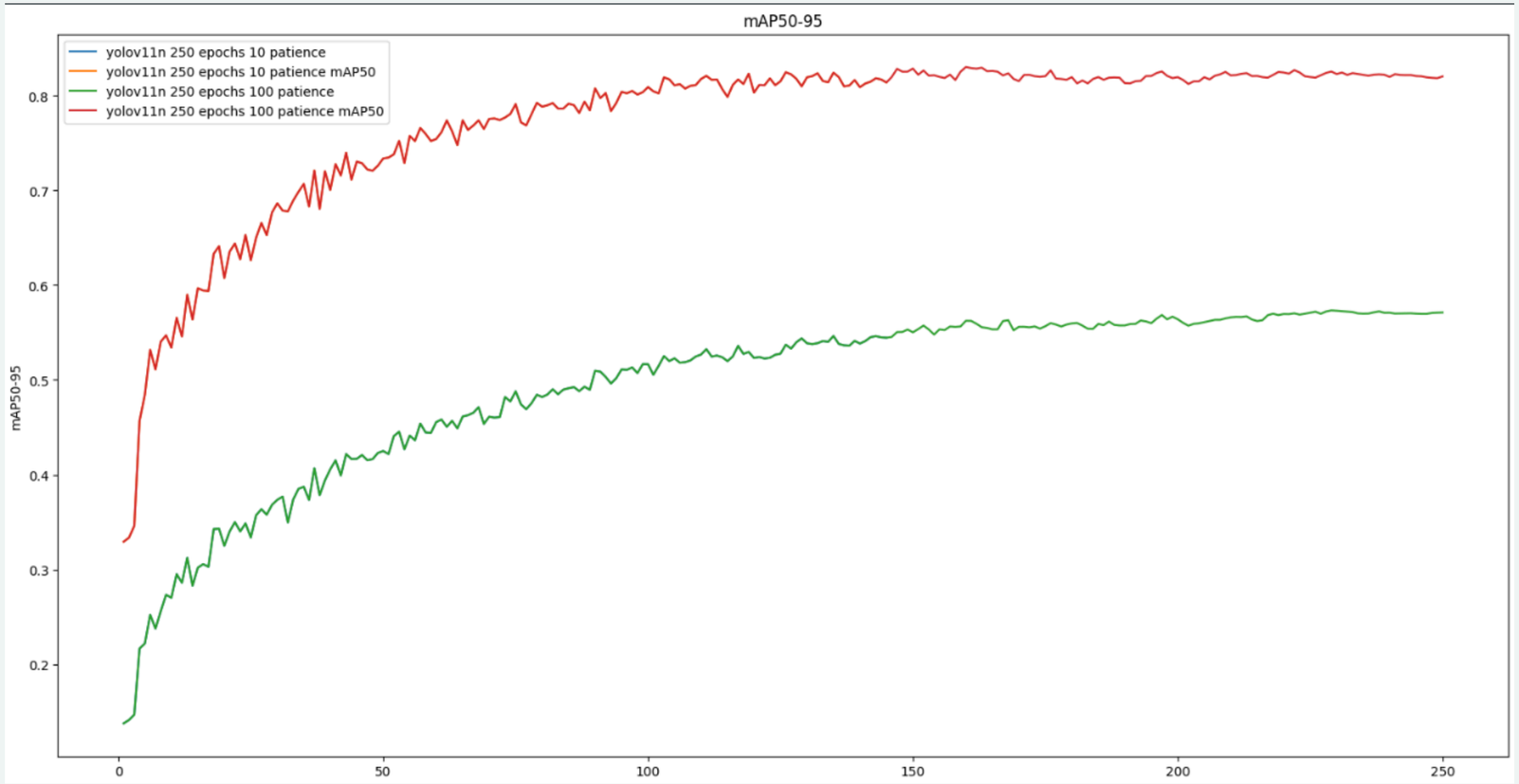
YOLOv11n: complex scenarios (crowds, small objects, high aspect ratio)



YOLOv11n surpassed in every metric

PATIENCE

How many epochs to train without improvement before stopping **“Early Stopping”**



YOLOv11n 10 patience vs 100 patience

- Very similar metrics, but 10 patience model stopped at 146 epochs
- Similar results with less computational power

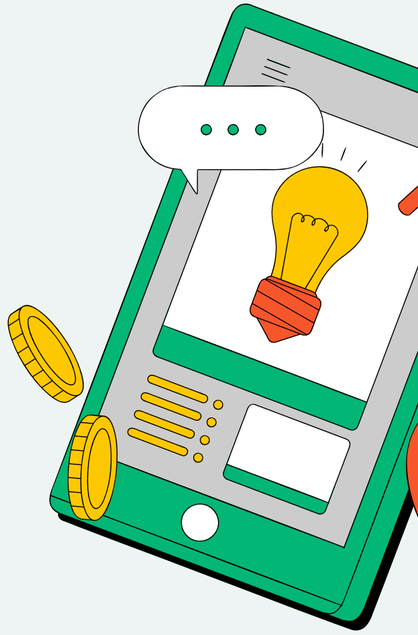
RESULTS

Compare w/ SOA

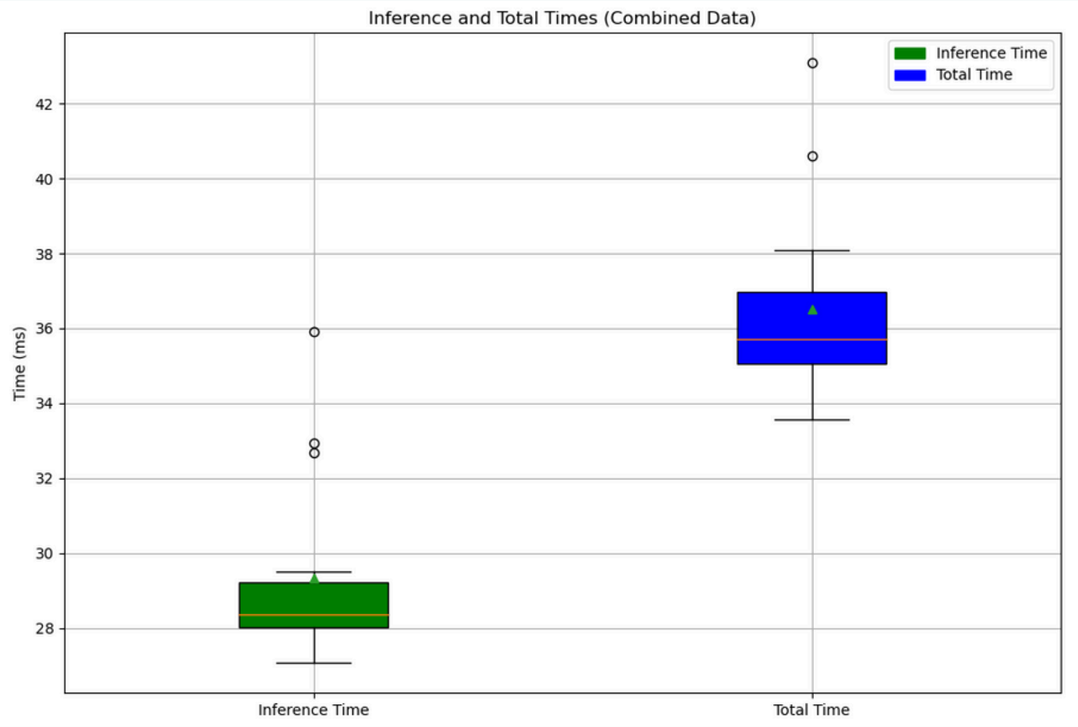
Processing/ Inference time

Compare Predictions w/ Labels

Test on a video



PROCESSING/ INFERENCE TIME



Using **YOLOv11n** for **250** Epochs

Patience of **10** and **16** Batch Size

Running in an **Nvidia Jetson**

Processing: **36.50** ms ± **1.32** ms

Inference: **29.33** ms ± **1.22** ms

Can process **27** Frames Per Second

PREDICTION VS. LABELS

Green boxes are **Labels**

Blue boxes are **predictions**

Some **misses,** but **accurate** boxes

VIDEO TEST

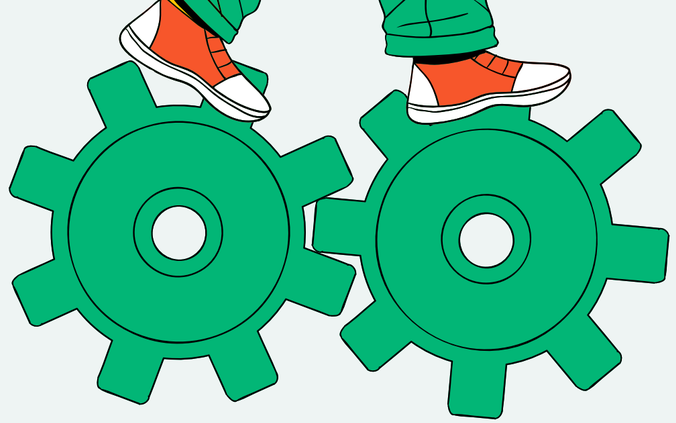
COMPARISON

Our approach with YOLO v11n significantly outperforms previous methods. By training on a diverse and challenging dataset, we achieved:

- F1-confidence of 84%
- Precision of 92.4%
- Recall-confidence of 92%
- mAP@0.5 of 84%.

Authors	Method	Results
Koch and Brilakis	Traditional Computer Vision	Precision: 82% Recall: 86%
Tedeschi and Benedetto	Machine Learning	Precision: 70% Recall: 70% F1-score: 70%
Buza	Traditional Computer Vision	mAP@0.5: 81%
Lokeshwor	Machine Learning	Precision: 95% Recall: 81%
Stpete_ishii	YOLO-NAS	Precision@50: 95% Recall@50: 95.77% F1@50: 1.89% mAP@50: 37.56%
Mir Tahmid	Compact Convolutional Transformers	Precision: 90%

FUTURE WORK



ADDITION 1

Use segmentation an try to estimate the size of the pothole, as to only notify of potentially dangerous ones

ADDITION 2

Use more classes, classifying the potholes based on size

ADDITION 3

Increase the number of epochs of the final model to 600 or 1200 epochs, since it didn't overfit, some gains may have been left out

ADDITION 4

Experiment with other hyper-parameters, such as learning rate, momentum, warm up other optimizers (especially SGD)

THE END
+ QUESTIONS