Configuration Manual

MSc Research Project
M.Sc Data Analytics

Pooja Santhosh
Student ID: x23274875

School of Computing
National College of Ireland

Supervisor: John Kelly

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Pooja Santhosh………………………………………………………………………………

Student ID: …………x23274875…………………………………………………………………………………

Programme: ………M.Sc Data Analytics…………………………… **Year:** ……2024-2025..

Module: MSc Research Practicum

Supervisor: …………John Kelly……………………………………………………………………….………

Submission Due Date: ………11/08/2025…………………………………………………………………………….………

Project Title: CNN-Based Deep Learning Models for Automated Tuberculosis Detection from Chest X-rays
………………………………………………………….………

Word Count: ………1035……………………… **Page Count**……………..3.…………………….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: POOJA SANTHOSH ……………………………………………………………………………………………………

Date: …11/08/2025………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Pooja Santhosh
Student ID: x23274875

1 Introduction

Tuberculosis (TB) remains one of the leading health concerns of the world especially in the low-resource regions where radiologists who can help one with diagnosis are scarce. Although chest X-ray imaging is prevailing examinations used in the screening of TB, it relies strongly on interpretation of experts and may prejudice performance and delay in areas that are under-privileged. Luckily, due to the advent of artificial intelligence, methods of deep learning, and Convolutional Neural Networks (CNNs) have a very high potential of automating the process of analyzing medical images. The purpose of this project is to carry out the comparison of the performance of the three most popular and popular pre-trained CNN architectures ResNet18, DenseNet121, and EfficientNet-B0 to see which model is more suitable in the problem of detecting TB based on a chest X-ray. In this config file we will discuss in details of the executing of the codes and the analysis of the research that has been done

2 Section A: System Configurations

Environment: Google Colab
Installed Packages:

- torch, torchvision
- scikit-learn
- matplotlib, seaborn
- tqdm
- PIL
- ultralytics (YOLOv8)

3 Section B: Code Explanation

Step 1: Kaggle API & Dataset Download

- Upload kaggle.json to authenticate Kaggle API.
- Create .kaggle directory, move kaggle.json there, and set permissions.
- Download TBX-11 dataset from Kaggle.

Step 2: Dataset Preparation

- Folders have been moved to the root TBX11K, NAID path.
- train_transform and val_test_transform (resize, normalization) are defined
- Dataset has been loaded with ImageFolder and split into train, validation, and test sets using StratifiedShuffleSplit.
- A custom Dataset is implemented class to apply transforms only to the chosen split.

Step 3: Handling Class Imbalance

- Compute class_weights for loss balancing.
- Use WeightedRandomSampler to oversample minority classes in the training set.

Step 4: Model Building

- Functions to build:
  - ResNet50
  - DenseNet121

    - EfficientNetB0
    - YOLO Models

- Replace final classification layer to match **3 output classes**: Healthy, Sick, TB.

Step 5: Training Function

- Use Adam optimizer, learning rate = 1e-4, CrossEntropyLoss with class weights.
- Track train/validation losses and validation ROC-AUC.
- Implement early stopping based on validation loss.

Step 6: Evaluation Function

- Generate classification report and confusion matrix.
- Calculate ROC-AUC and plot ROC curves (One-vs-Rest).

Step 7: Training & Testing

- Train and evaluate each model (ResNet50, DenseNet121, EfficientNetB0).
- Plot training metrics (loss & ROC-AUC curves).

Step 8: YOLOv8 Dataset Organization

- Organize images into YOLOv8-compatible train/, val/, test/ directories, each containing class subfolders.
- Confirm number of images in each split.

4 Section C: File Locations

- Google Colab Notebook: https://drive.google.com/drive/folders/1tebY5E_azKbUPbXQLrEsFYvVspsWIYU9?usp=drive_link
- Dataset Source: Kaggle dataset usmanshams/tbx-11
  TB_Chest_Radiography_Database: https://drive.google.com/drive/folders/1yZ81dV121f243zZGZkDhrdHOyUBi_-sT?usp=drive_link

5 Section D: How to execute code

Step-1: Authenticate Kaggle

- Upload your kaggle.json file to Colab.
- Run setup cell to create .kaggle directory and set permissions.

Step-2: Download & Extract Dataset

- Use Kaggle API to download TBX-11 dataset.
- Unzip and move folders to correct locations.

Step-3: Run Data Preparation

- Apply transformations and create train/val/test splits.
- Check class distribution.

Step-4: Train Models

- Run training cells for each model.
- Observe loss and ROC-AUC plots.

Step-5: Evaluate Models

- Review classification reports, confusion matrices, and ROC curves.

Step-6: Organize YOLOv8 Dataset

- Execute dataset organization cell.
- Verify YOLOv8 format and class folders.

Step-7: Run YOLOv8

- Load YOLO model from ultralytics.
- Train using model.train() on organized dataset.

6 Section E: References

[1] Wekalao, J. (2025). Graphene-enhanced terahertz metamaterial biosensor for tuberculosis detection with XGBoost-based machine learning optimization. Plasmonics, pp.1–16.

[2] Acharya, V., Dhiman, G., Prakasha, K., Bahadur, P., Choraria, A., Prabhu, S., Chadaga, K., Viriyasitavat, W., & Kautish, S. (2022). AI-assisted tuberculosis detection and classification from chest X-rays using a deep learning normalization-free network model. Computational Intelligence and Neuroscience, 2022(1), p.2399428.

[3] Hrizi, O., Gasmi, K., Ben Ltaifa, I., Alshammari, H., Karamti, H., Krichen, M., Ben Ammar, L., & Mahmood, M.A. (2022). Tuberculosis disease diagnosis based on an optimized machine learning model. Journal of Healthcare Engineering, 2022(1), p.8950243.

[4] Tasci, E., Uluturk, C., & Ugur, A. (2021). A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection. Neural Computing and Applications, 33(22), pp.15541–15555.

[5] Nafisah, S.I., & Muhammad, G. (2024). Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence. Neural Computing and Applications, 36(1), pp.111–131.

[6] Bantelay, L.M., Abebe, M., Rajendran, R.S., & Sungheetha, A. (2023, November). Heuristic Pneumonia and Tuberculosis Detection in X-Ray Images Using Convolutional Neural Networks. In 2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS) (pp. 1–7). IEEE.

[7] Huy, V.T.Q., & Lin, C.M. (2023). An improved densenet deep neural network model for tuberculosis detection using chest x-ray images. IEEE Access, 11, pp.42839–42849.