Supporting Information for: Deep Learning-based Framework for Mycobacterium Tuberculosis Bacterial Growth Detection for Antimicrobial Susceptibility Testing

Hoang-Anh T Vo[a], Sang Nguyen[a], Ai-Quynh T Tran[a], Han Nguyen[a], Hai Ho Bich[b,c], Philip W Fowler[c,d,e], Timothy M Walker[b,c], Thuy Thi Nguyen[a,*]

[a]*School of Science, Engineering & Technology (SSET), RMIT University, Ho Chi Minh, Vietnam*
[b]*Oxford University Clinical Research Unit, Ho Chi Minh, Vietnam*
[c]*Nuffield Department of Medicine, University of Oxford, Oxford, United Kingdom*
[d]*Health Protection Research Unit in Healthcare Associated Infections and Antimicrobial Resistance, University of Oxford, Oxford, United Kingdom*
[e]*National Institute of Health Research Oxford Biomedical Research Centre, John Radcliffe Hospital, Oxford, United Kingdom*

1. Deep Learning

Deep learning is considered a branch of Machine Learning in Artificial Intelligence, which is motivated by the human brain's function through connections or synapses of nerve cells (neurons) [1]. Leveraging a collection of advanced Artificial Neural Network (ANN) research, deep neural networks are constructed with an input layer and many hidden layers before the output layer. Specifically, high-level data abstraction is derived through multiple processing layers containing complex structures or nonlinear transformation layers [2]. In the last decade, different models using the Convolutional Neural Network (CNN) have been designed, and they apply to many fields of object detection, image recognition, and speech recognition [3]. Some of them have demonstrated outstanding performance in object detection tasks; therefore, they were chosen for experimentation in our study.

1.1. Faster R-CNN:

Faster R-CNN is a single, unified network for object detection, composed of a region proposal network and the Fast R-CNN detector [4]. In particular, a region proposal network (RPN) receives an input image (of any size) and generates a collection of rectangular object proposals, each accompanied by an objectness score. These proposed regions are then used to train the Fast R-CNN detector.

*Corresponding author: thuy.nguyen43@rmit.edu.vn

1.2. Mask R-CNN:

Mask R-CNN is a conceptually simple, flexible, and general framework for segmenting object instances [5]. It extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the class and box outputs. The architecture is composed of two modules—the RPN and Fast R-CNN—with an additional key element for pixel-to-pixel alignment.

1.3. Inception-ResNet:

Inception-ResNet is a deep convolutional neural network architecture that combines the strengths of the Inception architecture with those of residual networks (ResNet). It uses Inception modules to capture features at different scales with parallel convolutions of varying sizes and employs residual connections from ResNet to address the vanishing gradient problem and enhance training efficiency [6].

1.4. YOLOv8:

Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model that enhances the performance and versatility of earlier YOLO models by introducing new features and improvements [7]. In 2016, Joseph Redmon and Santosh Divvala developed the YOLO architecture [8]. The underlying principle of YOLO is to divide the input image into a grid and make predictions about bounding boxes and class probabilities for each grid cell.

2. Performance Metrics

To evaluate the performance of SOTA deep learning models for the growth detection task in TMAS, we adopted widely used evaluation metrics, including precision, recall, F1-score, and mAP [9]. These metrics are computed based on the following fundamental components:

- **True Positive (TP)**: The model accurately identifies the presence of microbial growth where growth is indeed present.
- **False Positive (FP)**: The model incorrectly identifies the presence of microbial growth in areas where no growth is actually present.
- **True Negative (TN)**: The model accurately identifies the absence of microbial growth where no growth is present.
- **False Negative (FN)**: The model fails to identify microbial growth, incorrectly predicting the absence of growth in areas where growth is actually present.

2.1. Precision

Precision (also known as positive predictive value) is the proportion of the accurate positive predictions over the total number of positive predictions. In the clinical context, Precision demonstrates the percentage of positive wells in which bacteria are present out of the total wells flagged to contain bacterias. A high Precision denotes fewer false positives, resulting in more confident decision-making and fewer unnecessary follow-up steps.

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

2.2. Recall

Recall (also called Sensitivity) is the ratio of true positive predictions to the total number of actual positive samples. In the clinical context, it measures the ability of the method to identify all wells that have MTB growth correctly. A high recall ensures that few wells with bacteria are missed, thereby preventing under-diagnosis or underestimation of MTB presence.

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

2.3. F1

Although Precision and Recall are good performance metrics, neither of them alone provides a comprehensive analysis. A model can achieve perfect or near-perfect Precision but low Recall and vice versa. To address this limitation, the F1 Score, defined as the harmonic mean, enables the merging of these metrics into one balanced measure. In a clinical context, the F1 Score is crucial for MTB detection, where accuracy is needed to avoid unnecessary treatments while ensuring that no true cases are overlooked.

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

2.4. mAP

mAP is defined as the mean of Average Precision (AP) across all detected classes, providing an overall measure of Precision and recall across different thresholds. A high mAP indicates a consistent performance of the method across different thresholds. In the clinical context, it is crucial when the method needs to be adaptable to different patient groups, stages of disease progression.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

3. Training Process

3.1. Model Selection

The model learning phase is a critical step in the deep learning training process for growth detection, plate reading, and interpretation. During this phase, a large number of annotated images were fed as input for training, enabling the deep learning models to learn representative features from the images without manual feature extraction. These extracted features were then used to classify images into different categories. In this study, four advanced deep learning models were chosen: Faster R-CNN [4], Mask R-CNN [5], Inception-ResNet [6], and YOLOv8 [10]. These models are among the state-of-the-art models for object detection and segmentation, capable of providing accurate and rapid results. An efficient and popular technique called transfer learning was employed, which involves fine-tuning a pre-trained model with our dataset for the specific task of plate reading [11].

3.2. Model Training

The primary goal of the training phase was to minimize the loss function by achieving convergence. Several key parameters were customized to optimize performance. The learning rate was initially modified to determine the optimal value, as an excessively high learning rate may cause premature convergence, while a too low rate delays convergence [12]. Instead of a fixed value, the learning rate was set to 0.0001 and decayed by a factor of 0.5 every 2 epochs. Additionally, the choice of optimizer impacts convergence; different optimizers such as SGD, Adam, and RMSprop update model parameters using varying mechanisms. In our case, the Adam optimizer was chosen due to its fast convergence speed and lower requirements for memory and computational resources. Observations during training indicated that despite minor fluctuations, the loss values steadily decreased, and the metrics—Accuracy, Precision, Recall, and F1 Score—gradually improved. To validate the consistency and reproducibility of TMAS, we performed five independent training runs, each using a differently shuffled version of the training and validation set. For each shuffled set, the models were trained for 20 epochs. The resulting variation in those performance metrics was minimal (ranging from 0.001 to 0.031), demonstrating the model's robustness and reliability.

4. Analysis of Training and Validation Loss Trends

A detailed analysis of the training and validation loss graphs was provided for each state-of-the-art model—including YOLOv8, Faster R-CNN, Mask R-CNN, and Inception-ResNet—with the losses represented by blue and orange lines, respectively. As shown in Fig. 1, all models generally exhibited a downward trend in losses, indicating effective learning throughout the training process.

Specifically, YOLOv8 demonstrated a narrowing gap between training and validation losses over time, suggesting improved generalization. Although early

epochs showed variability in both losses across folds, later epochs marked a marked reduction, enhancing stability and consistency. The Faster R-CNN model displayed a stable trend with minimal fluctuations, and the narrow convergence between its training and validation loss curves suggested robust generalization without significant overfitting or underfitting. The Mask R-CNN model showed an early decrease in both losses, though the validation loss plateaued while the training loss continued to decrease—indicative of potential overfitting—yet the variability across folds remained low. In contrast, the Inception-ResNet model maintained high and nearly constant loss values throughout training, suggesting limited improvement and possibly a need for adjustments in the training strategy.

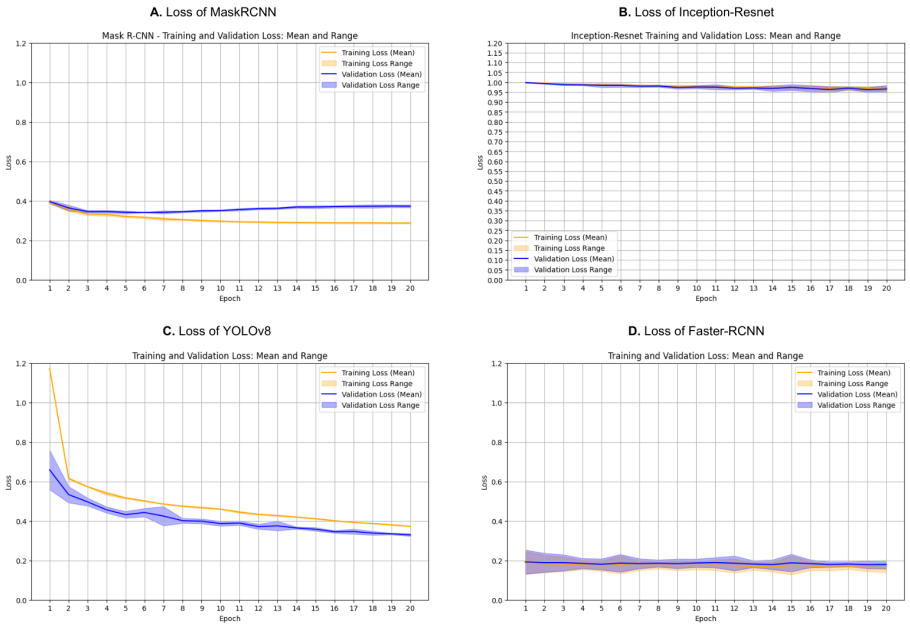

Fig. S1. Loss graph for different proposed models.

References

[1] A. U. Ibrahim, E. Guler, M. Guvenir, K. Suer, S. Serte, M. Ozsoz, Automated detection of Mycobacterium tuberculosis using transfer learning, The Journal of Infection in Developing Countries 15 (05) (2021) 678–686. doi:10.3855/jidc.13532.

[2] Z. Hao, Deep learning review and discussion of its future development, MATEC Web of Conferences 277 (2019) 02035. doi:10.1051/matecconf/201927702035.

[3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.

[4] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, version Number: 3 (2015). doi:10.48550/ARXIV.1506.01497.

[5] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, version Number: 3 (2017). doi:10.48550/ARXIV.1703.06870.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, version Number: 2 (2016). doi:10.48550/ARXIV.1602.07261.

[7] Ultralytics: Yolo vision (2024).
URL https://github.com/ultralytics/ultralytics

[8] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640 [cs] (May 2016). doi:10.48550/arXiv.1506.02640.

[9] R. Szeliski, Computer vision: algorithms and applications, Springer Nature, 2022.

[10] Ultralytics, YOLOv8.
URL https://docs.ultralytics.com/models/yolov8

[11] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, M. A. Azim, Transfer learning: a friendly introduction, Journal of Big Data 9 (1) (2022) 102. doi:10.1186/s40537-022-00652-w.

[12] J. Guo, AI Notes: Parameter optimization in neural networks.
URL https://www.deeplearning.ai/ai-notes/optimization/