Enhancing Railway Safety: A Machine Learning Approach for Automated Detection of Missing Track Bolts

Aayush Damai, Hwapyeong Song, Husnu S. Narman, Ph.D., Alexander Lambert, and Ammar Alzarrad, Ph.D

ASCE International Conference on Computing in Civil Engineering (i3CE 2025)
May 13, 2025

Outline

- Introduction
- Methodology
- Results
- Conclusion

Railroad Maintenance

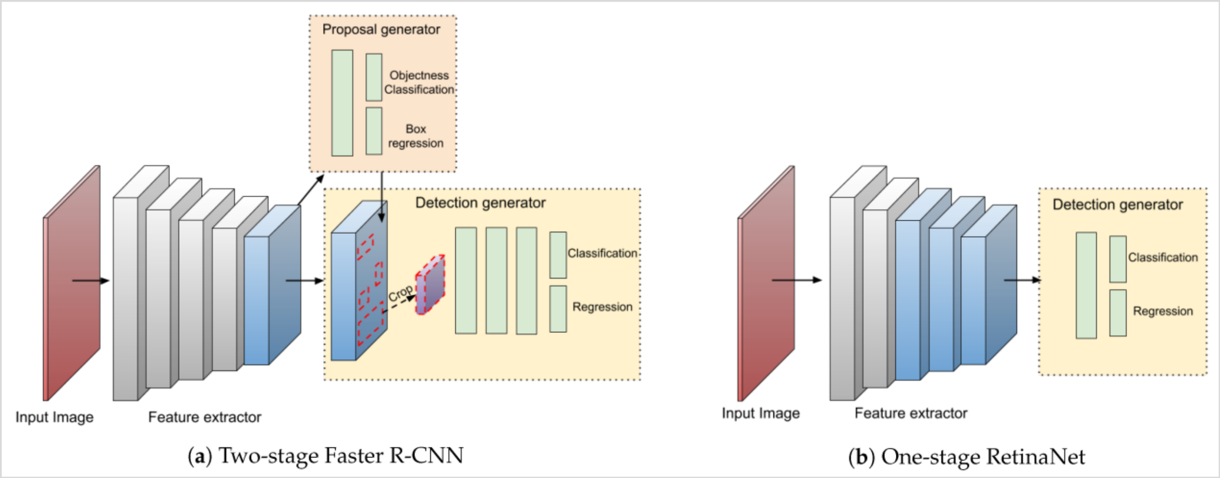
- Constant stress from heavy usage leads to wear and tear
- Loose or missing bolts can cause train derailments
- Current inspection methods are often slow and inefficient
- Require personnel to walk along tracks in dangerous weather, posing safety risks

Automated Detection of Track Bolts Using Object Detection

- Computer vision technique to detect objects in images and videos
- Trained on large datasets for accurate recognition
- Automatically detects missing track bolts
- Enhances rail safety for both freight and passenger transport

You Only Look Once (YOLO)

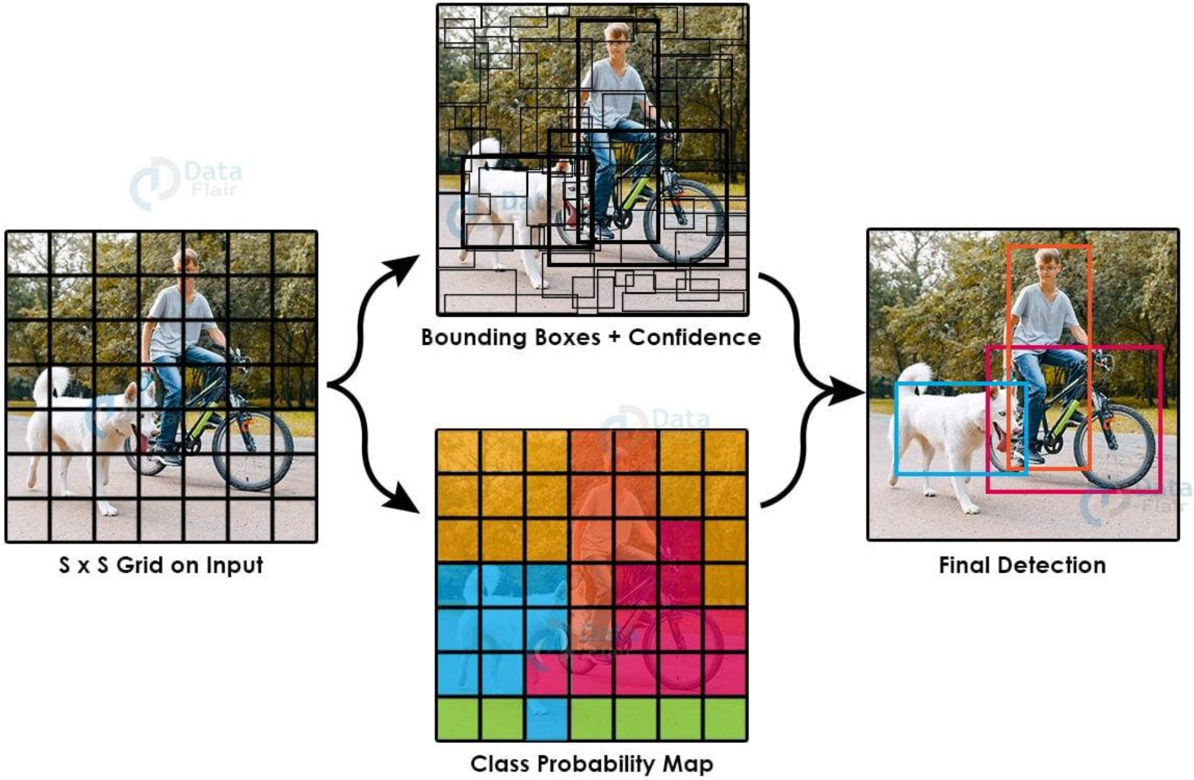
[1]



- YOLO is a real-time object detection model using CNNs.
- Unlike two-stage models, it detects objects in one pass.
- This study focuses on YOLOv5, v8, v9, and v10.

You Only Look Once (YOLO)

[2]



YOLO Sizes

- YOLO has various sub-versions optimized for different hardware.
- Smaller models are faster and suitable for low-power devices, but with reduced accuracy.
- Larger models deliver higher accuracy but require more processing power.

[3]

Model	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)
YOLOv8n	80.4	0.99	8.7
YOLOv8s	128.4	1.20	28.6
YOLOv8m	234.7	1.83	25.9
YOLOv8l	375.2	2.39	43.7
YOLOv8x	479.1	3.53	68.2

- Choosing the right version depends on the trade-off between speed and accuracy.

Model Comparisons – YOLOv5

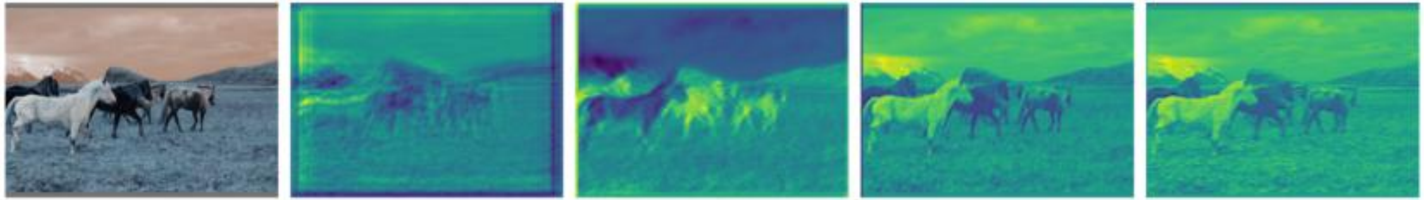
- YOLOv5 introduced major architectural improvements, including CSP-Darknet53 as its backbone. [8]
- YOLOv5u, an updated version, includes anchor-free detection features from YOLOv8.
- Anchor-free split head improves bounding box prediction.
- This study uses the YOLOv5u version for all tests.

Model Comparisons – YOLOv8

- YOLOv8 introduces anchor-free and objectness-free split-head architecture. [9]
- Enables faster, more efficient predictions with minimal accuracy loss.
- Chosen for this research due to anchor-free detection, like YOLOv5u.
- Features an enhanced backbone and neck, improving feature extraction.

Model Comparisons – YOLOv9

[10]



(a) Input Image (b) PlainNet (c) ResNet (d) CSPNet (e) GELAN

- YOLOv9 introduces two key innovations: PGI and GELAN.
  - PGI (Programmable Gradient Information) reduces data loss as layers increase.
  - GELAN (Generalized Efficient Layer Aggregation Network) combines multi-level features using parallel and efficient connections to improve speed and accuracy without extra complexity.
- Preserves image structure and improves computational efficiency.
- Solves a limitation found in previous YOLO versions.

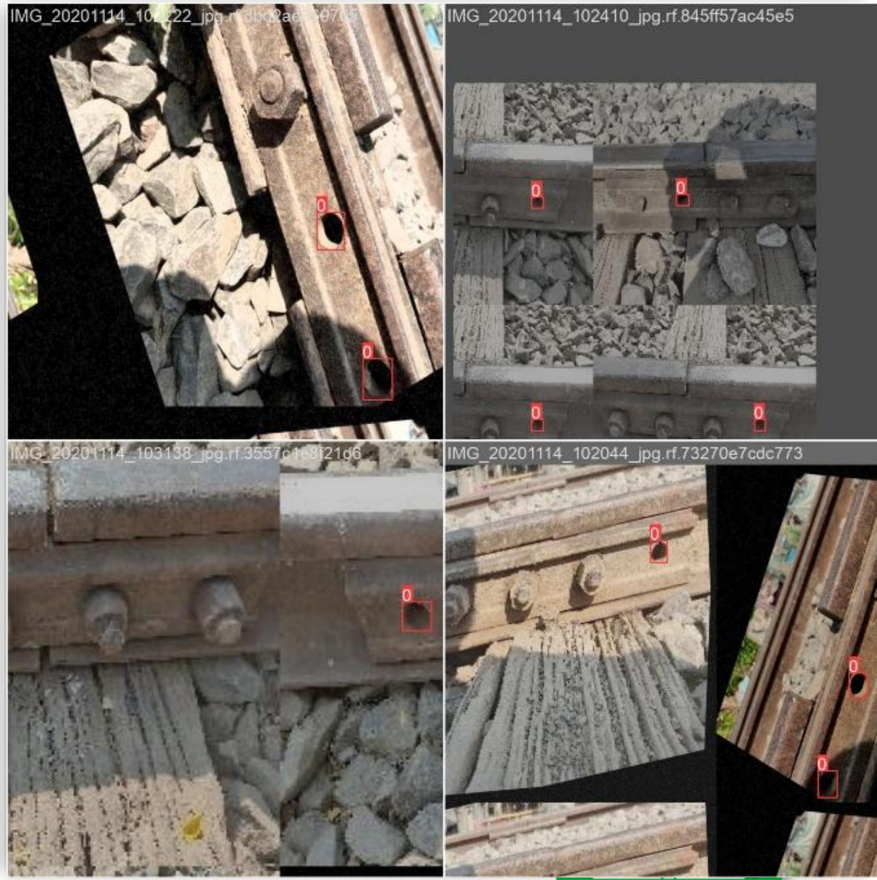
Model Comparisons – YOLOv10

- YOLOv10 uses an updated CSPNet backbone for better gradient flow and speed. [11]
- Introduces NMS-free detection for lower latency.
- Includes a lightweight classification head to reduce redundant computation.
- Chosen for its high computational efficiency, though less effective for small objects.

Dataset

[4-7]

- High-quality, diverse dataset is essential for accurate model training.
- Images differ in bolt location and orientation to improve robustness
- Sourced from Roboflow and Kaggle, then labeled and augmented
- 152 for training, 39 for testing, and 73 for validation (Total 264 images)
- Dataset contains no True Negatives (TN).



Evaluation Metrics for Model Performance

- Precision

  The proportion of correct positive predictions out of all predicted positives

- **Recall**

  The proportion of actual positives correctly identified by the model

- F1 Score

  The harmonic mean of precision and recall, balancing both metrics

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + \boldsymbol{FN}}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

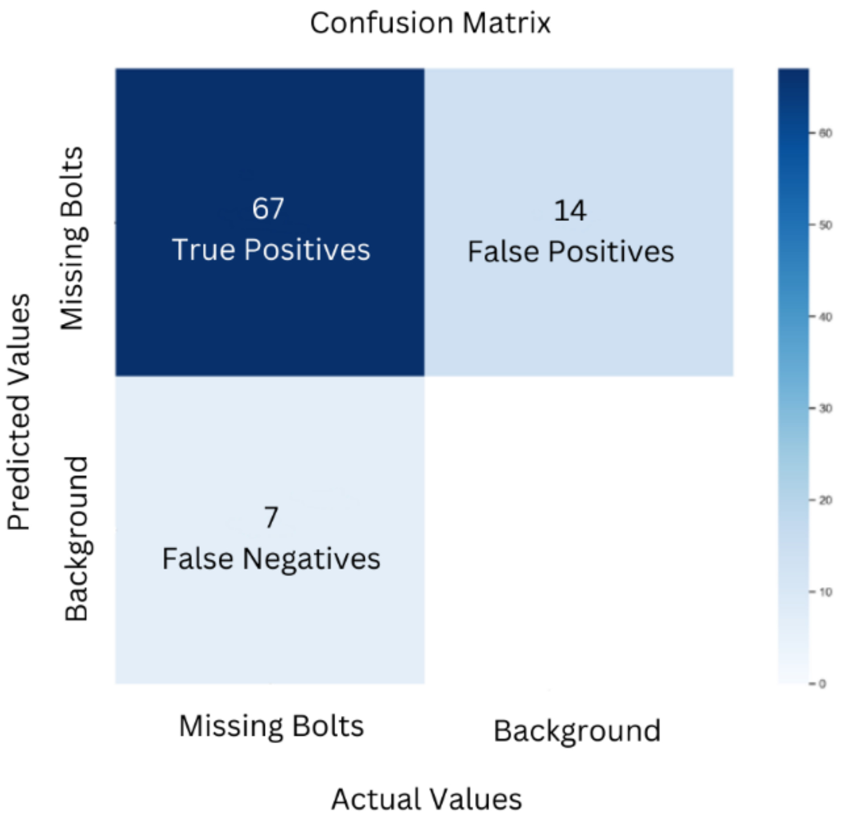
YOLOv5u Model Results

- YOLOv5lu showed the best overall performance, with the highest F1 and recall.
- Recall was prioritized to minimize False Negatives in detecting missing bolts.
- Precision ranked fourth, but recall was more critical for this task.

Model	Precision	Recall	F1
YOLOv5nu	0.7848	0.8378	0.8105
YOLOv5n6u	0.8378	0.8378	0.8378
YOLOv5su	**0.8472**	0.8243	0.8356
YOLOv5s6u	0.8133	0.8243	0.8188
YOLOv5mu	0.7750	0.8378	0.8052
YOLOv5m6u	0.7738	0.8784	0.8228
YOLOv5lu	0.8272	**0.9054**	**0.8645**
YOLOv5l6u	0.7209	0.8378	0.7750
YOLOv5xu	0.7033	0.8649	0.7758
YOLOv5x6u	**0.8472**	0.8243	0.8356

YOLOv5u Model Results

- False Negatives occur when missing bolts are misclassified as background.
- This slightly lowers the overall score.
- The low number of False Negatives contributes to a high recall of 0.9054.



YOLOv8 Model Results

- YOLOv8x and YOLOv8m-oiv7 are top-performing models.
- YOLOv8m-oiv7 has the highest recall at 0.8919, closely following the top score of 0.9054.
- Since recall is the priority, YOLOv8m-oiv7 is preferred in this study.

Model	Precision	Recall	F1
YOLOv8n	0.7901	0.8649	0.8258
YOLOv8n-oiv7	0.7763	0.7973	0.7867
YOLOv8s	0.8250	**0.8919**	0.8571
YOLOv8s-oiv7	0.8769	0.7703	0.8201
YOLOv8m	0.7619	0.8649	0.8101
YOLOv8m-oiv7	0.8462	**0.8919**	0.8684
YOLOv8l	0.7952	**0.8919**	0.8408
YOLOv8l-oiv7	0.8378	0.8378	0.8378
YOLOv8x	**0.9028**	0.8784	**0.8904**
YOLOv8x-oiv7	0.7126	0.8378	0.7702

YOLOv9 Model Results

- YOLOv9t and YOLOv9e both achieved the highest recall score of 0.9054.
- YOLOv9e showed slightly better precision at 0.7701, compared to 0.7444 for YOLOv9t.

Model	Precision	Recall	F1
YOLOv9t	0.7444	**0.9054**	0.8171
YOLOv9s	**0.8025**	0.8784	**0.8387**
YOLOv9m	0.6989	0.8784	0.7784
YOLOv9c	0.7857	0.8919	0.8354
YOLOv9e	0.7701	**0.9054**	0.8323

- It also had the highest F1 score, indicating a stronger balance between precision and recall.
- With these metrics, both models are strong candidates for detecting missing track bolts.

YOLOv9 Model Results

Model	mAP@50 ± CI	mAP@50-95 ± CI	Recall ± CI
YOLOv9t	**0.73720 ± 0.0111**	**0.43336 ± 0.00844**	**0.74035 ± 0.01053**
YOLOv9e	0.57562 ± 0.02370	0.29706 ± 0.01361	0.57689 ± 0.01972

- A confidence interval (CI) test shows that YOLOv9t outperforms YOLOv9e in mAP@50, mAP@50-95, and recall.
- The differences are statistically significant, confirming YOLOv9t as the best-performing model in the YOLOv9 series.

YOLOv10 Model Results

- YOLOv10l showed the best overall performance among the YOLOv10 variants.
- It shares the highest recall of 0.8378 with YOLOv10m and YOLOv10b but outperforms them in all other metrics.
- Its precision is 0.9118, the highest among all YOLO models tested.

Model	Precision	Recall	F1
YOLOv10n	0.8358	0.7568	0.7943
YOLOv10s	0.8462	0.7432	0.7914
YOLOv10m	0.7294	**0.8378**	0.7799
YOLOv10b	0.7045	**0.8378**	0.7654
YOLOv10l	**0.9118**	**0.8378**	**0.8732**
YOLOv10x	0.8356	0.8243	0.8299

Best Models

Model	Precision	Recall	F1
YOLOv5lu	0.8272	**0.9054**	0.8645
YOLOv8x	0.9028	0.8784	**0.8904**
YOLOv9t	0.7444	**0.9054**	0.8171
YOLOv9e	0.7701	**0.9054**	0.8323
YOLOv10l	**0.9118**	0.8378	0.8732

Future Work

- Incorporate video data and diverse weather conditions to improve model robustness.
- Explore newer YOLO versions and test non-YOLO real-time detection models.
- Add a “bolt present” class to evaluate the impact on model performance.

Conclusion

- Machine learning can improve railway safety by detecting missing track bolts efficiently and reliably.
- Four YOLO models were tested: YOLOv5, YOLOv8, YOLOv9, and YOLOv10.
- YOLOv5lu achieved the highest recall score of 0.9054 and showed the best overall performance.
- YOLOv8x can be the best model for detecting missing bolts.
- Results confirm that AI-based object detection is effective for railway defect detection.

References

[1] M. Carranza-García, Jesús Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, “On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data,” *Remote Sensing*, vol. 13, no. 1, pp. 89, Dec. 2020, doi: https://doi.org/10.3390/rs13010089.

[2] U.S. Hansen, “Object Detection: Models, Use Cases, Examples,” *Encord.com*, Apr. 14, 2023. https://encord.com/blog/object-detection/ (accessed Apr. 15, 2025).

[3] Ultralytics, “YOLOv8,” Ultralytics.com, 2023. https://docs.ultralytics.com/models/yolov8/#__tabbed_1_1.

[4] salmaneunus, “Railway Track Fault Detection 1,” *Kaggle.com*, Dec. 09, 2020. https://www.kaggle.com/code/salmaneunus/railway-track-fault-detection-1%20.

[5] R. track, “Railway Track Fault Detection Dataset,” *Roboflow*, 2024. https://universe.roboflow.com/railway-track-crack-detection-using-yolov5/railway-track-fault-detection-hrem8.

[6] SYSTEM THINKING PROJECT, “Damage Detection In Tracks Dataset,” *Roboflow*, 2023. https://universe.roboflow.com/system-thinking-project-4rfww/damage-detection-in-tracks.

[7] Technofly Solution, “Railwaytrackcrackdetection Dataset,” *Roboflow*, 2022. https://universe.roboflow.com/technofly-solution/railwaytrackcrackdetection.

[8] G. Jocher, "Ultralytics/yolov5," GitHub, 2020. [Online]. Available: https://github.com/ultralytics/yolov5.

[9] G. Jocher, A. Chaurasia, and J. Qiu, "YOLOv8 - Ultralytics YOLO Docs," Ultralytics YOLO Docs, 2023. [Online]. Available: https://docs.ultralytics.com/models/yolov8/.

[10] C. Wang, I. Yeh, and H. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," arXiv, 2024. [Online]. Available: https://arxiv.org/abs/2402.13616.

[11] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," arXiv preprint arXiv:2405.14458, 2024.

Thank you

Q & A