Comparative Analysis of YOLOv11 and YOLOv12 Segmentation Models

1. Dataset Distribution

The dataset used for training consisted of 627 images and 1151 annotated instances across four damage categories.

Class	Number of Instances
Corrosion Induced Spalling	171
Crack	605
Peeling	177
Spalling	198
Total	1151

2. Training Setup

Aspect	YOLOv11-seg	YOLOv12-seg
Batch Size	32	32
Training Epochs (total before stopping)	762	796
Best Epoch (EarlyStopping)	662	696
Optimizer	SGD	SGD
Default Learning Rate	0.01	0.01
Input Image Size	640×640	640×640

3. Comparative Results

Overall Performance

Metric	YOLOv11	YOLOv12	Interpretation
Box Precision (P)	0.841	0.813	YOLOv11 more precise
Box Recall (R)	0.757	0.779	YOLOv12 slightly better recall
Box mAP@50	0.825	0.822	Nearly identical
Box mAP@50-95	0.646	0.637	YOLOv11 slightly stronger
Mask Precision (P)	0.844	0.816	YOLOv11 better
Mask Recall (R)	0.766	0.769	Very close
Mask mAP@50	0.828	0.811	YOLOv11 better
Mask mAP@50-95	0.566	0.544	YOLOv11 better

Per-Class Performance (with Instances)

Class	Instances	YOLOv11 Box mAP@50	YOLOv12 Box mAP@50	YOLOv11 Mask mAP@50	YOLOv12 Mask
Induced Spalling	171	0.934	0.945	0.922	0.918
Crack	605	0.976	0.938	0.978	0.938
Peeling	177	0.781	0.778	0.790	0.777
Spalling	198	0.611	0.628	0.631	0.611

4. Comparative Analysis

YOLOv11 consistently outperforms YOLOv12 in precision and mAP metrics, while YOLOv12 achieves slightly higher recall. Class imbalance plays a significant role: YOLOv11 performs better on the majority class (Crack), while YOLOv12 shows a slight edge on Spalling. Despite YOLOv12 having a deeper architecture, it does not outperform YOLOv11 on this dataset, indicating YOLOv11's efficiency.

5. Conclusion

YOLOv11 is the better choice for this dataset, providing higher precision and stronger mAP values. YOLOv12 shows marginally better recall and better handling of Spalling but underperforms on the dominant Crack class. For deployment, YOLOv11 is recommended as it balances accuracy and efficiency. Future improvements should focus on data augmentation for minority classes and experimenting with optimizers like AdamW.