Object Detection and Tracking: Camera-based 2D Object Detection (YOLOX)

Lee Xuan Le, Chloe 22410022
Valen Tang Wenwen 22410023
黄翔 12110426
钟靖宸 12110320

CONTENT

01

Background & Significance

What is Object Detection?

- **Purpose:** Identifies and classifies object positions in images
- **Driving force:** Powered by deep convolutional neural networks (CNNs) for real-time feature extraction

Popular Models

- **YOLO:** You Only Look Once
- **R-CNN:** Region-based Convolutional Neural Networks

Applications of Object Detection

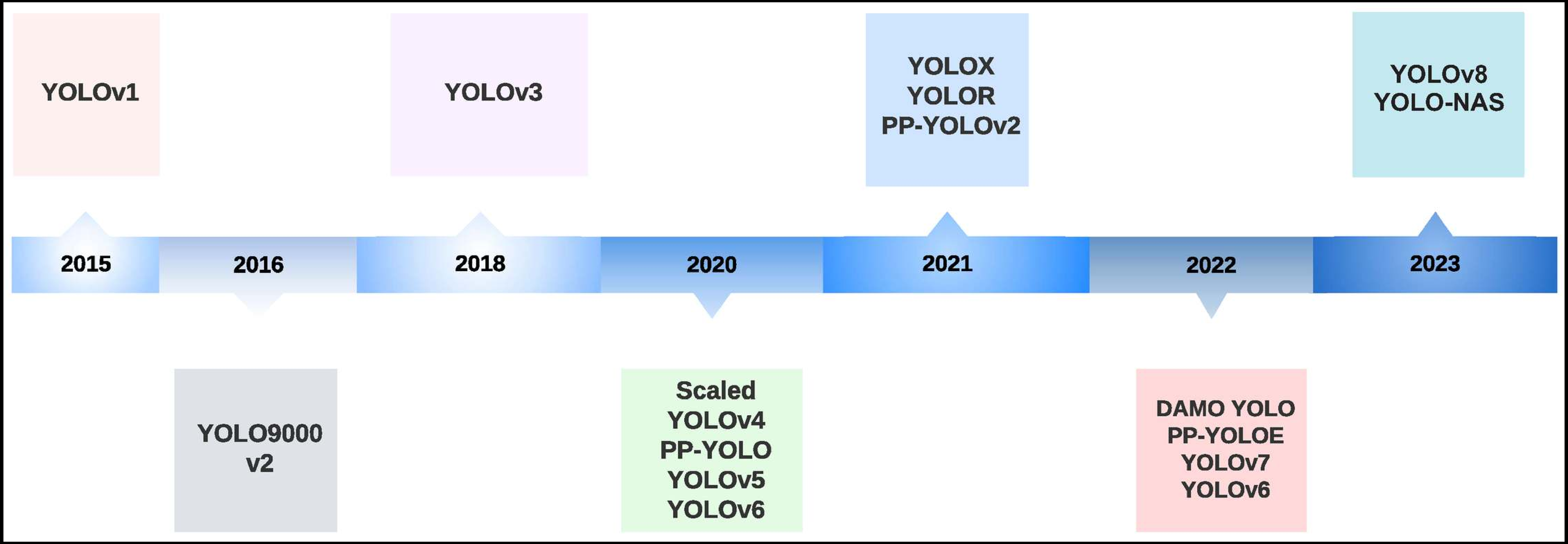
- Robotics and automation
- Self-driving cars
- Facial recognition systems

Challenges & Ethical Concerns

- **Potential misuse:** Deepfake technology
  - Misleading content
  - Non-consensual videos
- **Threats:** Privacy and security issue

YOLOX

Origins of YOLOX



YOLOX Features

- Architecture
    - Backbone: Extract features from the input image
    - Neck: Combines features from multiple scales in the backbone
    - Head: Utilises the extracted features to conduct predictions and classification
- Anchor free-design
- SimOTA label assignment
- Decoupled head

YOLOX Feature: Anchor free-design

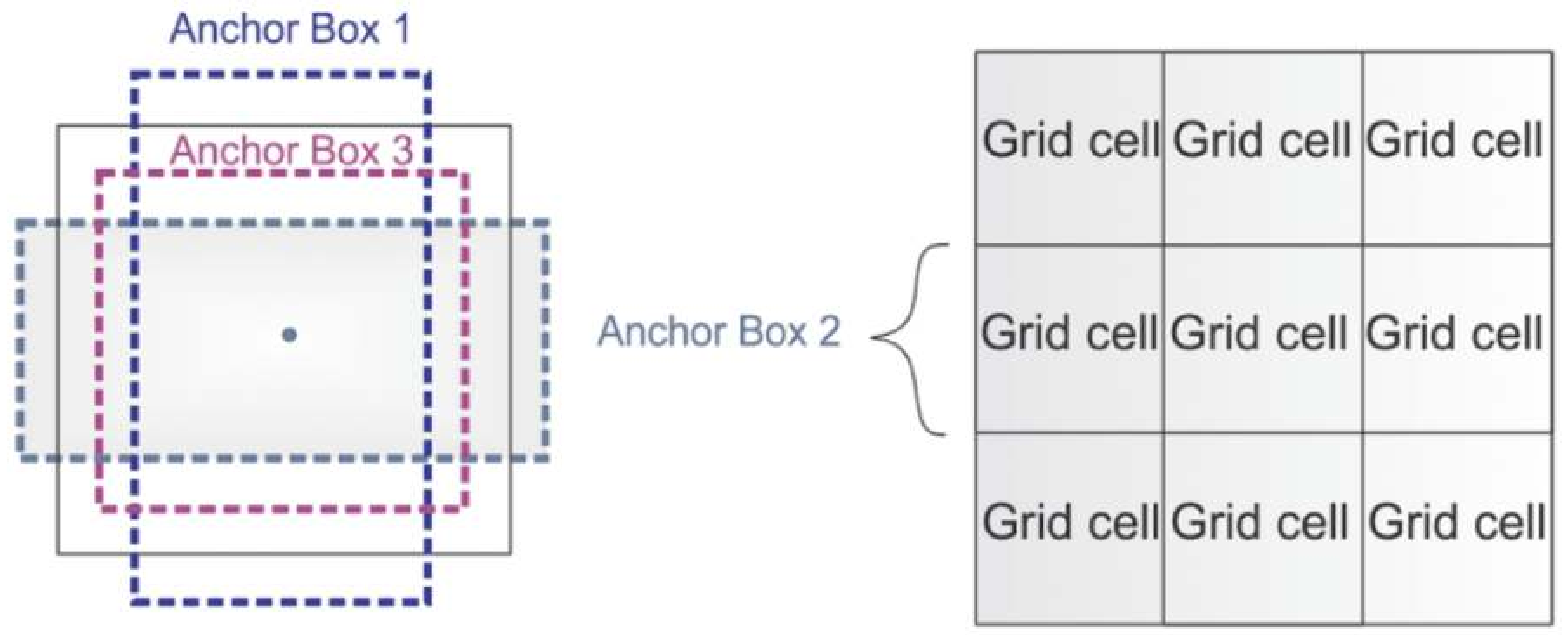


Figure 2.3: Anchors used in YOLO (Boesch, 2024)

Benefits

- **Greater flexibility:** Better adapt to objects of different shapes and sizes
- **Greater efficiency:** Reduced required predictions
- **Enhanced overall processing speed**

YOLOX Feature: SimOTA Label Assignment

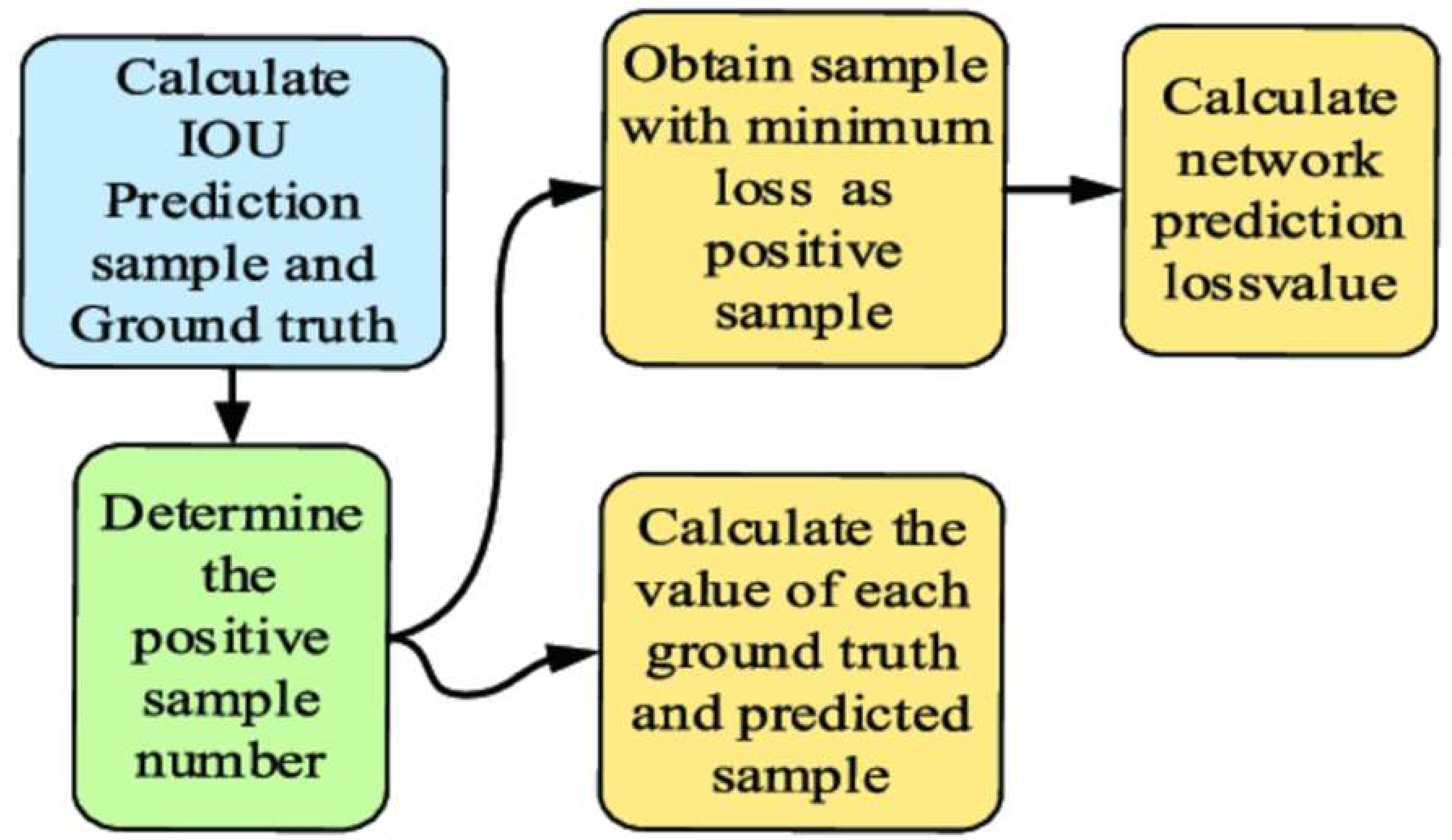


Figure 2.4: The computational process of SimOTA (Hu et al., 2022)

Benefits

- Simplified version as compared to OTA
- Requires only a single iteration to approximate the assignments
- Reduced training time

YOLOX Feature: Decoupled Head

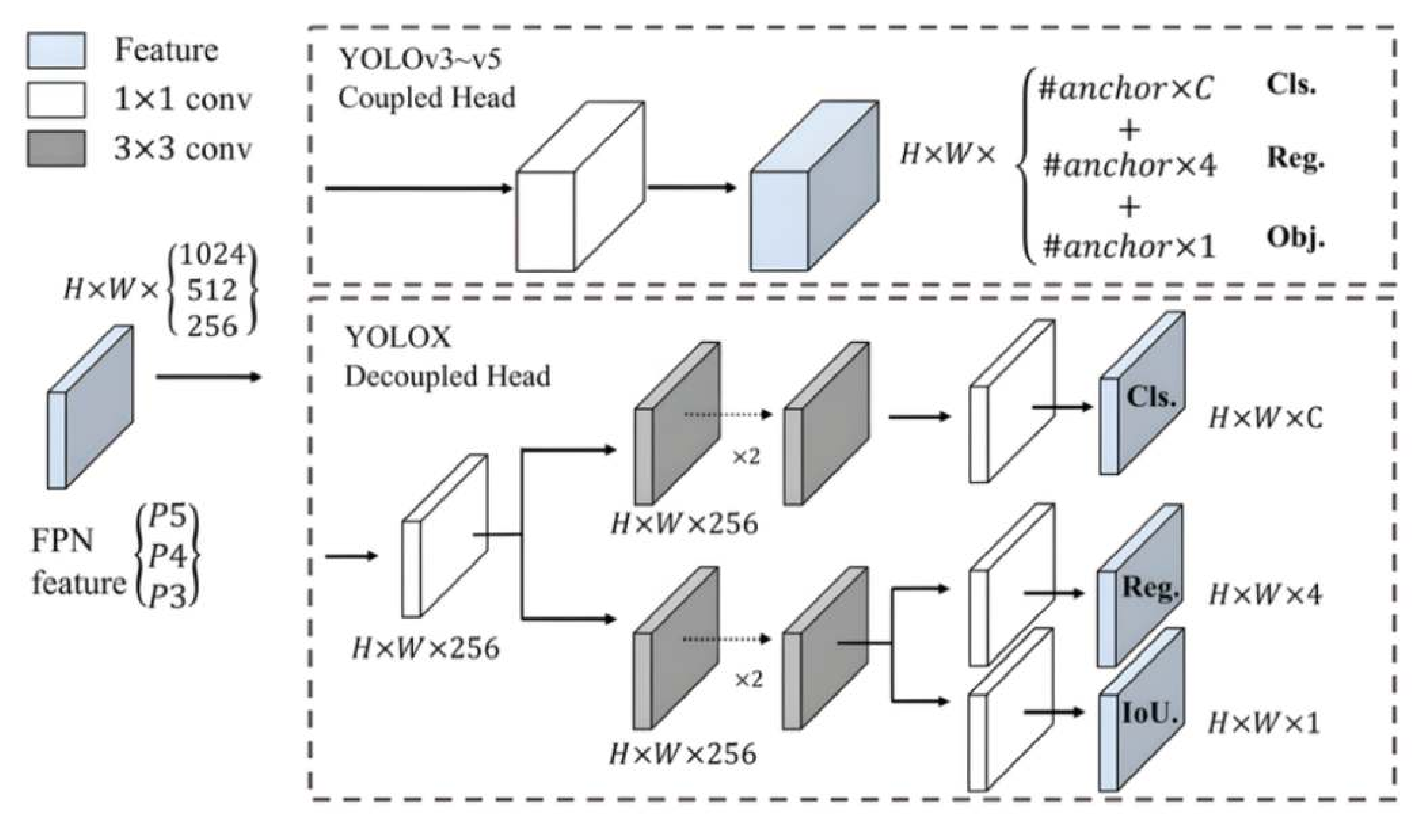


Figure 2.5: Decoupled Head Structure of YOLOX (Zhang et al., 2023)

Benefits

- Effectively target classification and regression tasks
- Separates the tasks for optimality
- Utilise fewer parameters
- Decreasing its computational expenses and overhead

Current Research Status

Case Study: Agricultural Industry

Agricultral Use: Weed Detection Model

- **Purpose:** Identify and classify crops, pests and diseases
- **Key features:**
  - **Deep network connection lightweight attention mechanism:** Recognise the different types of weeds
  - **Deconvolution layer:** Within the residual module
  - **Generalised Intersection over Union (GIoU):** Boost learning capability
- **Results**
  - **Average detection accuracy:** Almost 95% in the performance assessment,
  - **F1 value:** 0.07% improvement
  - **AP value:** 1.16% improvement
  - **Detection rate:** >88% for the detection rate of maize seedlings and weed recognition

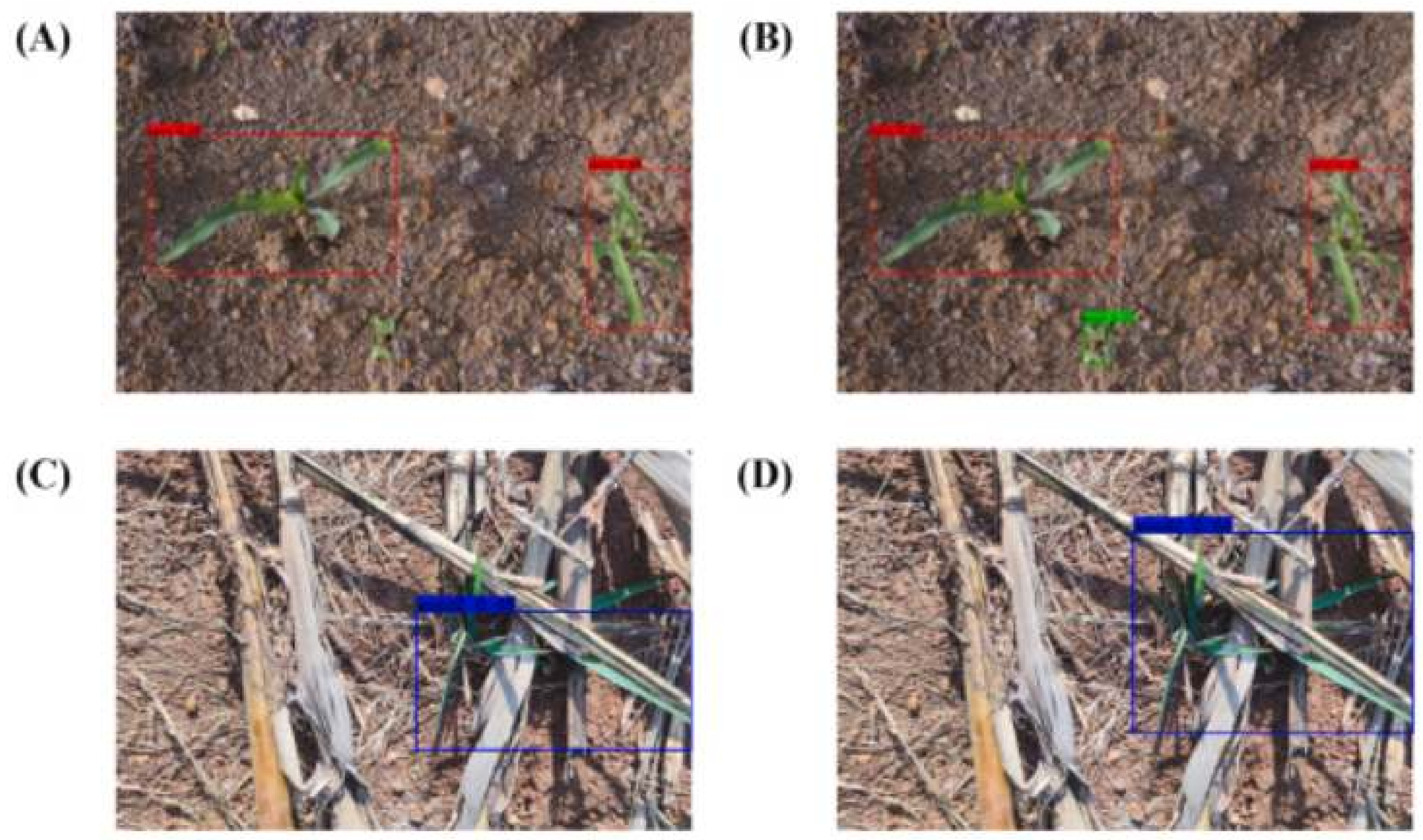


Figure 2.7: Side-by-side Comparison of Original YOLOX versus Improved YOLOX Detection Results (Zhu et al., 2024)

Case Study: Algal Bioprocessing

Chemical Use: Algae Detection Model

- **Purpose:** Identify and classify mixed algae species
- **Key features:**
  - **Starting model input:** Multi-level and morphology microalgae image dataset
  - **Focal loss:** Solve the quantity disparity challenge
  - **DIoU loss:** For regression loss, shortened processing time
- **Results**
  - **Precision:** >95%
  - **Recall:** >93%
  - **mAP value:** 3.33% improvement compared to earlier YOLOX-s
- **Learning points**
  - **YOLOX paired with other existing tools:** Could improve results
  - **Research opportunities:** Many combinations of tools

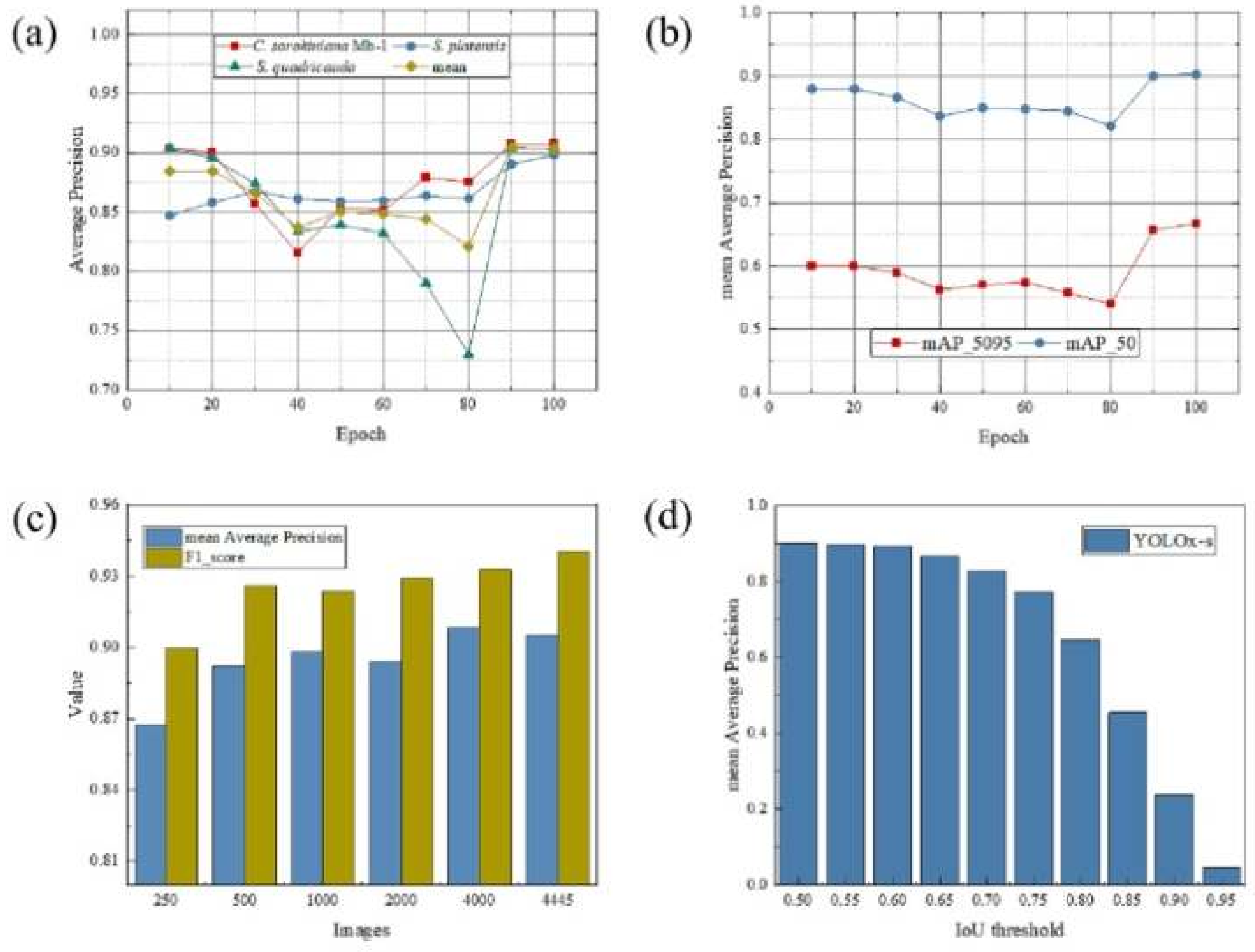


Figure 2.10: Graphical Results of YOLOX-s (Yan et al., 2023)

Research challenges

- **Limited effectiveness:** Dependency on dataset characteristics
- **Time-consuming:** Requires appropriate training data for new scenarios
- **Challenges with object detection:** Struggles with very small or overly large objects
- **Limited comparative analysis:** Lack of benchmarking against other models
- **Narrow focus:** Does not account for industry standards, economic dynamics and regulatory problems

03 Completed Work

SimOTA equation

$$c_{ij} = L_{cls}^{ij} + \lambda L_{reg}^{ij}$$

- **cij:** Total cost of matching a ground-truth object (gi) with a predicted prediction (pj)
- **Lclsij:** Measures the error in predicting the class of the ground-truth object (gi)
- **Lregij:** Measures the error in predicting the location and size of the ground-truth object (gi)
- **λ:** Balancing coefficient, controls the relative importance of classification (Lclsij) and regression loss (Lregij)

Aim:

1. Calculate the cost (cij).
2. Select the top k predictions (pj) with the lowest costs for each ground-truth object (gi).

Outcome:

- Find the **best match** between predictions (pj) and ground-truth objects (gi)
- Ensure accurate label assignment for training.

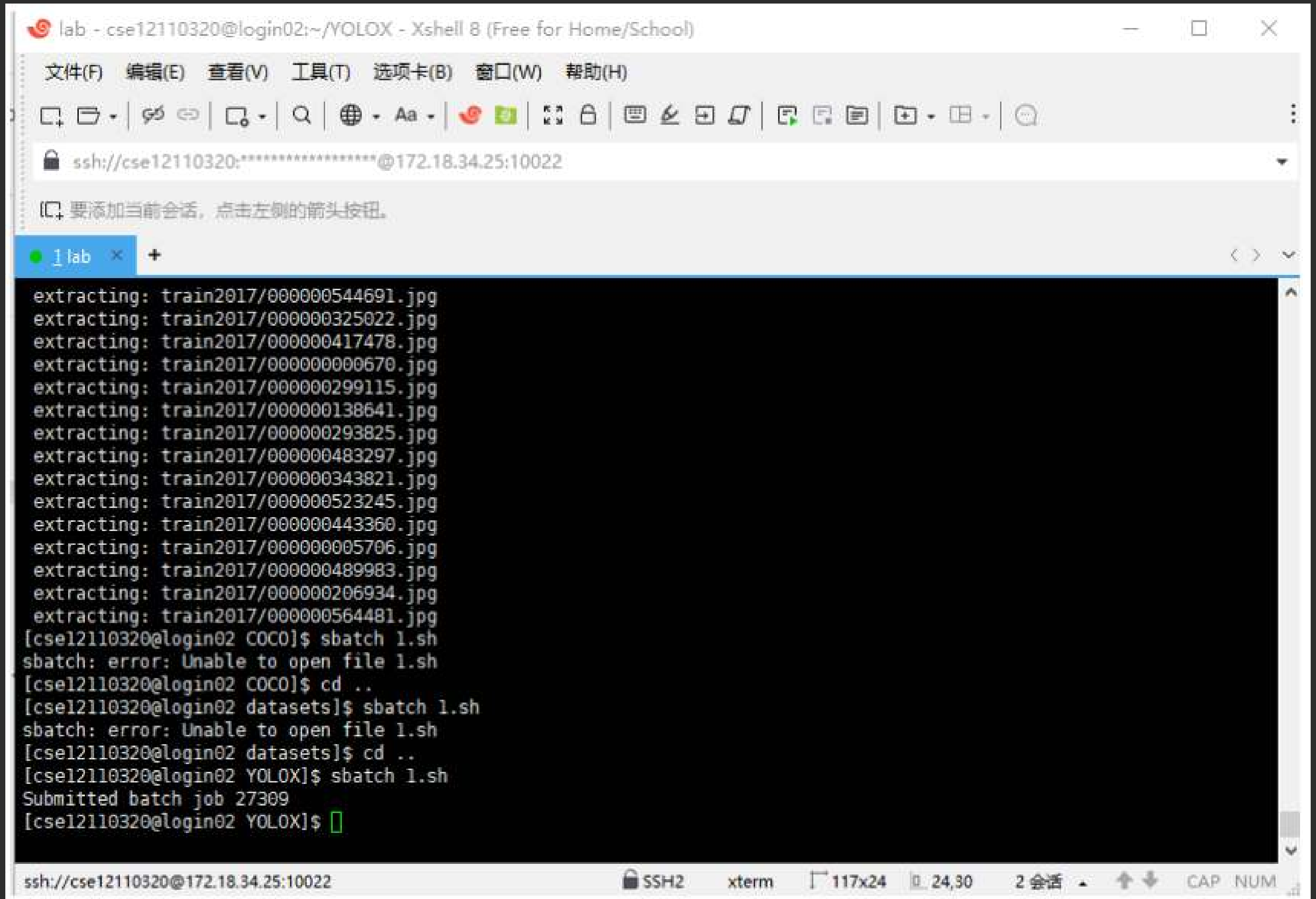
Analysis of Baseline Results

Models	AP (%)	Parameters	GFLOPs
YOLOv4-Tiny [30]	21.7	6.06 M	6.96
PPYOLO-Tiny	22.7	4.20 M	-
YOLOX-Tiny	**32.8** (+10.1)	5.06 M	6.45
NanoDet[3]	23.5	0.95 M	1.20
YOLOX-Nano	**25.3** (+1.8)	0.91 M	1.08

- **YOLOX-Tiny has an AP of 32.8%:** Much higher than its counterparts, implies better accuracy
- **YOLOX-Nano has an AP of 25.3%:** Achieves balance between efficiency and accuracy
- YOLOX-Tiny has **moderately sized parameters (5.06M)** vs YOLOX-Nano has extremely **lightweight parameters (0.91M)**, indicating lower model complexity.
- **Computational cost:** YOLOX-Tiny has GFLOPs of **6.45**, which is relatively **efficient** for its accuracy, while YOLOX-Nano has GFLOPs of **1.08**, making it much more **efficient** but with slightly **lower accuracy.**

Deployment Results

- **Dataset:** COCO 2017 dataset to run a YOLOX-s model
- **Step 1:** Connect to the server and run the model



Deployment Results

- **Step 2: Execute comments**

1 [6].sh - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
#!/bin/bash
#SBATCH -o job.%j.out          # 脚本执行的输出将被保存在当job.%j.out文件下，%j表示作业号;
#SBATCH --partition=titan       # 作业提交的指定分区队列为titan
#SBATCH --qos=titan            # 指定作业的QOS
#SBATCH -J myFirstGPUJob        # 作业在调度系统中的作业名为myFirstJob;
#SBATCH --nodes=1              # 申请节点数为1,如果作业不能跨节点(MPI)运行, 申请的节点数应不超过1
#SBATCH --ntasks-per-node=6     # 每个节点上运行一个任务，默认一情况下也可理解为每个节点使用一个核心;
#SBATCH --gres=gpu:1           # 指定作业的需要的GPU卡数量，集群不一样，注意最大限制;


python -m yolox.tools.train -n yolox-s -d 1 -b 64 --fp16 -o
```

第 1 行，第 1 列 | 100% | Unix (LF) | UTF-8

Deployment Results

- **Step 3: Obtain results**

04 Research Plan & Expected Results

Week	**Task**	**Expected outcomes**
12	• Topic selection • Identify the purpose and direction of the project • Conduct secondary research on the current problem	• Conduct a thorough literature review of YOLOX to identify existing gaps and establish a baseline understanding of the problem. • Identify the advantages and limitations of YOLOX in object detection.
13	• Construct dataset • Train the results • Complete and submit report and ppt	• Gather a diverse dataset for testing, ensuring the dataset includes challenging scenarios like small objects. • Train the YOLOX model on the dataset using existing architecture as a baseline to establish starting performance metrics.

Week	**Task**	**Expected outcomes**
14	• Improve the accuracy of working code • Dataset refinement	• Implement proposed enhancements, such as fine-tuning hyperparameters to improve detection accuracy. • Clean and label datasets, ensuring proper annotations for bounding boxes and object identities.
15	• Collect and analyse results • Visualise possible trends • Further refinement of code	• Evaluate the improved model on validation datasets, analyse performance metrics, and compare with baseline results to measure progress. • Generate visualisations of key trends to identify remaining issues and areas for further refinement. • Enhance code efficiency by reducing unnecessary processing and improving how the system handles tasks in real time.

Week	**Task**	**Expected outcomes**
16	• Summary of key results • Presentation of results	• Summarise the research findings, highlighting the performance gains, improvements made and any unresolved challenges. • Present the final outcomes, including comparisons to the baseline and implications for future work.

15

Potential Challenges & Solutions

Challenge 1

Dataset lacks diversity

Solutions

- Apply developed **data augmentation techniques** (e.g. brightness adjustments) to simulate different conditions
- **Synthetic data** could be produced to generate scenarios that are less or underrepresented in the real dataset
- Additional **real-world datasets** can be collected to improve representation

Challenge 2

Training YOLOX on large training datasets presents significant computational challenges

Solutions

- Leverage **cloud platforms** (e.g. AWS) to access high-performance computing resources
- Start with a **small dataset** initially and gradually increase the sample size
- Implementation of **model pruning** to reduce unnecessary parameters
- **Mixed-precision training**, using lower-precision data types without sacrificing model performance

Challenge 3

Model may fail to effectively generalise to fresh data

Solutions

- **Dropout:** Randomly set a proportion of the weights to zero during training
- **Weight decay:** Adds a penalty to the loss function, discouraging excessively or overly large weights
- **Label smoothing:** Softening the target labels during training
- **Class imbalance:** Mitigated through oversampling underrepresented classes or using weighted loss functions, higher weights are assigned to less frequent classes

Thank you!