Week 14: Peer Review of the Breast Cancer Detection Project

Yujia Cao, Wendi Yuan, Yuhan Zhao

We had the opportunity to collaborate with Yinda and Alice on their project focusing on breast cancer detection, reviewing their GitHub repository, and providing constructive feedback to enhance their work. Overall, their effort demonstrates thoughtful organization, technical depth, and a commitment to addressing an important healthcare challenge.

GitHub Repository Structure

Folder naming and structure clarity: The repository is thoughtfully organized, but folder names like "doc" could be more explicit. The folder structure provides a logical framework, making the project easy to navigate for those familiar with the subject. However, certain folder names, such as "doc," lack clarity and could benefit from more descriptive naming conventions to make the repository more accessible for non-expert users.

Additional README.md files: Adding README.md files to key folders would improve clarity and accessibility for new users.While the primary README.md file is clear and helpful, adding explanatory README.md files in subfolders (e.g., "doc") would guide users in understanding the purpose and content of these directories. This enhancement would make the repository more user-friendly, especially for individuals less familiar with the project's structure.

Minimizing unnecessary files: Files like ".gitkeep" and "__init__.py" in the "src" folder could be clarified or removed to reduce clutter. The inclusion of files such as .gitkeep and __init__.py adds unnecessary complexity to the repository. If these files serve no critical purpose, they should be removed to enhance the repository's readability and maintain a clean organizational structure.

Code and Jupyter Notebooks

Documentation and logical flow: The notebooks are well-documented, with clear annotations and a logical progression.The notebooks effectively communicate the project's development, with annotations that explain each step and a logical structure that allows readers to follow along easily. The clear progression of tasks from data preprocessing to model deployment showcases the team's methodological rigor.

Enhanced visualization tools: Visualizations are effective but could benefit from additional tools like confusion matrices or Grad-CAM for deeper model insights.The inclusion of visualizations in the "src" folder highlights a commitment to making data insights accessible. Incorporating advanced tools like confusion matrices or Grad-CAM would provide a more comprehensive understanding of model predictions, bridging the gap between raw outputs and their practical implications.

Professionalism and error handling: Informal expressions and lack of robust error handling could detract from the professionalism of the notebooks.While the informal commentary adds an engaging tone, it may distract from the professional presentation of the work. Strengthening error handling and including more detailed docstrings or comments explaining critical code segments would improve both the usability and professional appearance of the notebooks.

Code Outputs

Insightful performance metrics: Outputs effectively demonstrate key metrics like losses and accuracy. The outputs provide a clear view of model performance, with metrics like losses and accuracy helping to evaluate progress. These insights guide iterative improvements and demonstrate a thorough understanding of model evaluation techniques.

Interpretability for stakeholders: Raw predictions could be translated into user-friendly labels or probabilities for better clarity. Transforming raw predictions into interpretable outputs, such as “Prediction: Malignant with 95% confidence,” would make the results more accessible to stakeholders. This approach ensures that technical outputs are easily understood by non-technical users.

Context for advanced techniques: Methods like quantization and pruning need better contextual explanation to highlight their impact. Advanced optimization techniques like quantization and pruning are critical to improving model performance, but their significance is not fully explained. Including a brief context about how these methods affect the model’s efficiency and deployment would enhance understanding for both technical and non-technical audiences.

Reports

Narrative and technical depth: Reports are well-structured, illustrating both technical progress and real-world applicability. The reports are comprehensive and present a clear narrative of the project's technical advancements. Each week’s report builds on previous progress, emphasizing the application of TensorFlow Lite in breast cancer detection. This detailed documentation showcases the team’s ability to connect technical development with practical solutions.

Linking progress to the problem statement: Weekly progress could be more explicitly tied to the main problem statement. While the reports provide valuable insights into the project’s weekly progress, they could better connect each phase to the central problem statement. Strengthening this link would help demonstrate how individual efforts contribute to solving the overarching challenge of improving breast cancer detection.

Consistency and additional challenges: Formatting inconsistencies and limited discussion of challenges (e.g., bias, integration) reduce the impact of the reports. Addressing formatting inconsistencies (e.g., spacing, indentation) would enhance the professional appearance of the reports. Additionally, discussing challenges such as model bias, ethical concerns, or integration with existing healthcare systems would add depth and demonstrate a more holistic understanding of the project’s implications.