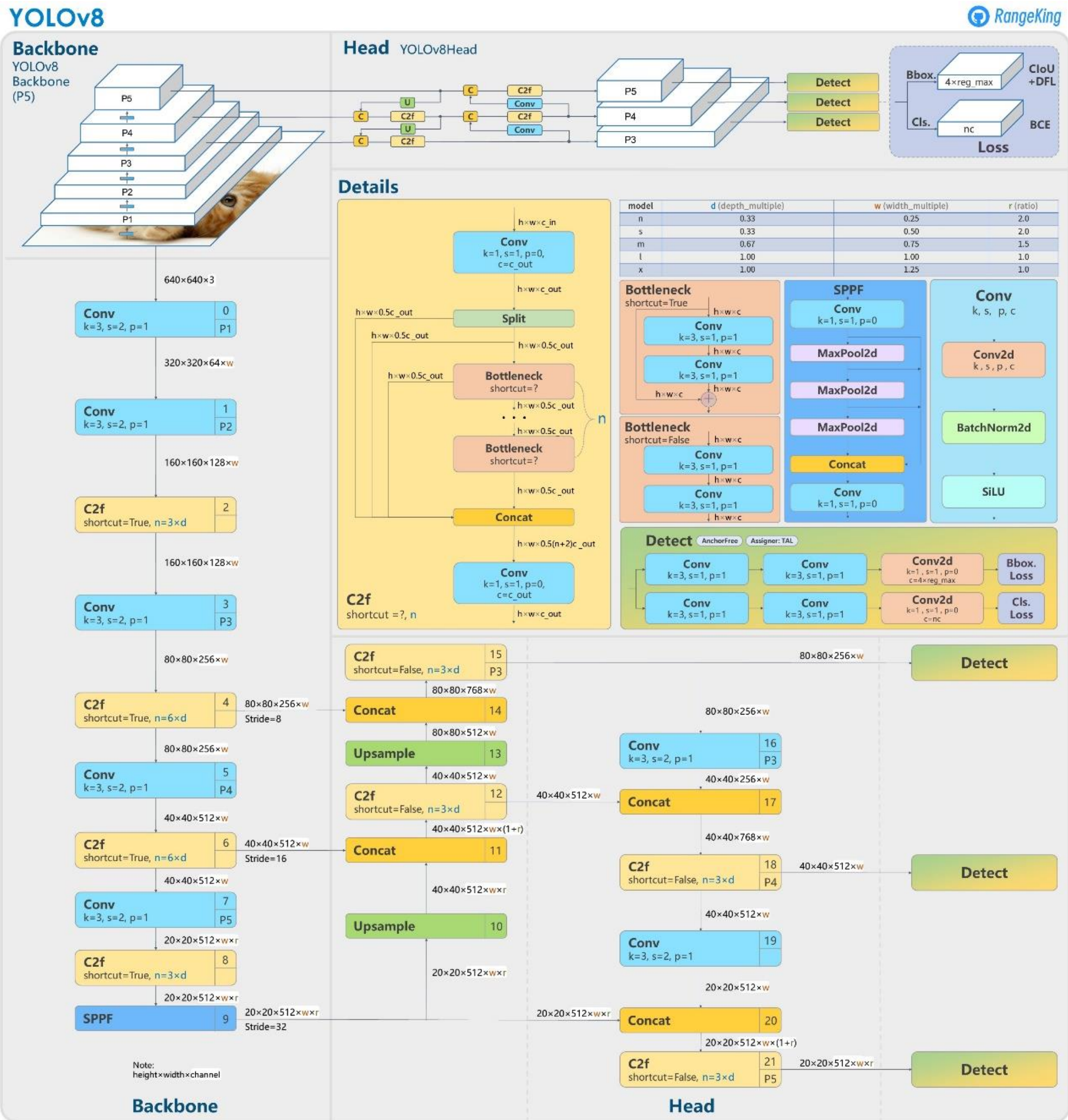
HW2-Long-Tailed Object Detection

1. Model Description

➢ Architecture

- According to paper [1], the object detection model for small objects performs well on yolov8, so this assignment uses the yolov8 model.



YOLOv8 Architecture, visualisation made by GitHub user RangeKing

2. Implementation Details

➢ Preprocessing

- Remove data where the BoundingBox width or height equals 0.

- Convert to YOLO format.
- Remove images without ground truth.
- Based on the number of images corresponding to each category, the data is divided into 90% Training and 10% Validation.

➢ Hyperparameters

- Training

Parameter	Value
epochs	400
patience	30
batch	1
imgsz	1920
pretrained	False

- Prediction

Parameter	Value
conf	0.001
iou	0.7
imgsz	1920
max_det	500

➢ Augmentation

- YOLOv8 data augmentation training hyperparameters.

Parameter	Value
hsv_h	0.015
hsv_s	0.7
hsv_v	0.4
translate	0.1
scale	0.5
fliplr	0.5
mosaic	1.0
mixup	0.3

➢ Loss functions

- box_loss : The part used to optimize the difference between the predicted bounding box and the true bounding box
- cls_loss : Used to optimize the model's prediction accuracy for object categories. Classification loss ensures that the model can correctly identify which category the object in the image belongs to.

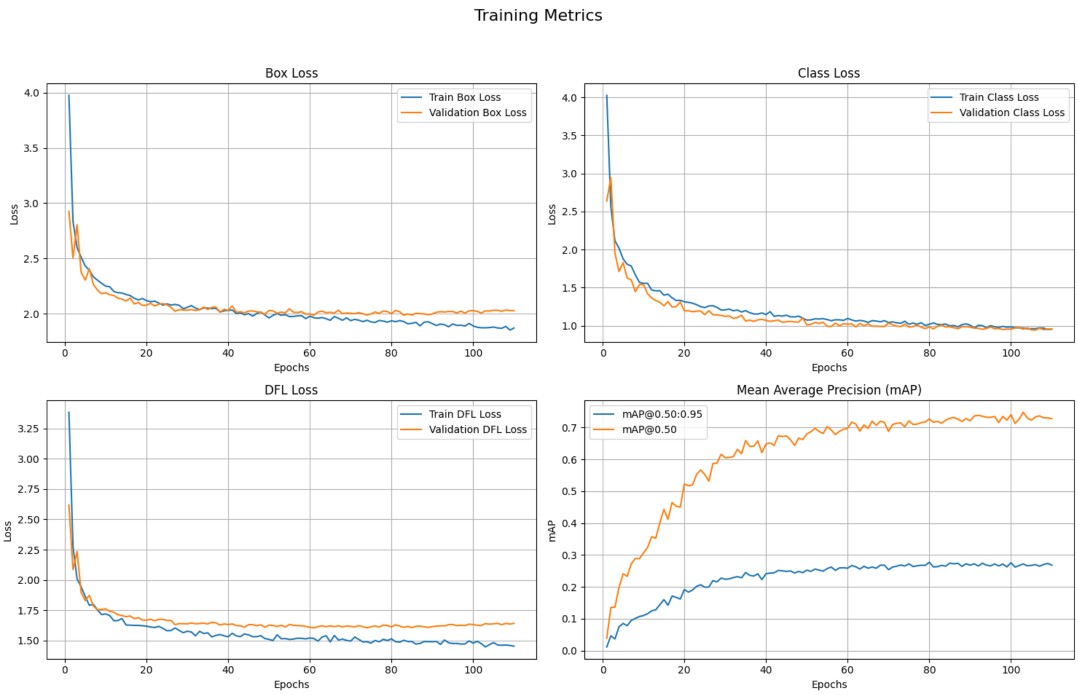
- dfl_loss : Solve the class imbalance problem in object detection and improve the performance of the model when dealing with small objects and difficult samples.

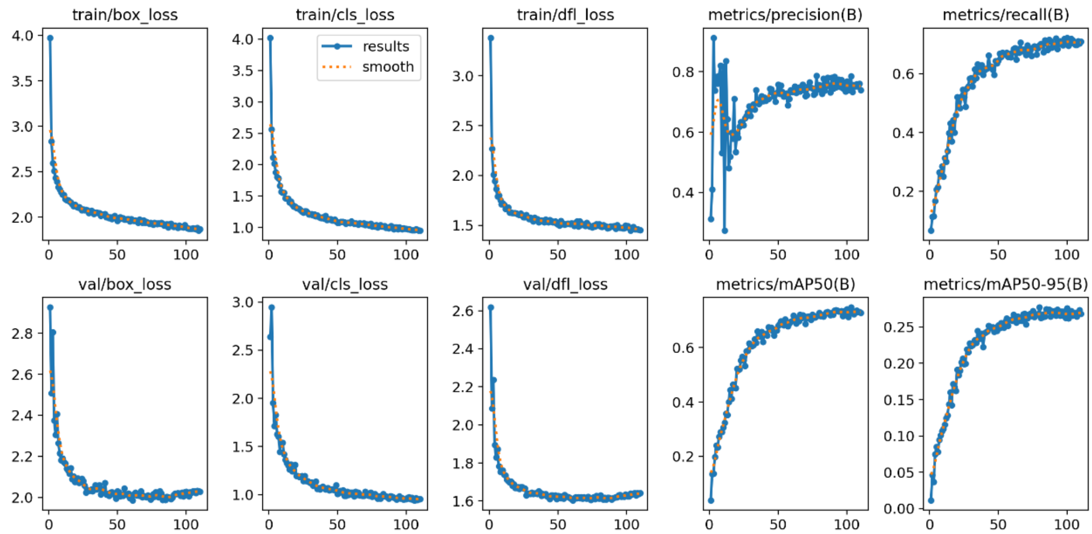
➢ Training strategies

- This assignment focuses on Long-Tailed and small object detection, therefore YOLOv8 is adopted for training. Before training, the validation dataset is split into 10% for Long-Tailed objects. During training, the input image resolution is enlarged to 1920 to better detect small objects. In addition, according to the augmentation strategy for class imbalance object detection in paper [2], the mosaic is set to 1 and the mixup is set to 0.3. With the adjustment of other augmentation parameters, the results show that the mAP@50–95 evaluation index is significantly improved.

3. Result Analysis

➢ Quantitative improvements





➢ Visualizations

- Predict results

➢ Long-Tailed Mitigation

- Regarding the Long-Tailed dataset problem, one of my approaches is to adjust the mosaic and mixup settings in data augmentation, following the experimental findings in paper [2] , which leads to improved performance.

```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 6/6 1.2it/s 5.2s
                   all         96       3359      0.743      0.731       0.73      0.275
                   car         93       2333      0.908      0.938      0.946      0.415
                   hov         59        159      0.828      0.874      0.861       0.35
                person         60        399      0.553      0.369      0.409       0.11
            motorcycle         49        468      0.681      0.742      0.703      0.224
Speed: 6.2ms preprocess, 36.9ms inference, 0.0ms loss, 0.3ms postprocess per image
Results saved to /home/m11315025/Desktop/CVPDL/hw2_M11315025/code_M11315025/src/runs/detect/val
The model's mAP@0.50:0.95 on the validation set is: 0.2748
```

Evaluation after adding mosaic and mixup

```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━ 6/6 1.2it/s 5.2s
                   all         96       3359      0.766      0.689      0.726      0.272
                   car         93       2333      0.915      0.923      0.936      0.409
                   hov         59        159      0.868      0.862      0.883      0.351
                person         60        399      0.621      0.238       0.38        0.1
            motorcycle         49        468      0.662      0.733      0.706      0.226
Speed: 6.4ms preprocess, 36.8ms inference, 0.0ms loss, 0.3ms postprocess per image
Results saved to /home/m11315025/Desktop/CVPDL/hw2_M11315025/code_M11315025/src/runs/detect/val
The model's mAP@0.50:0.95 on the validation set is: 0.2718
```

Evaluation before adding mosaic and mixup

4. Conclusion

- The focus of this assignment is on how to address the Long-Tailed problem. Some classes contain extremely few samples, which causes the model to be biased toward majority classes. In the experiments, I tested multiple data augmentation strategies and different training parameters, and also referenced methods from related literature to improve the evaluation score slightly. However, the model still suffers from overfitting, which needs to be further addressed in the future by applying more targeted techniques to improve performance. Overall, the Long-Tailed issue cannot be fully solved simply by increasing data volume or training epochs. Additional strategies are required to achieve a balanced feature distribution across classes.

5. Reference

[1] Tariq, M.F., & Javed, M.A. (2025). Small Object Detection with YOLO: A Performance Analysis Across Model Versions and Hardware. *ArXiv, abs/2504.09900*.

[2] Crasto, N. (2024). Class Imbalance in Object Detection: An Experimental Diagnosis and Study of Mitigation Strategies. *ArXiv, abs/2403.07113*.