DIAGNOSIS OF PULMONARY TUBERCULOSIS DISEASE USING DEEP LEARNING

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

K.TEJASWI	(21UECM0108)	**(20190)**
K.KEERTHI	(21UECM0107)	**(20317)**
SANAGAVARAPU SHALINI	(21UECM0213)	**(20217)**

Under the guidance of
Dr. K. ANTONY KUMAR, M.E.,Ph.D.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

January, 2024

DIAGNOSIS OF PULMONARY TUBERCULOSIS DISEASE USING DEEP LEARNING

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

K.TEJASWI	(21UECM0108)	**(20190)**
K.KEERTHI	(21UECM0107)	**(20317)**
SANAGAVARAPU SHALINI	(21UECM0213)	**(20217)**

Under the guidance of
Dr. K. ANTONY KUMAR, M.E.,Ph.D.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

January, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”DIAGNOSIS OF PULMONARY TUBERCULOSIS DISEASE USING DEEP LEARNING” by ”K.TEJASWI (21UECM0108), K.KEERTHI (21UECM0107), SANAGAVARAPU SHALINI (21UECM0213)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Dr. K. Antony Kumar
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January, 2024

Signature of Head of the Department
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January, 2024

Signature of the Dean
Dr. V. Srinivasa Rao
Professor & Dean
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. Wealso declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

K.TEJASWI

Date: / /

K.KEERTHI

Date: / /

SANAGAVARAPU SHALINI

Date: / /

APPROVAL SHEET

This project report entitled DIAGNOSIS OF PULMONARY TUBERCULOSIS DISEASE USING DEEP LEARNING by K.TEJASWI (21UECM0108), K.KEERTHI (21UECM0107), SANAGAVARAPU SHALINI (21UECM0213) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners	**Supervisor**
	Dr.K. ANTONY KUMAR, M.E.,Ph.D.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for provid- ing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr. M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor, Dr. K. ANTONY KUMAR, M.E.,Ph.D.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E., Mr. SHARAD SHANDHI RAVI, M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

K.TEJASWI	**(21UECM0108)**
K.KEERTHI	**(21UECM0107)**
SANAGAVARAPU SHALINI	**(21UECM0213)**

ABSTRACT

Tuberculosis(TB) is a communicable disease caused by Mycobacterium tuberculosis. It poses a global health challenge with its airborne transmission, affecting millions worldwide. The insidious nature of TB, often latent for years before manifest- ing symptoms, underscores the urgency for precise and timely detection to preventits progression to severe and potentially fatal stages. Early detection and intervention are crucial in curbing its spread. This project addresses the critical issue of TB de- tection through the development of a novel prototype utilizing Convolutional Neural Network (CNN) algorithms. CNNs are a class of Deep Learning (DL) algorithms designed for image recognition and processing. Inspired by the visual processingof the human brain, CNNs excel in extracting hierarchical features from input data, making them widely employed in various computer vision tasks. The convolutional layers of the CNN systematically apply filters to the input X-ray images, enablingthe extraction of intricate spatial hierarchies and critical features indicative of TB.By focusing on the initial stages of TB, the project aims to enhance diagnostic ac- curacy, enabling early intervention and treatment to curtail the spread of the disease. This innovative approach not only contributes to the advancement of medical diag- nostics but also holds the potential to significantly reduce the impact of tuberculosis on public health, ultimately saving lives through early intervention

Keywords: Convolutional Neural Network, Deep Learning, Early Detection, Features Extraction, Filters, Image Recognition, X-Ray Image .

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
CAD	Computer Aided Diagnosis
CNN	Convolutional Neural Networks
DCNN	Deep Convolutional Neural Networks
DL	Deep Learning
ML	Machine Learning
TB	Tuberculosis
WHO	World Health Organisation

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Tuberculosis (TB) stands as one of humanity’s oldest and persistent health challenges, wielding a profound impact on global public health. Caused by the bacterium Mycobacterium tuberculosis, this infectious disease primarily targets the lungs, although it can affect various organs and systems within the body. TB spreads through the air when an infected individual coughs or sneezes, making it highly contagious. The World Health Organization (WHO) estimates that nearly one-quarter of the global population is infected with the TB bacteria, with a vast majority residing in developing regions. Despite significant strides in medical science, TB continues to be a leading cause of morbidity and mortality worldwide. The emergence of drug-resistant strains further complicates treatment efforts, requiring innovative approaches to tackle this resilient pathogen.Societal factors such as poverty, overcrowded living conditions, and compromised immune systems contribute to the persistence of TB in certain populations. As the world strives for comprehensive global health, understanding and addressing the multifaceted chal- lenges posed by tuberculosis remain critical in the pursuit of effective prevention, diagnosis, and treatment strategies.

Convolutional Neural Network (CNN) represent a groundbreaking advance-ment in Artificial Intelligence (AI), particularly in the realm of image processingand pattern recognition. These Deep Learning (DL) models mimic the visual pro- cessing of the human brain and excel at extracting intricate features from images. In the field of healthcare, CNNs have revolutionized diagnostic capabilities. For TB, CNNs analyze medical imaging, such as chest X-rays, with remarkable accuracy.By identifying characteristic patterns indicative of tuberculosis, these networks con-tribute significantly to early and precise diagnosis, enabling prompt intervention and treatment. The integration of CNNs in medical imaging showcases the potential of AI to enhance healthcare outcomes and address global health challenges.

1.2 Aim of the Project

The aim of this project is to develop a robust CNN model for the automated detection of TB through the analysis of chest X-rays. Leveraging the power of deep learning, the objective is to create an accurate and efficient diagnostic model that can assist healthcare professionals in identifying tuberculosis cases swiftly and reliably. By training the CNN on a comprehensive dataset of chest X-ray images, the project seeks to enhance the model's ability to discern subtle patterns indicative of tuberculosis, thereby contributing to early detection and timely intervention.

1.3 Project Domain

DL represents a subfield of Machine Learning (ML) that has garnered immense attention and success, particularly in tasks involving complex data and intricate patterns. At its core, DL is inspired by the neural networks of the human brain, employing artificial neural networks with multiple layers (deep neural networks) to learn and make predictions from data. CNN, a specialized form of DL, have proven exceptionally effective in image recognition tasks the very foundation of this tuberculosis detection project. DL algorithms autonomously extract hierarchical features from data, allowing them to discern patterns and representations that might be challenging for traditional machine learning approaches.

In the context of this TB detection project, the DL model, specifically the CNN, serves as a potent tool for automated analysis of chest X-ray images. The multi-layered architecture of the CNN enables it to automatically learn and differentiate between subtle features indicative of TB, facilitating accurate and efficient detection. Through a process known as backpropagation, the CNN refines its parameters during training optimizing its ability to recognize complex patterns within the X-ray images. This adaptability is crucial in medical imaging, where variations in patient demographics and disease presentation require a model capable of nuanced learning.

The versatility of DL extends beyond its ability to discern patterns, it allows for the extraction of high-level representations from raw data. The project's reliance on DL aligns with the broader trend in healthcare, where AI technologies are increas- ingly employed to enhance diagnostic accuracy, expedite processes, and alleviate the burden on healthcare professionals. As the project delves into the realm of deep learning, it seeks not only to develop a sophisticated TB detection model but also to contribute to the evolving landscape of AI applications in healthcare, ultimately improving patient care and outcomes.

1.4 Scope of the Project

The project holds significant promise within the domain of healthcare, specifically in the early and accurate detection of TB through the utilization of CNN. The scope of this endeavor is multifaceted, encompassing advancements in both technology and public health. Foremost, the project seeks to address a critical need in TB diagnosis by leveraging the capabilities of deep learning. The deployment of CNNs enables the automated analysis of chest X-ray images, presenting a streamlined and efficient approach for identifying subtle patterns indicative of TB. By doing so, the project aims to contribute to early detection, a crucial factor in the successful management and treatment of TB.

The scope extends to the potential impact on healthcare infrastructure, particularly in regions where TB remains a significant public health concern. The automation of the diagnostic process through the trained CNN model has the capacity to alleviate the workload of healthcare professionals, enabling them to focus on treatment planning and patient care. Moreover, the project envisions scalability, aiming to develop a model that can be integrated into existing healthcare systems with relative ease, thereby broadening its accessibility and impact.

The project aligns with the global trend of integrating AI into healthcare practices. As healthcare systems worldwide grapple with increasing demands and complexity, the incorporation of advanced technologies like deep learning becomes imperative. The success of this project could serve as a testament to the potential of AI-driven diagnostic tools in transforming the landscape of healthcare delivery.

Chapter 2

LITERATURE REVIEW

Rahman et al, in 2020 presented a reliable tuberculosis detection method using chest X-rays with deep learning, segmentation, and visualization techniques. The authors propose a modified U-Net architecture with bidirectional Convolutional Long Short Term Memory (BConvLSTM) to combine feature maps extracted from the corresponding contracting path and the previous expanding up-convolutional layer. The proposed method achieves high accuracy and outperforms other state-of-the-art methods. However, the dataset used in this study is limited to only two classes (TB and normal), which may limit the generalizability of the proposed method to more diverse datasets.he proposed method requires a large amount of computational resources, which may limit its practical application in resource-limited settings.

Hooda et al, in 2017 presented a deep-learning-based method for TB detection using chest radiographs. The proposed method uses a CNN architecture with 19 layers, including Conv and ReLu layers, achieving an accuracy of 94.73 using the Adam optimizer. The method was evaluated on two publicly available datasetsand compared with existing approaches, demonstrating competitive performance.However, the study's limitation lies in the relatively small dataset size, warranting validation on larger and more diverse datasets. Additionally, the impact of the proposed method on clinical practice and its generalizability to different populations remains to be investigated. Further clinical studies are required to validate theproposed method.

Kant et al, in 2018 proposed a deep learning-based method for automated detection of TB using sputum microscopy images. The proposed method can detect andlocalize drug-sensitive Mycobacterium tuberculosis bacilli, potentially speeding up the TB screening process and reducing human error. However, the dataset used inthe study is small compared to what is generally used to train deep learning models, and the method's accuracy and generalizability may improve with a larger dataset. Additionally, the proposed method only addresses the detection of TB and does not provide a complete automated pipeline for TB diagnosis using sputum smear microscopy.

Yadav et al, in 2018 explored the use of Deep Learning to classify X-ray images of potential Tuberculosis patients, achieving an impressive overall accuracy of 94.89 on the augmented images. The researchers utilized Coarse-to-Fine Knowledge Transfer, data augmentation, and fine-tuning techniques to enhance the learning rate and reduce errors. However, the study's limitation is that the model is specifically trained for the China dataset, and the accuracy cannot be guaranteed on chest X-ray images taken in different settings. Nonetheless, the findings suggest that these techniques, when used in conjunction, can yield high accuracy results in classifying medical data, which could significantly impact the diagnosis and treatment of Tuberculosis patients.

Rajaraman et al, in 2020 demonstrated the effectiveness of ensemble learning and modality-specific knowledge transfer in enhancing TB detection from chest radio-graphs. By training deep learning models on large-scale chest x-ray collections and combining modality-specific knowledge through ensemble learning, they achieved superior performance compared to state-of-the-art literature. However, the study's limitations include evaluation with a small sample size, lack of reported values for certain performance metrics, and the computational expense of ensemble methods. Additionally, the need for interpretability and visualization studies to explain model predictions is highlighted.

Liu et al, in 2017 proposed a novel deep learning method using CNN and transfer learning to classify TB manifestations in chest X-ray images, achieving an 85.68 classification accuracy in a large TB dataset. The method shows potential for accurate TB diagnosis in resource-poor settings and offers stability across various CNN architectures. However, the drawback of the paper is the lack of detailed discussion on the generalizability of the proposed method to diverse populations and healthcare settings, as well as the potential biases in the dataset used for training the CNN models.

Nguyen et al, in 2017 presented an enhanced approach for tuberculosis detection from chest X-ray images using transfer learning and a novel method for acquiring low-level features. The study demonstrates that low-level features from ImageNet weights are inadequate for imaging tasks like X-rays. Additionally, the authors develop a tuberculosis screening application and recommend optimal architectures for improved generalization. However, the research is constrained by a small dataset and lacks external validation, potentially limiting the applicability of the proposed method to broader clinical settings.

Stirenko et al, in 2018 explored the application of deep learning, lung segmentation, and data augmentation for tuberculosis detection from chest X-ray images. It demonstrates the effectiveness of these techniques, even with a small and unbal- anced dataset, showcasing potential for accurate predictions. However, a drawback of the study is the limited exploration of the impact of more extensive data augmen- tation techniques on tuberculosis predictions. Further investigation into the role of information loss during data augmentation and the effects of outliers in the dataset could enhance the comprehensiveness of the findings.

Ho et al, in 2019 investigated the application of DCNNs for TB detection from chest radiographs, demonstrating promising results and emphasizing the potential of deep learning in revolutionizing TB diagnosis. However, a drawback of the paper is the limited discussion on the challenges and limitations of using DCNNs for TB detection, such as potential biases in the training data, interpretability of the deep learning models, and the need for validation on diverse and representative datasets. A more comprehensive analysis of these limitations would enhance the paper’s contribution to the field of automated TB detection.

Norval et al, in 2019 explored the use of CNN to detect Pulmonary Tuberculosis from chest X-ray images. It investigates various image preprocessing methods anda hybrid approach to improve detection accuracy, with promising results. However, the study’s drawback lies in the limited discussion of potential limitations or challenges associated with the proposed methods. Additionally, while the researchis supported by South African National Research Foundation Grants and Incentive Grant, the specific limitations of the study, such as the size and representativeness of the dataset, are not thoroughly addressed.

Researchers presented diverse methods for TB detection using deep learning and imaging techniques. Novel architectures, such as modified U-Net with BCon- vLSTM, CNN with 19 layers, and ensemble learning, demonstrated high accuracy. Challenges include limited dataset diversity, computational resource requirements, and potential biases. Some methods address specific TB aspects, like sputum mi- croscopy images. While promising, studies lack comprehensive discussions on lim- itations, dataset biases, and broader applicability. Future research should focus on larger, more diverse datasets, addressing computational constraints, and evaluating practical implications for TB diagnosis in various settings.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system for TB diagnosis relies heavily on manual interpretation of medical imaging, particularly chest X-rays, by healthcare professionals. In this traditional approach, radiologists examine X-ray images for characteristic signs of TB, such as abnormal lung patterns or the presence of nodules. However, this processis inherently subjective and can be prone to human error, especially in cases wherethe disease manifestation is subtle or in its early stages. The demand for efficientand accurate diagnostics becomes crucial given the global prevalence of TB andthe need for prompt intervention. Moreover, the manual interpretation of chest X- rays is labor-intensive and time-consuming, contributing to delays in diagnosis and treatment initiation. In resource-constrained settings, where the burden of TB is often highest, these delays can have significant consequences for patient outcomes.

In recent years, computer-based prediction systems have emerged as an alternative to manual assessments. These systems employ ML techniques to automatically analyze chest X-ray images and provide predictions regarding the presence or ab- sence of TB. While these computer-based approaches offer a degree of automation and efficiency, they face challenges related to accuracy. Many existing systems ex- hibit limitations in achieving high levels of accuracy and specificity, leading to con- cerns about their reliability in real-world clinical settings. False positives and false negatives are not uncommon, impacting the overall effectiveness of these systems.

The accuracy of ML techniques in the existing systems is often contingent upon the size and diversity of the training datasets. Inadequate representation of diverse patient populations or variations in imaging conditions can hinder the generalization capability of these models.As a result, there is a recognized need for advancements in computational methods that can enhance the precision and reliability of TB detection from X-ray images. The limitations of existing systems underscore the importance of continuous research and development efforts to refine ML approaches, aiming for more accurate and robust diagnostic tools in the realm of medical imaging.

3.2 Proposed System

The proposed system for TB detection represents a transformative leap in diagnostic methodologies to address the limitations of the existing manual interpretation approach. At the core of the proposed system is the implementation of CNN, a form of DL renowned for its prowess in image analysis. The system is designed to autonomously scrutinize chest X-ray images, discerning intricate patterns associated with TB with a high degree of accuracy. By training the CNN on a diverse and ex- tensive dataset of chest X-rays, the model learns to recognize subtle manifestationsof the disease, offering a more objective and consistent diagnostic approach.

One of the primary advantages of the proposed system is its potential to significantly reduce the subjectivity and variability inherent in manual interpretation. The CNN's ability to systematically analyze images ensures a standardized and reproducible assessment, mitigating the risk of human error and enhancing overall diagnostic reliability. Moreover, the proposed system holds the promise of expedit- ing the diagnostic process, providing a timely response crucial for initiating prompt interventions and treatment plans. In regions where access to skilled radiologists may be limited, the proposed system becomes especially valuable, offering a tool that augments existing healthcare capabilities.

The scalability and adaptability of the proposed system further contribute to its potential impact.Once trained on a diverse dataset, the CNN model can be deployed across various healthcare settings, catering to different populations and demographics. Additionally, the system facilitates continuous learning and improvement, adapting to evolving patterns in TB manifestations and enhancing its diagnostic capabilities over time.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of the proposed TB detection project is underscored by its potential to streamline healthcare processes, reduce overall diagnostic costs, and improve patient outcomes. While initial investments are required for hardware, software, and data acquisition, the long-term economic benefits are substantial. The automation of TB diagnosis through CNN promises significant time savings, enabling healthcare professionals to allocate their expertise more efficiently. This, in turn, can contribute to increased patient throughput and reduced waiting times for diagnostic results. The economic impact is particularly pronounced in regions facing a scarcity of skilled radiologists, where the proposed system offers a cost-effective solution to bridge the expertise gap.

Furthermore, the proposed system has the potential to minimize the economic burden associated with delayed TB detection. Timely identification of the disease can lead to earlier interventions, reducing the overall cost of treatment and prevent- ing the spread of the disease within communities. In resource-constrained settings, where TB is prevalent, the economic feasibility of the project is heightened by its capacity to operate autonomously, requiring minimal ongoing labor costs once the system is implemented.

3.3.2 Technical Feasibility

The technical feasibility of the proposed TB detection project is robust, driven by advancements in deep learning and the proven effectiveness of CNN in image analysis. The availability of powerful processesor, ensures the computational ca- pabilities required for training complex neural networks on extensive chest X-ray datasets. Open-source deep learning frameworks like TensorFlow and PyTorch pro- vide a solid foundation for model development and deployment, offering a wealth of pre-built tools and libraries that expedite the implementation process. The use of cloud computing platforms, such as AWS or Google Cloud, further enhances tech- nical feasibility by providing scalable resources, eliminating the need for substan- tial upfront hardware investments, and facilitating remote access to computational power.

As the project hinges on image data, the technical feasibility is also bolstered by the accessibility of diverse and comprehensive chest X-ray datasets, ensuring that the CNN can be trained on a representative range of cases. Overall, the convergence of advanced hardware, software frameworks, and cloud technologies positions the project as technically feasible, with the potential to contribute significantly to the automated and accurate detection of TB through innovative DL methodologies.

3.3.3 Social Feasibility

The social feasibility of the proposed TB detection project holds substantial promise in addressing key societal challenges associated with the diagnosis and management of TB. One of the primary social benefits is the potential to enhance healthcare accessibility, particularly in underserved regions where the burden of TB is high, and skilled healthcare professionals are scarce. The automated and objective nature of CNN in analyzing chest X-rays could alleviate the strain on overburdened healthcare systems, facilitating quicker and more widespread diagnostics. Addition- ally, the project aligns with the broader global health agenda by contributing to the early detection of TB, a crucial factor in reducing transmission rates and preventing the progression of the disease to advanced stages.

Furthermore, the proposed system has the potential to address disparities in healthcare access by providing a standardized diagnostic approach. By minimizing the subjectivity inherent in manual interpretation, the CNN-based system ensures consistent and reliable diagnostics, reducing disparities in healthcare outcomes that may arise due to variations in expertise or resource availability. This social inclu- sivity is particularly pertinent in regions with diverse demographic profiles, wherethe proposed system can adapt to different populations and contribute to equitable healthcare delivery.

3.4 System Specification

3.4.1 Hardware Specification

- **Central Processing Unit (CPU):**
  - Processor 12th Gen Intel(R) Core(TM) i5-1235U
- **Memory (RAM):**
  - DDR4 (Minimum 32GB)
- **Storage:**
  - Solid State Drive (SSD)

3.4.2 Software Specification

- **Programming Language:**
  - Python (version 3.6 or higher)
- **Integrated Development Environment (IDE):**

  - Visual Studio Code (latest version) or Jupyter Notebooks

- **Cloud Platform :**
  - Google Colab or AWS for scalable cloud-based computations

3.4.3 Standards and Policies

Google Colab

Google Colab, short for Colaboratory, is an online platform provided by Google that offers a cloud-based environment for running Python code, particularly popular for its integration with Jupyter Notebooks. Unlike local environments, Google Colab operates entirely in the cloud, enabling collaborative coding and resource-intensive tasks without the need for extensive hardware specifications. While Google Colab doesn't adhere to a specific ISO/IEC standard like other applications, its usage aligns with general cybersecurity and data protection principles. Users are encouraged to follow best practices in securing their notebooks, such as restricting access and managing data privacy according to their organizational or institutional standards.
Standard Used: ISO/IEC 27001

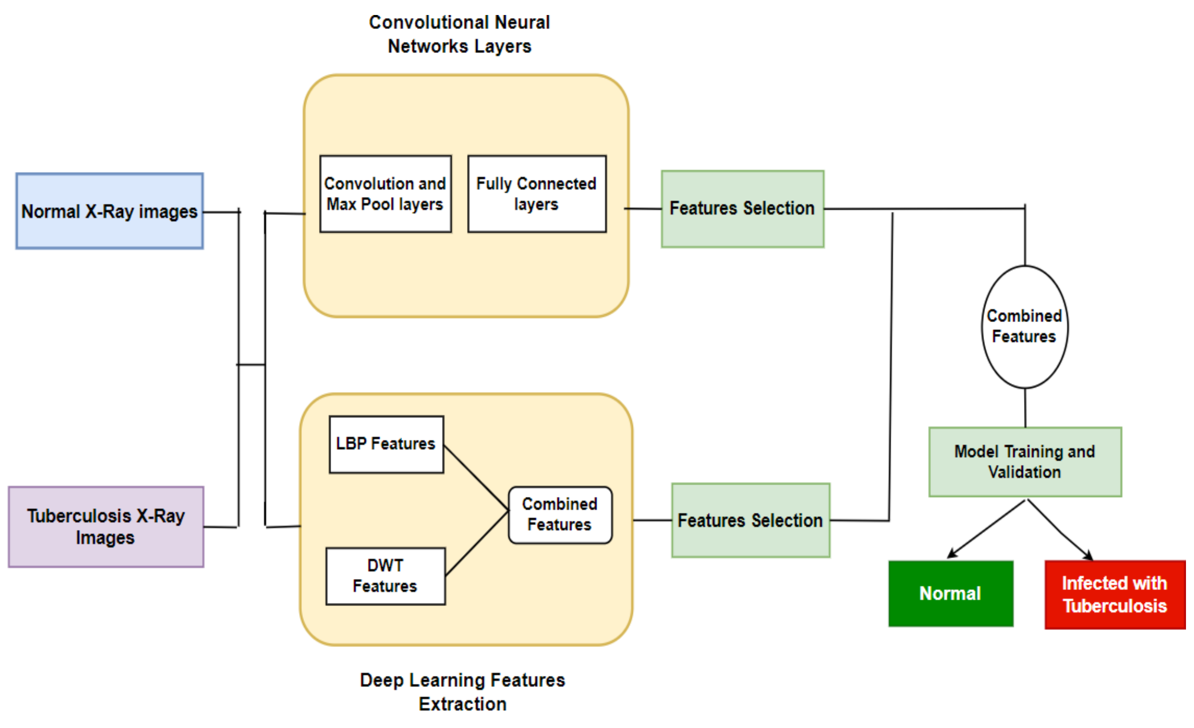
Visual Studio Code

Visual Studio Code (VS Code) is a source-code editor developed by Microsoft, known for its versatility and support for various programming languages. While it doesn't adhere to a specific ISO/IEC standard, Microsoft, the developer of Visual Studio Code, maintains a commitment to security and privacy in its software development practices. Therefore, while Visual Studio Code itself may not have a direct standard, users can benefit from Microsoft's broader adherence to industry-recognized standards for information security and data protection. It's advisable for users to implement their own security measures and adhere to relevant standards based on their specific use cases and organizational requirements.

Chapter 4

METHODOLOGY

4.1 General Architecture



Convolutional Neural Networks Layers
Normal X-Ray images
Convolution and Max Pool layers
Fully Connected layers
Features Selection
Combined Features
Model Training and Validation
Tuberculosis X-Ray Images
LBP Features
Combined Features
DWT Features
Features Selection
Normal
Infected with Tuberculosis
Deep Learning Features Extraction

Figure 4.1: **General Architecture**

The figure 4.1 shows the architecture diagram of the system. It illustrates a sophisticated system for TB detection, employing CNN within a DL framework. Input chest X-ray images undergo feature extraction and pattern recognition stages, facilitated by the CNN layers. The Convolutional Layer will identify and filter the pattrens from the input image. The Max Pool Layer will reduce the spatial dimensions of the output given by the convolutional layer. The Fully Connected Layers will get the output from the Max Pool Layer and classify the image into normal or TB infected. These layers learn intricate patterns indicative of TB. The patterns consist of discrete wavelet transformation features, logical binary patterns etc. All the combined fea- tures will be used in the prediction of input image. The output provides automated, objective, and rapid TB diagnosis, promising a transformative impact on healthcare.

4.2 Design Phase

4.2.1 Data Flow Diagram

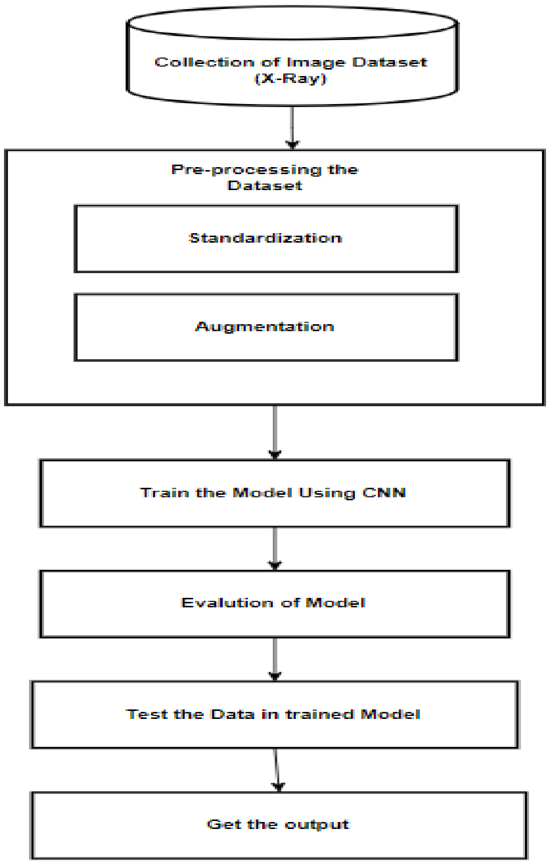

Collection of Image Dataset (X-Ray)
Pre-processing the Dataset
Standardization
Augmentation
Train the Model Using CNN
Evalution of Model
Test the Data in trained Model
Get the output

Figure 4.2: **Data Flow Diagram**

The figure 4.2 depicts the flow of data which is used in this prototype. The dataset is divided into distinct sets for comprehensive training and rigorous testing. Prior to model training, the dataset undergoes meticulous preprocessing, employing various techniques to enhance its quality and relevance. Through labeling, the data is systematically organized, paving the way for effective model training with the designated training dataset. Post-training, the model's efficacy is rigorously assessed using the testing dataset, allowing for a robust evaluation of its performance and generalization capabilities.

4.2.2 Use Case Diagram

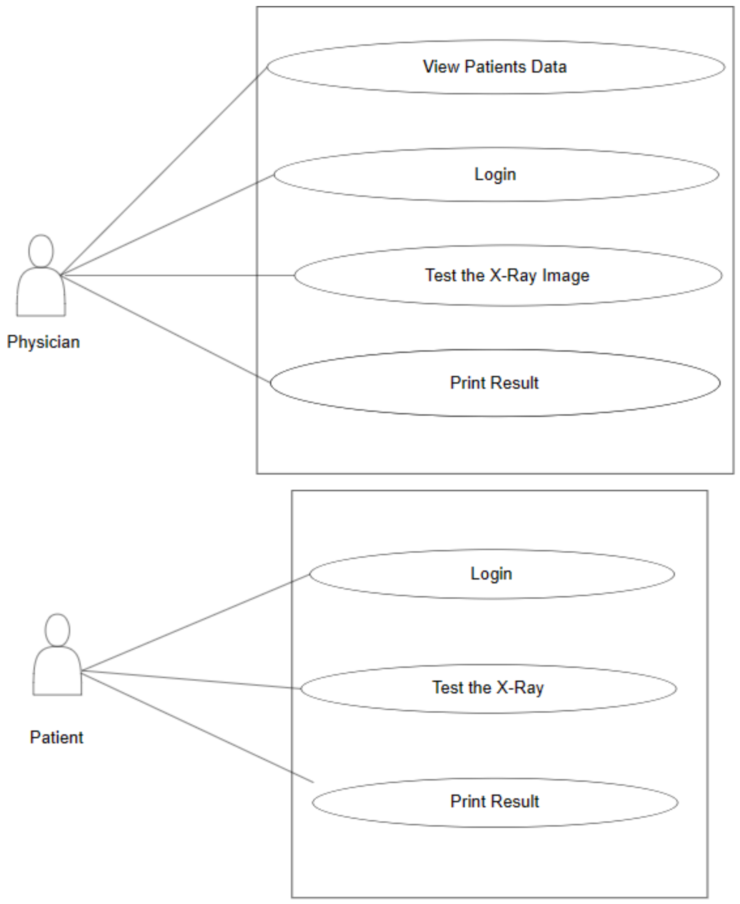


Figure 4.3: **Use Case Diagram**

The figure 4.3 displays the use-case diagram of the prototype. It depicts a user- friendly website catering to physicians and patients for tuberculosis detection through chest X-ray analysis. Physicians, as primary users, can access the site to upload patient X-ray images for automated tuberculosis screening. The system's functionalities include image processing, disease detection, and result display. This user-centric platform fosters collaboration between healthcare professionals and pa- tients, providing an intuitive and accessible interface for streamlined tuberculosis diagnosis.

4.2.3 Sequence Diagram

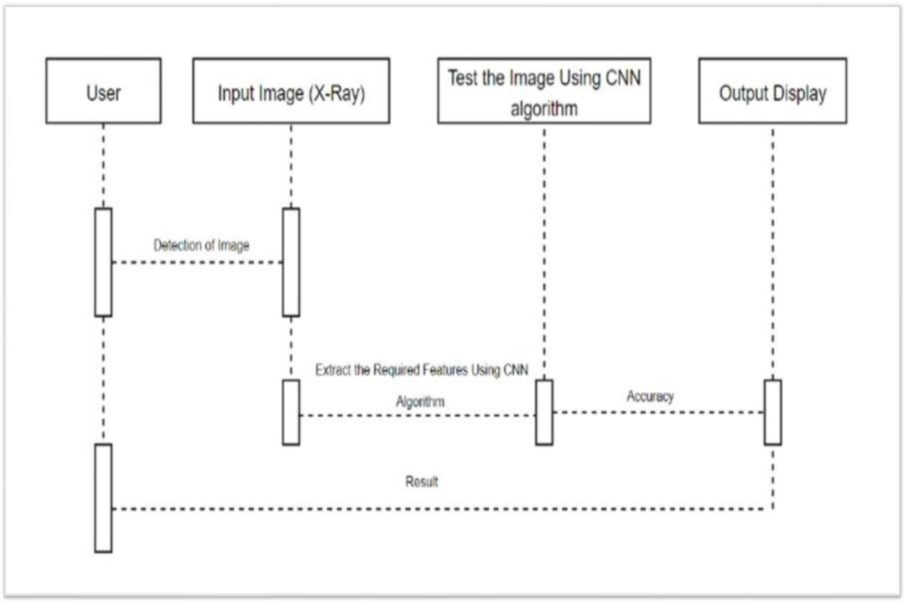

User
Input Image (X-Ray)
Test the Image Using CNN algorithm
Output Display
Detection of Image
Extract the Required Features Using CNN Algorithm
Accuracy
Result

Figure 4.4: **Sequence Diagram**

The figure 4.4 illustrates the sequence diagram of the commences with the user up-loading a chest X-ray image on the system's interface. The system uses CNN to analyze intricate patterns indicative of tuberculosis. The model computes the re-sults through its learning algorithms, determining the presence or absence of the disease. Subsequently, the system displays the outcome on the user interface, pro- viding physicians and patients with a clear and timely diagnosis. This sequential flow ensures a seamless and efficient process, combining user input, sophisticated com- putation, and result communication for an effective and user-friendly tuberculosis detection experience.

4.2.4 Activity Diagram

The figure 4.5 depicts the Activity Diagram of the prototype. It encapsulates the entire process, initiating with the user uploading a chest X-ray image. The system then engages in preprocessing techniques, optimizing the data for model training. Post-preprocessing, the model is trained on the labeled dataset, acquiring the abilityto discern tuberculosis patterns. As a user inputs an image for testing, the system activates the trained model to conduct real-time analysis. The computed results are then displayed on the interface, offering a comprehensive activity flow from user input to model computation and result display, ensuring an end-to-end, efficient, and automated tuberculosis detection process.

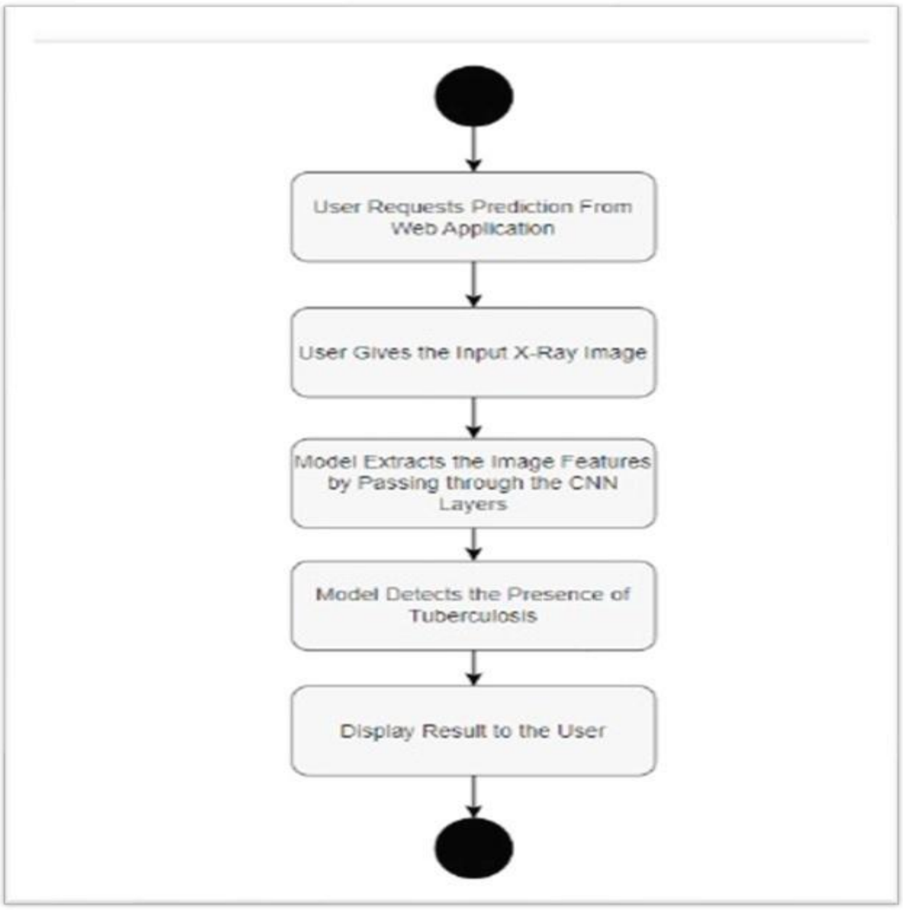

User Requests Prediction From Web Application
User Gives the Input X-Ray Image
Model Extracts the Image Features by Passing through the CNN Layers
Model Detects the Presence of Tuberculosis
Display Result to the User

Figure 4.5: **Activity Diagram**

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

1. Set constants: ‘datasetpath‘ is the path to the dataset.‘imgheight‘ and ‘imgwidth‘ are image dimensions (128x128).
2. Define a function ‘preprocessimages‘:- Parameters: ‘datasetpath‘, ‘imgheight‘, ‘imgwidth‘. Initialize empty lists ‘images‘ and ‘labels‘.
2.1. Iterate over classes in ‘datasetpath‘.For each image in a class:
2.2. Load and resize the image using OpenCV. Normalize pixel values to [0, 1]. Append preprocessed image to ‘images‘ and class label to ‘labels‘. Return NumPy arrays for ‘images‘ and ‘labels‘.
3. Call ‘preprocessimages‘ to load and preprocess images.
4. Shuffle the data using ‘shuffle‘ from scikit-learn.

5. Convert string labels to integer labels using a dictionary.
6. Split the dataset into training and testing sets.
7. Create a Sequential CNN model: Conv2D, MaxPooling2D, Flatten, Dense, Dropout layers.
8. Compile the model with Adam optimizer, binary crossentropy loss, and accuracy metric.
9. Evaluate the model on the test set.
10. Save the trained model to ’tbdetectionmodel3.h5’.

4.3.2 Pseudo Code

```

dataset_path = '/content/dataset/TB_Chest_Radiography_Database'
img_height, img_width = 128, 128

def preprocess_images(dataset_path, img_height, img_width):
images, labels = preprocess_images(dataset_path, img_height, img_width)
images, labels = shuffle(images, labels, random_state=42)
label_to_index = {label: idx for idx, label in enumerate(set(labels))}
integer_labels = np.array([label_to_index[label] for label in labels])
X_train, X_test, y_train, y_test = train_test_split(images, integer_labels, test_size=0.2,
    random_state=21)
model = Sequential()
model.compile(optimizer=Adam(lr=0.0001), loss='binary_crossentropy', metrics=['accuracy'])
datagen = ImageDataGenerator(
    rotation_range=20,
    width-shift-range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom-range=0.2,
    horizontal-flip=True,
    fill_mode='nearest'
)
history = model.fit(datagen.flow(X_train, y_train, batch_size=32), epochs=10, validation_data=(
    X_test, y_test))
test_loss, test_accuracy = model.evaluate(X_test, y_test)
model.save('tb_detection_model3.h5')
```

4.4 Module Description

4.4.1 Module- 1:- Collection of Dataset

The dataset collection process for the TB detection involves meticulous curation and selection to ensure the diversity, relevance, and quality of the data used for training and evaluation of the CNN algorithm. The primary focus was on assembling a comprehensive dataset of chest X-ray images that accurately represent the varied manifestations of tuberculosis, particularly in its early stages.

1. Inclusion Criteria: Only high-quality chest X-ray images meeting specific inclusion criteria were considered. These criteria included images with clear indications of TB, diverse demographics, and various stages of disease progression.
2. Annotation and Labeling: Each X-ray image in the dataset was meticulously annotated and labeled by qualified medical professionals. Annotations included the presence or absence of tuberculosis, the location and extent of abnormalities, and any additional relevant information.
3. Balance and Representativeness: It should achieve a balanced representation of different TB manifestations, ensuring that the dataset reflects the diversity of cases encountered in clinical practice. This helps prevent biases in the CNN’s learning process.

The curated dataset for this project comprises a total of 6,300 chest X-ray images. Specifically, there are 2,800 images representing cases of TB, and 3,500 images depicting normal, healthy conditions. The collective size of this dataset is 696 megabytes, encompassing a diverse range of chest X-ray data. This dataset will serve as the foundation for training and evaluating the CNN algorithm, ensuring a robust and comprehensive approach to the early detection of TB through medical imaging analysis. The inclusion of a substantial number of both TB affected and normal images is designed to enable the CNN model to learn and differentiatebetween pathological and healthy conditions, enhancing its diagnostic accuracy.

The dataset utilized for this project is collected from the website:
https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset?resource=download

S.No	Types of Images	Count
1.	Health X-ray Images Total	6,300
2.	Tuberculosis Affected X-ray Images	2,800
3.	Normal X-ray Images	3,500

Table 4.1: **Data Composition**

4.4.2 Module-2 :- Pre-Processing the Dataset

Data preprocessing is a crucial step in the data analysis. It involves cleaning, transforming, and organizing raw data into a format suitable for analysis or training machine learning models. The goal of data preprocessing is to improve the quality and reliability of the data, making it more suitable for downstream tasks.The various preprocessing techniques used for preprocessing the dataset involves standardization, augmentation, x-ray contrast enhancement, histogram equalization etc.

Resizing: Resizing is a crucial preprocessing technique in the development of a CNN. It involves adjusting the dimensions of the input images to a uniform size, typically a square format, to ensure consistency in the dataset. This step is essential because CNN architectures expect fixed input dimensions. Resizing not only facilitates efficient model training but also helps in reducing computational complexity. Moreover, it ensures that the model can handle input images of varying sizes during both training and inference phases.

Augmentation: Image augmentation is a technique used in the preprocessing of images to artificially increase the size of a training dataset by applying various transformations to the existing images. The purpose of image augmentation is to improve the generalization and robustness of ML models, especially in computer vision tasks. By exposing the model to a variety of augmented images, it learns tobe invariant to certain changes inthe input data.

Normalization: Normalization is a vital preprocessing technique employed in the development of CNN. The purpose of normalization is to standardize pixel values across images, ensuring that the model is less sensitive to variations in illumination and contrast. Typically, pixel values are scaled to a common range, such as [0, 1],

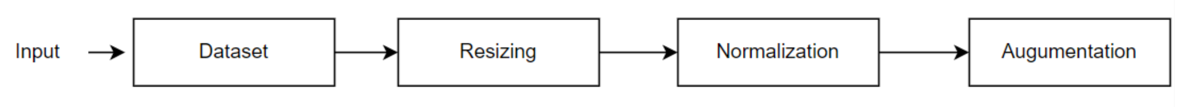


Figure 4.6: **Preprocessing the Dataset**

making it easier for the neural network to converge during training. This process aids in stabilizing the learning process by preventing the dominance of high-intensity features and mitigating issues related to varying image intensities. Normalization is particularly crucial in medical image analysis, where consistency in pixel values is essential for accurate and robust model predictions. Standardizing pixel intensities through normalization enhances the model's ability to extract meaningful patterns and improves its generalization to unseen data. By incorporating normalization as a preprocessing step, the CNN becomes more resilient to variations in image intensity, ultimately contributing to the model's effectiveness in detecting TB in X-ray images

4.4.3 Module-3 :-Training the Model

Training the CNN model for this project involves a rigorous process to impartthe network with the ability to accurately distinguish between TB affected and normal chest X-ray images. The curated dataset, comprising 6,300 images (2,800 tuberculosis-affected and 3,500 normal), served as the basis for model training. The dataset was partitioned into training, validation, and testing sets to ensure robust performance evaluation. During training, the CNN's layers were fine-tuned through backpropagation, optimizing the model's weights and biases to minimizethe classification error. Transfer learning techniques were explored, leveraging pre-trained models such as those from ImageNet to expedite the training processand enhance the model's ability to generalize across diverse datasets.

To address class imbalance, weighted loss functions were implemented, giving higher importance to the minority class of TB-affected images. The training process involved multiple epochs, allowing the CNN to iteratively learn and refine its feature representations. Regular validation on the validation set and, finally, evaluation on the test set ensured the model's proficiency in detecting TB while minimizing false positives. The trained CNN stands poised to contribute to the early diagnosis of TB

through accurate analysis of chest X-ray images.
The Algorithm used in training the model is CNN. It consist of several layers that work together to extract hierarchical features from input data. Three fundamental layers in a typical CNN architecture include:
1. **Convolutional Layer:** The convolutional layer is the cornerstone of CNNs. It applies convolutional operations to the input data using filters or kernels, whichare small learnable matrices. These filters systematically slide over the input, performing element-wise multiplications and aggregating the results. This process captures spatial hierarchies and extracts features, enabling the network to recognize patterns.
2. **Pooling Layer:** Following convolution, the pooling layer is employed to down-sample the spatial dimensions of the feature maps. Common pooling operations include max pooling, where the maximum value within a defined region is retained,or average pooling, where the average value is computed. Pooling reduces compu- tational complexity, focuses on essential information, and enhances the network's translational invariance, making it less sensitive to spatial variations.
3. **Fully Connected Layer:** The fully connected layer, also known as the dense layer, integrates the features learned by the convolutional and pooling layers. Neurons in this layer are connected to every activation in the previous layer, forming a dense matrix of weights. The fully connected layer performs high-level abstraction, capturing complex relationships between features and facilitating thefinal classification or regression output.

4.4.4 Module-4:- Testing the Model

The testing phase of the CNN model involves assessing its proficiency in accurately classifying TB-affected and normal chest X-ray images. The model was subjected to a separate and previously unseen test dataset, distinct from the training and validation sets, to evaluate its generalization capabilities. During testing, the model processes each image through its learned feature representations, producing predictions based on the acquired knowledge from the training phase. The true performance of the model was then assessed by comparing its predictions againstthe ground truth labels of the test dataset.

The output of the testing phase manifested as a set of evaluation metrics, includ- ing accuracy, precision ,recall providing a comprehensive assessment of the model’s classification performance. Precision quantified the model’s ability to correctly identify TB cases, while recall measured its effectiveness in capturing all actual positive cases. This comprehensive evaluation allowed for a thorough understanding of the model’s strengths and areas for potential improvement, ensuring its reliability in contributing to the early diagnosis of TB through chest X-ray analysis.

4.5 Steps to execute/run/implement the project

4.5.1 Step 1 - Setup Environment

- Ensure Python is installed.
- Install required packages: TensorFlow, scikit-learn, OpenCV.
- Set up a working directory for the project.

4.5.2 Step 2 - Download Dataset

- Obtain the TB Chest Radiography Database.
- Place the dataset in a folder accessible to the project.

4.5.3 Step 3 - Project Structure

- Create a structured project folder.
- Include the script and a subfolder for the dataset.

4.5.4 Step 4 - Code Script

- Write the Python script with the required code.

4.5.5 Step 5 - Execute Code

- Run the script in a Python environment.
- This loads, preprocesses, and splits the dataset.

4.5.6 Step 6 - Create and Compile Model

- Create a Convolutional Neural Network model in the script.
- Define layers such as Conv2D, MaxPooling2D, Flatten, Dense, and Dropout.
- Compile the model using the Adam optimizer, binary crossentropy loss, and accuracy metric.

4.5.7 Step 7 - Training

- Train the model.
- Specify batch size and number of epochs.

4.5.8 Step 8 - Evaluation

- Evaluate the model on the test set using the ‘evaluate‘ method.
- Obtain metrics such as test loss and accuracy.

4.5.9 Step 9 - Results and Analysis

- Display or log training and testing results, including accuracy.
- Analyze the model’s performance and identify potential areas for improvement.

4.5.10 Step 10 - Save Model

- Save the trained model to a file.
- This enables reuse and deployment without retraining.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

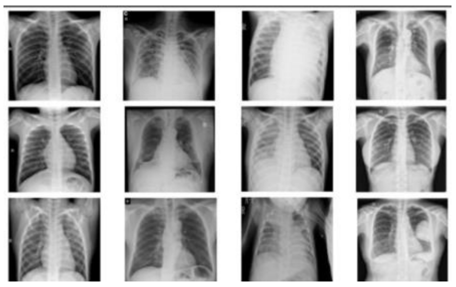
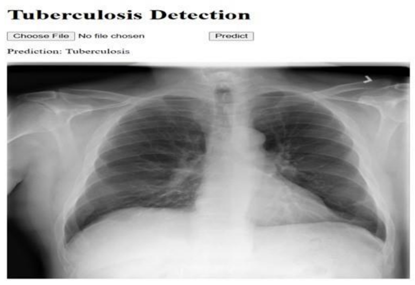


Figure 5.1: **X-Ray Images Dataset**

The above figure 5.1 illustrates the input of system which is a collection of chest X-ray images featuring varying stages of tuberculosis progression, alongside normal chest X-ray images.

5.1.2 Output Design

The figure 5.2 shows a chest X-ray for evaluation of tuberculosis presence. Analyzing distinct features reveals whether the individual is affected by TB or has a normal chest X-ray, aiding in efficient diagnosis.



Prediction: Tuberculosis

Figure 5.2: **Diagnosis Result**

5.2 Testing

The program undergoes rigorous testing to ensure its accuracy, reliability, and functionality. Various testing methodologies, including unit testing, assess indi- vidual components, while integration testing evaluates interactions between these components. Specialized testing for preprocessing techniques and machine learning models validates their effectiveness. Additionally, robustness testing examines the program's resilience to adverse conditions. Continuous testing and validation are integral to guaranteeing the program's efficacy in predicting tuberculosis from X-ray images while maintaining high standards of precision and performance.

5.3 Types of Testing

5.3.1 Unit testing

Unit testing is a software testing approach that involves the evaluation of indi- vidual units or components within a software application in isolation. These units typically refer to the smallest testable parts of the code, such as functions, meth-ods, or procedures. The primary objective of unit testing is to confirm that each unit behaves as intended, independently of the rest of the application.

Input

```
import os
import cv2
import matplotlib.pyplot as plt
import random

# Set the path to your dataset
dataset_path = '/content/dataset/TB_Chest_Radiography_Database'

# Function to pick a random image from the dataset
def pick_random_image(dataset_path):
    classes = os.listdir(dataset_path)
    class_name = random.choice(classes)
    class_path = os.path.join(dataset_path, class_name)
    image_name = random.choice(os.listdir(class_path))
    image_path = os.path.join(class_path, image_name)
    return image_path

# Get a random image path
image_path_to_test = pick_random_image(dataset_path)
```

```

# Load the original image
original_image = cv2.imread(image_path_to_test)
original_dimensions = original_image.shape[:2]

# Set the desired dimensions
desired_height, desired_width = 128, 128

# Resize the image
resized_image = cv2.resize(original_image, (desired_height, desired_width))
resized_dimensions=resized_image.shape[:2]

# Display the original and resized images
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)
plt.title(f"Original Dimensions: {original_dimensions}")
plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title(f"Resized Dimensions: {resized_dimensions}")
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.axis('off')

plt.show()
```

Test result

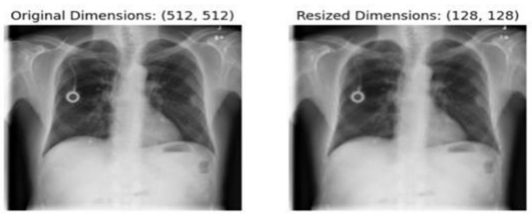

Original Dimensions: (512, 512)
Resized Dimensions: (128, 128)

Figure 5.3: **Result of Unit Test**

5.3.2 Integration testing

Integration testing is a software testing methodology that focuses on evaluating the interactions and dependencies between different components within a larger system. The goal of integration testing is to ensure that individual components, when combined, function correctly as a cohesive unit. This type of testing helps uncover errors related to data flow, control flow, and communication between modules.

Input

```
# Integration Test

# Set the image dimensions
expected_img_height, expected_img_width = 128, 128  # The expected dimensions

# Check preprocessing function
assert img_height == expected_img_height and img_width == expected_img_width, \
    f"Preprocessing dimensions do not match the expected dimensions ({expected_img_height}, {
        expected_img_width})"

# Check model input shape
assert model.layers[0].input_shape[1:3] == (expected_img_height, expected_img_width), \
    f"Model input shape does not match the expected dimensions ({expected_img_height}, {
        expected_img_width})"

print("Integration Test Passed!")
```

Test result

```
Integration Test Passed!
```

5.3.3 Functional testing

Functional testing is a crucial phase in software testing that focuses on verifying whether a software application or system performs its intended functions as specified in the requirements. It aims to ensure that the software meets the functional expectations and behaves according to the design specifications. During functional testing, the application is tested against its functional requirements, and test cases are designed to cover various scenarios, inputs, and system states.

Input

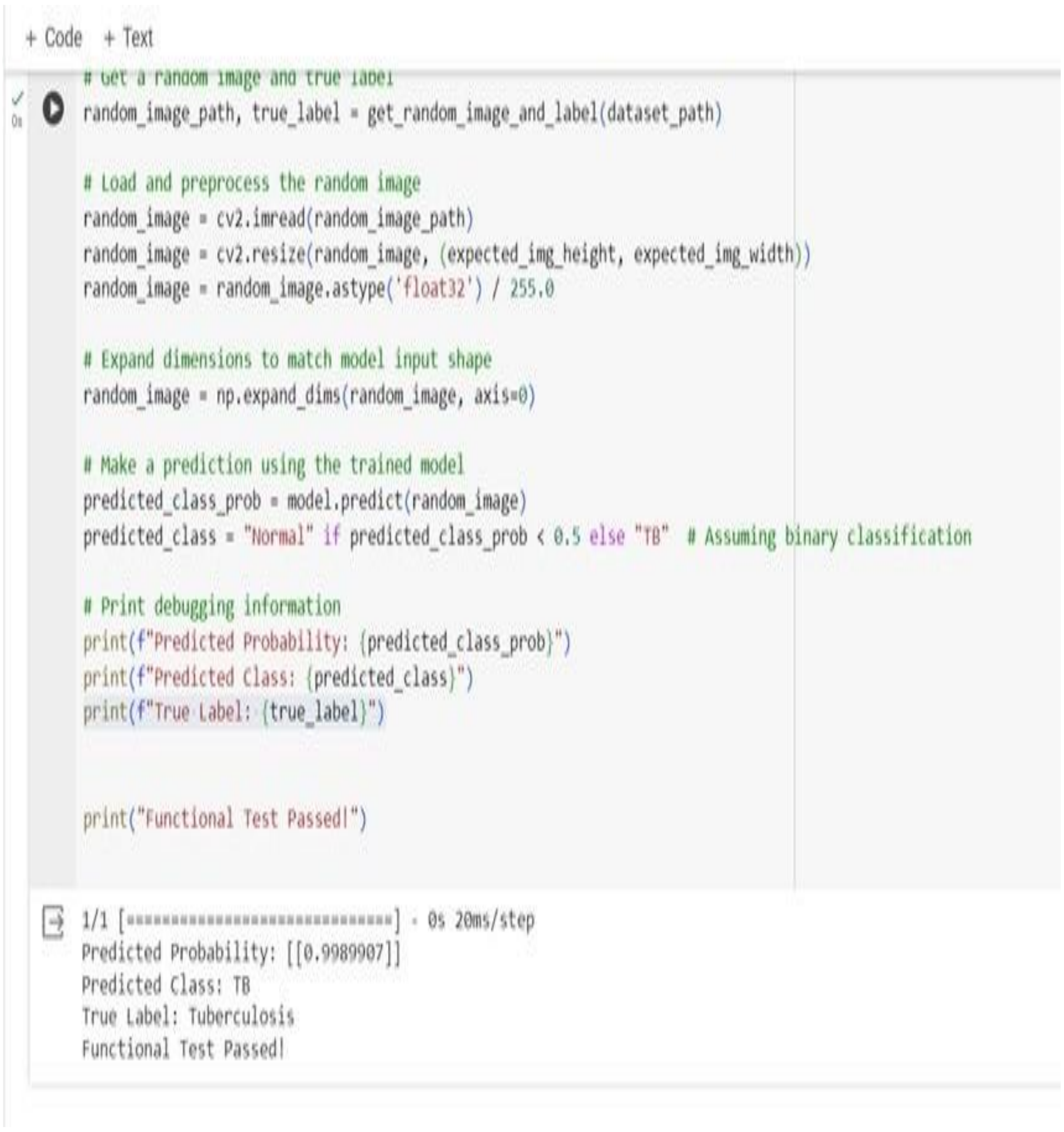
```
import os
import cv2
import matplotlib.pyplot as plt
import random

# Set the path to your dataset
dataset_path = '/content/dataset/TB_Chest_Radiography_Database'

# Revised function to get a random image path and true label from the dataset
```

```
def get_random_image_and_label(dataset_path):
    # Ensure the provided path is a directory
    if not os.path.isdir(dataset_path):
        raise NotADirectoryError(f"The provided path '{dataset_path}' is not a directory.")

    # Get a random class
    classes = os.listdir(dataset_path)
    class_name = random.choice(classes)

    # Ensure the selected class path is a directory
    class_path = os.path.join(dataset_path, class_name)
    if not os.path.isdir(class_path):
        raise NotADirectoryError(f"The path '{class_path}' is not a directory.")

    # Get a random image from the selected class
    image_name = random.choice(os.listdir(class_path))
    image_path = os.path.join(class_path, image_name)

    return image_path, class_name

# Get a random image and true label
random_image_path, true_label = get_random_image_and_label(dataset_path)

# Load and preprocess the random image
random_image = cv2.imread(random_image_path)
random_image = cv2.resize(random_image, (expected_img_height, expected_img_width))
random_image = random_image.astype('float32') / 255.0

# Expand dimensions to match model input shape
random_image = np.expand_dims(random_image, axis=0)

# Make a prediction using the trained model
predicted_class_prob = model.predict(random_image)
predicted_class = "Normal" if predicted_class_prob < 0.5 else "TB"  # Assuming binary classification

# Print debugging information
print(f"Predicted Probability: {predicted_class_prob}")
print(f"Predicted Class: {predicted_class}")
print(f"True Label: {true_label}")


print("Functional Test Passed!")
```

Test Result

```
1/1 [==============================] - 0s 20ms/step
Predicted Probability: [[0.9989907]]
Predicted Class: TB
True Label: Tuberculosis
Functional Test Passed!
```

5.3.4 Test Result



```
# Get a random image and true label
random_image_path, true_label = get_random_image_and_label(dataset_path)

# Load and preprocess the random image
random_image = cv2.imread(random_image_path)
random_image = cv2.resize(random_image, (expected_img_height, expected_img_width))
random_image = random_image.astype('float32') / 255.0

# Expand dimensions to match model input shape
random_image = np.expand_dims(random_image, axis=0)

# Make a prediction using the trained model
predicted_class_prob = model.predict(random_image)
predicted_class = "Normal" if predicted_class_prob < 0.5 else "TB"  # Assuming binary classification

# Print debugging information
print(f"Predicted Probability: {predicted_class_prob}")
print(f"Predicted Class: {predicted_class}")
print(f"True Label: {true_label}")


print("Functional Test Passed!")
```

```
1/1 [==============================] - 0s 20ms/step
Predicted Probability: [[0.9989907]]
Predicted Class: TB
True Label: Tuberculosis
Functional Test Passed!
```

Figure 5.4: **Result obtained by Testing**

The figure 5.3 illustrates the test result obtained by performing functional testing on the program.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of the proposed TB detection system lies in its ability to revolutionize and expedite the diagnostic process, addressing critical challenges associated with manual interpretation. Leveraging CNN, the system brings automation and objectivity to chest X-ray analysis, ensuring a standardized and consistent approach. The CNNs can systematically analyze vast datasets, capturing intricate patterns in- dicative of TB with remarkable accuracy, surpassing the capabilities of traditional methods. This not only enhances the speed of diagnosis but also mitigates the inher- ent subjectivity and variability associated with human interpretation.

The proposed system offers a timely response to the global urgency of TB detection, facilitating early interventions and reducing the risk of disease progression. By automating the identification of subtle manifestations, the system contributes to swifter patient triage and treatment initiation, crucial factors in improving overall health outcomes. The scalability of the system ensures its adaptability to diverse populations and healthcare settings, offering a versatile solution for regions with varying levels of expertise and resources. In essence, the efficiency of the proposed system lies in its potential to revolutionize TB diagnosis, bringing forth a new era of accuracy, speed, and accessibility in healthcare, ultimately making significant strides towards mitigating the impact of this infectious disease on a global scale.

6.2 Comparison of Existing and Proposed System

The comparison between the existing system of manual TB diagnosis and the proposed automated system underscores a paradigm shift in efficiency, accuracy, and accessibility. In the existing system, the reliance on human interpretation of chest X-rays introduces subjectivity and potential errors, leading to delays in diagnosis and treatment initiation. The manual approach, often constrained by the availability of skilled radiologists, is labor-intensive and may not be conducive to swift responses, particularly in regions with a high TB burden and limited healthcare resources.

On the contrary, the proposed system built on CNN and DL, presents a transformative leap in TB detection. By harnessing the power of AI, the proposed system automates the analysis of chest X-rays, systematically identifying patterns indicative of TB with high precision. This not only eliminates the subjective variability present in the manual approach but also accelerates the diagnostic timeline, facilitating prompt interventions and reducing the risk of disease transmission. The proposed system's efficiency is further evident in its scalability and adaptability. Unlike the existing system, the automated approach can be deployed across diverse healthcare settings, making it a valuable asset in regions with varying levels of expertise and resources. The CNNs' ability to continuously learn and adapt enhances the system's diagnostic capabilities over time, offering a dynamic solution that aligns with the evolving nature of TB manifestations.

The proposed system's integration of technology addresses the global imperative for improved healthcare access. By automating the diagnostic process, it has the po- tential to bridge the gap in expertise, especially in areas with a shortage of skilled healthcare professionals. This not only enhances the overall efficiency of TB detec- tion but also contributes to more equitable healthcare outcomes on a global scale.

The comparison between the existing manual system and the proposed automated system highlights a transformative shift from subjective, labor-intensive practices to objective, efficient, and scalable solutions. The proposed system not only enhances accuracy and speed in TB diagnosis but also holds the promise of improving accessibility and reducing disparities in healthcare delivery.

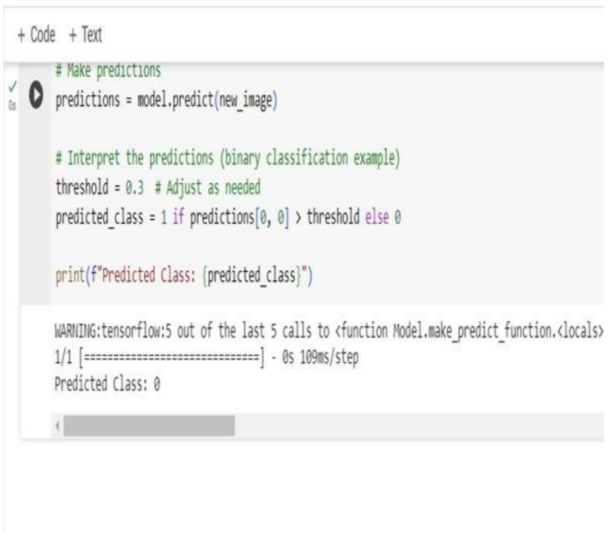
Epoch	1	2	3	4	5	6	7	8	9	10	Total
Accuracy	0.99	0.88	0.89	0.91	0.85	0.88	0.90	0.90	0.92	0.92	0.92

Table 6.1: **Epoch Comparison**

6.3 Sample Code

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
import os
import cv2
import numpy as np

# Set the path to your dataset
dataset_path = '/content/dataset/TB_Chest_Radiography_Database'

# Set the image dimensions
img_height, img_width = 128, 128  # Adjust as needed

# Function to load and preprocess images
def preprocess_images(dataset_path, img_height, img_width):
    images = []
    labels = []
# Load and preprocess the images
images, labels = preprocess_images(dataset_path, img_height, img_width)

# Shuffle the data
images, labels = shuffle(images, labels, random_state=42)



    # Continue with the rest of the code...
    # Create a CNN model
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(img_height, img_width, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    )

    # Train the model
    history = model.fit(datagen.flow(X_train, y_train, batch_size=32), epochs=10, validation_data=(
        X_test, y_test))

    # Evaluate the model
    test_loss, test_accuracy = model.evaluate(X_test, y_test)
    print(f"Test Accuracy: {test_accuracy * 100:.2f}%")

    # Save the model
    model.save('tb_detection_model3.h5')

except ValueError as e:
    print(f"Error splitting the dataset:{str(e)}")
```

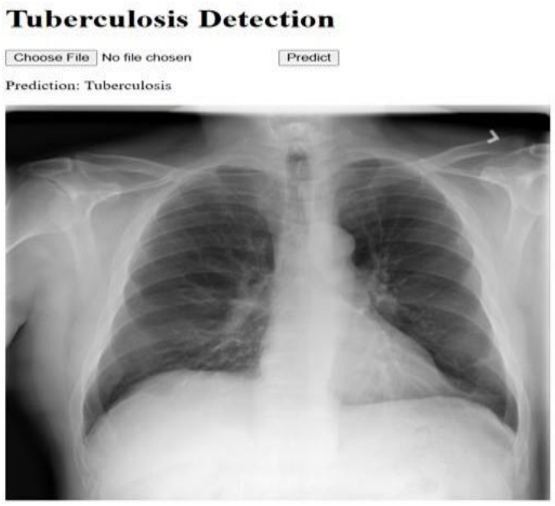
Output



```
# Make predictions
predictions = model.predict(new_image)

# Interpret the predictions (binary classification example)
threshold = 0.3  # Adjust as needed
predicted_class = 1 if predictions[0, 0] > threshold else 0

print(f"Predicted Class: {predicted_class}")
```

```
WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>
1/1 [==============================] - 0s 109ms/step
Predicted Class: 0
```

Figure 6.1: **Prediction of Tuberculosis Presence**



Tuberculosis Detection

Choose File No file chosen Predict

Prediction: Tuberculosis

Figure 6.2: **Diagnosis of Input Image through GUI**

The Figures 6.1 and 6.2 display the diagnostic outcomes for TB detection following the submission of an image.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The project represents a significant leap forward in the intersection of healthcare and artificial intelligence. The integration of CNN within the proposed system of- fers a transformative solution to the longstanding challenges associated with manual chest X-ray analysis. The project's success lies in its ability to automate and expedite the diagnostic process, providing an objective, accurate, and scalable approach to TB detection. By harnessing the power of deep learning, the system not only enhances the efficiency of diagnosis but also contributes to timely interventions, reducing the risk of disease progression and transmission. The adaptability of the model to diverse populations and healthcare settings enhances its practicality and potential global im- pact.

The incorporation of user-friendly interfaces, accessible to both physicians and patients, fosters a collaborative healthcare environment. The website's intuitive de- sign allows users to effortlessly upload X-ray images, facilitating a seamless inte- gration of technology into routine clinical practices. This project not only addresses the immediate need for improved TB diagnostics but also sets a precedent for the integration of artificial intelligence in healthcare, promising enhanced accessibility, efficiency, and accuracy. As the system undergoes real-world implementation and validation, its potential to revolutionize TB detection underscores the broader impli- cations of AI-driven solutions in shaping the future of healthcare, where innovation and technology converge for the betterment of global public health.

7.2 Future Enhancements

The future enhancements of the project are poised to further elevate its impact on healthcare. One avenue for improvement involves enhancing the model's accuracy by integrating additional convolutional layers, refining its ability to discern intricate patterns in chest X-ray images. This augmentation aims to boost diagnostic precision and sensitivity, especially in detecting subtle manifestations of TB. Moreover, continuous updates to the dataset will contribute to the model's adaptability, ensuring it remains robust against evolving disease patterns.

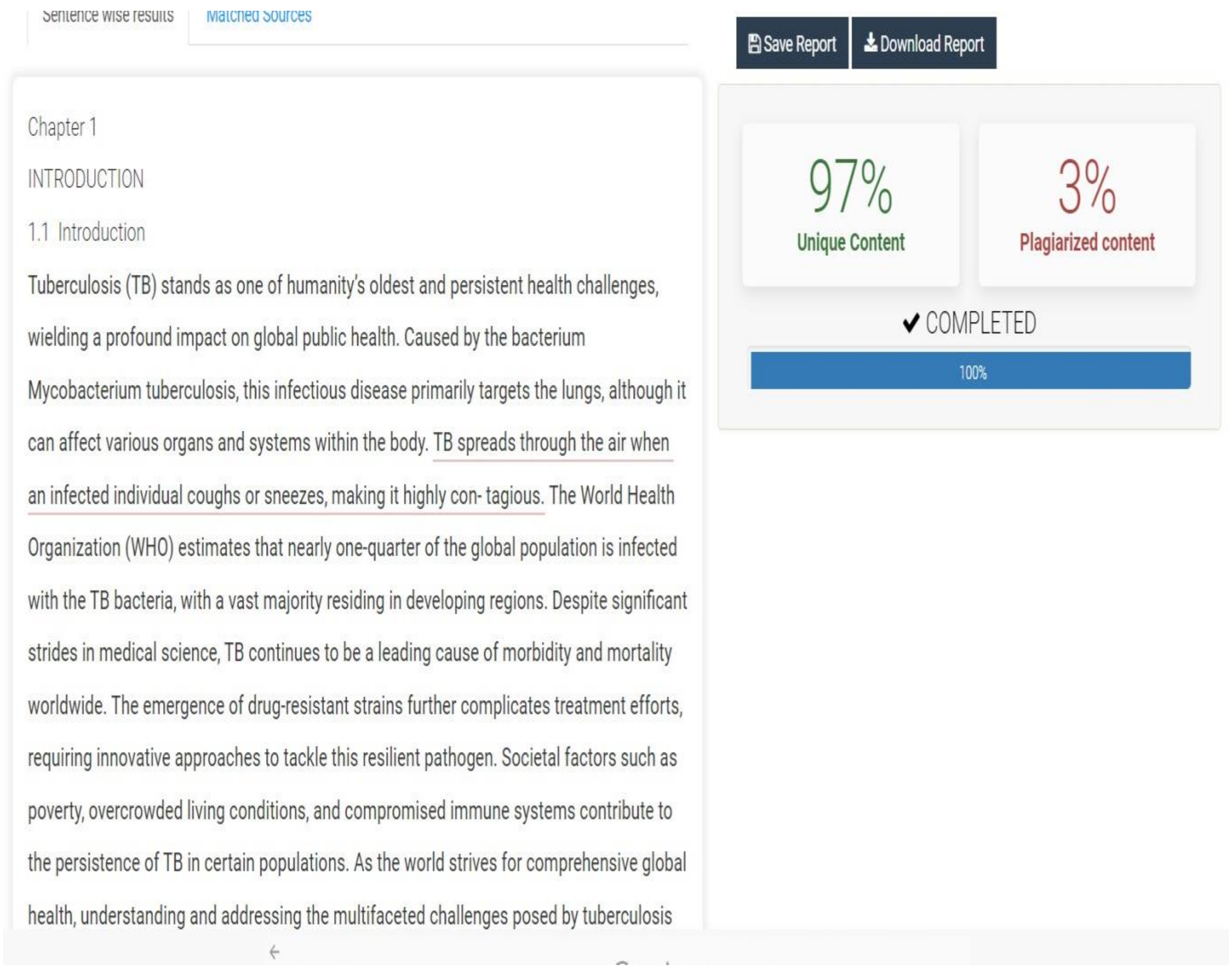
Integration with emerging technologies, such as three-dimensional imaging or multi-modal data fusion, could enhance the system's diagnostic capabilities, providing a more comprehensive understanding of TB presentations. Collaborations with healthcare institutions for real-world data integration and continuous validation will strengthen the model's reliability. Additionally, incorporating explainability fea- tures, elucidating the model's decision-making process, can enhance user trust and facilitate its integration into routine clinical workflows, marking a dynamic trajectory for the project's evolution and continued contributions to TB diagnostics.

Integration with advanced imaging modalities, such as positron emission tomography scans or magnetic resonance imaging, can provide a more comprehensive understanding of disease progression. Incorporating real-time analysis capabilities and edge computing could expedite diagnosis, particularly in urgent clinical sce- narios. A focus on interpretability and transparency through advanced visualization tools can offer insights into the model's decision-making process, fostering greater user confidence and acceptance. Continuous refinement of the model through trans- fer learning on diverse datasets from various geographic regions can enhance its generalization across populations.

Chapter 8

PLAGIARISM REPORT

8.1 Plagiarism Report



Chapter 1

INTRODUCTION

1.1 Introduction

Tuberculosis (TB) stands as one of humanity's oldest and persistent health challenges, wielding a profound impact on global public health. Caused by the bacterium Mycobacterium tuberculosis, this infectious disease primarily targets the lungs, although it can affect various organs and systems within the body. TB spreads through the air when an infected individual coughs or sneezes, making it highly con- tagious. The World Health Organization (WHO) estimates that nearly one-quarter of the global population is infected with the TB bacteria, with a vast majority residing in developing regions. Despite significant strides in medical science, TB continues to be a leading cause of morbidity and mortality worldwide. The emergence of drug-resistant strains further complicates treatment efforts, requiring innovative approaches to tackle this resilient pathogen. Societal factors such as poverty, overcrowded living conditions, and compromised immune systems contribute to the persistence of TB in certain populations. As the world strives for comprehensive global health, understanding and addressing the multifaceted challenges posed by tuberculosis

Figure 8.1: **Plagarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

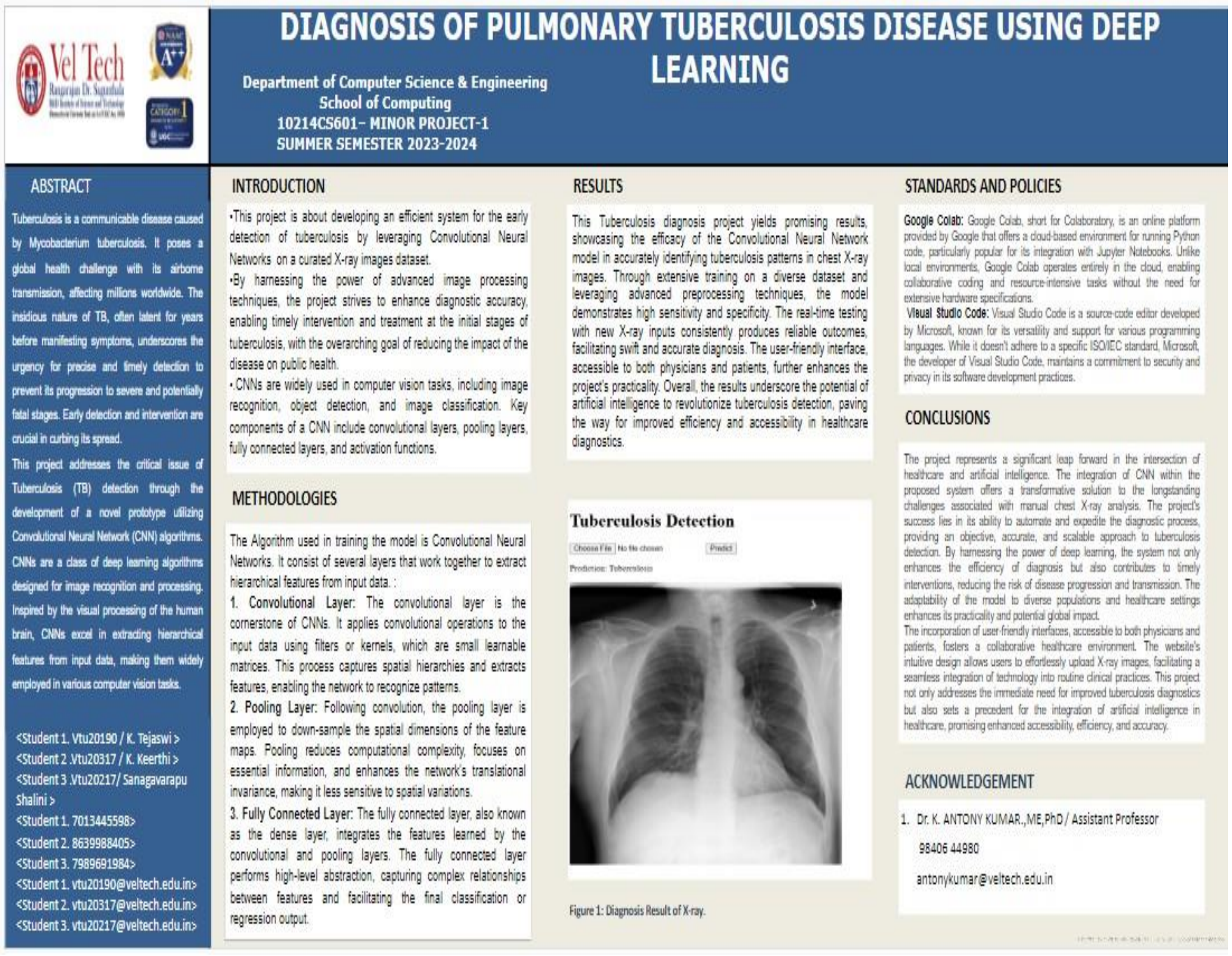
9.1 Source Code

```
from google.colab import drive
drive.mount('/content/drive')
# Import necessary libraries
from google.colab import drive
import zipfile
import os

# Mount Google Drive
drive.mount('/content/drive')

# Replace 'YOUR_FOLDER_ID' with the actual folder ID of your dataset
folder_id='1DoyAcZj3ZvurJag4t2MrLv3VSFjUHq F_'

# Set the path for your dataset zip file on Google Drive
dataset_zip_path = '/content/drive/MyDrive/Tbdataset/archive.zip'

# Specify the destination directory to extract the datasete
xtracted_path='/content/dataset/'

# Create the destination directory if it doesn't exist
os.makedirs(extracted_path,exist_ok=True)

# Extract the dataset from the zip file
with zipfile.ZipFile(dataset zip path, 'r') as zip_ref:z
      ipref.extractall(extractedpath)

# Check the contents of the extracted directory
print("Dataset has been uploaded and extracted to:", extracted path)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image importImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
```

```python
import os
import cv2
import numpy as np

# Set the path to your dataset
dataset_path = '/content/dataset/TB_Chest_Radiography_Database'

# Set the image dimensions
img_height, img_width = 128, 128  # Adjust as needed

# Function to load and preprocess images
def preprocess_images(dataset_path, img_height, img_width):
    images = []
    labels = []

    # Assuming subdirectories in the dataset path represent different classes
    classes = os.listdir(dataset_path)

    for class_name in classes:
        class_path = os.path.join(dataset_path, class_name)
        if os.path.isdir(class_path):
            for image_name in os.listdir(class_path):
                image_path = os.path.join(class_path, image_name)

                # Check if the file is an image
                if image_name.lower().endswith(('.png', '.jpg', '.jpeg')):
                    try:
                        # Load and preprocess each image
                        image = cv2.imread(image_path)
                        image = cv2.resize(image, (img_height, img_width))
                        image = image.astype('float32') / 255.0  # Normalize pixel values
                        images.append(image)
                        labels.append(class_name)
                    except Exception as e:
                        print(f"Error processing image {image_path}: {str(e)}")

    return np.array(images), np.array(labels)

# Load and preprocess the images
images, labels = preprocess_images(dataset_path, img_height, img_width)

# Shuffle the data
images, labels = shuffle(images, labels, random_state=42)

# Convert string labels to integer labels
label_to_index = {label: idx for idx, label in enumerate(set(labels))}
integer_labels = np.array([label_to_index[label] for label in labels])

# Split the dataset into training and testing sets
try:
    X_train, X_test, y_train, y_test = train_test_split(images, integer_labels, test_size=0.2, random_state=21)

    # Continue with the rest of the code...
    # Create a CNN model
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(img_height, img_width, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(128, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation='sigmoid'))  # Binary classification

    # Compile the model
    model.compile(optimizer=Adam(lr=0.0001), loss='binary_crossentropy', metrics=['accuracy'])

    # Data augmentation
    datagen = ImageDataGenerator(
        rotation_range=20,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest'
    )

    # Train the model
    history = model.fit(datagen.flow(X_train, y_train, batch_size=32), epochs=10, validation_data=(X_test, y_test))

    # Evaluate the model
    test_loss, test_accuracy = model.evaluate(X_test, y_test)
    print(f"Test Accuracy: {test_accuracy * 100:.2f}%")

    # Save the model
    model.save('tb_detection_model3.h5')

except ValueError as e:
    print(f"Error splitting the dataset: {str(e)}")

    from tensorflow.keras.models import load_model
import cv2
import numpy as np

# Load the trained model
```

```
model = load_model('tb_detection_model3.h5')

# Function to preprocess a new image
defpreprocessimage(image_path, img_height, img_width):
    image = cv2.imread(image_path)
    image = cv2.resize(image, (img_height, img_width))
    image = image.astype('float32') / 255.0
    image = np.expand_dims(image, axis=0)  # Add batch dimension
    return image

# Path to the new image you want to test
new_image_path = '/content/Tuberculosis-7.png'

# Preprocess the new image
new_image = preprocess_image(new_image_path, img_height, img_width)

# Make predictions
predictions = model.predict(new_image)

# Interpret the predictions (binary classification example)
threshold = 0.3  # Adjust as needed
predicted_class = 1 if predictions[0, 0] > threshold else 0

print(f"Predicted Class: {predicted_class}")
```

9.2 Poster Presentation



DIAGNOSIS OF PULMONARY TUBERCULOSIS DISEASE USING DEEP LEARNING

Department of Computer Science & Engineering
School of Computing
10214CS601– MINOR PROJECT-1
SUMMER SEMESTER 2023-2024

ABSTRACT

Tuberculosis is a communicable disease caused by Mycobacterium tuberculosis. It poses a global health challenge with its airborne transmission, affecting millions worldwide. The insidious nature of TB, often latent for years before manifesting symptoms, underscores the urgency for precise and timely detection to prevent its progression to severe and potentially fatal stages. Early detection and intervention are crucial in curbing its spread.

This project addresses the critical issue of Tuberculosis (TB) detection through the development of a novel prototype utilizing Convolutional Neural Network (CNN) algorithms. CNNs are a class of deep learning algorithms designed for image recognition and processing. Inspired by the visual processing of the human brain, CNNs excel in extracting hierarchical features from input data, making them widely employed in various computer vision tasks.

<Student 1. Vtu20190 / K. Tejaswi >
<Student 2 .Vtu20317 / K. Keerthi >
<Student 3 .Vtu20217/ Sanagavarapu Shalini >
<Student 1. 7013445598>
<Student 2. 8639988405>
<Student 3. 7989691984>
<Student 1. vtu20190@veltech.edu.in>
<Student 2. vtu20317@veltech.edu.in>
<Student 3. vtu20217@veltech.edu.in>

INTRODUCTION

•This project is about developing an efficient system for the early detection of tuberculosis by leveraging Convolutional Neural Networks on a curated X-ray images dataset.
•By harnessing the power of advanced image processing techniques, the project strives to enhance diagnostic accuracy, enabling timely intervention and treatment at the initial stages of tuberculosis, with the overarching goal of reducing the impact of the disease on public health.
•.CNNs are widely used in computer vision tasks, including image recognition, object detection, and image classification. Key components of a CNN include convolutional layers, pooling layers, fully connected layers, and activation functions.

METHODOLOGIES

The Algorithm used in training the model is Convolutional Neural Networks. It consist of several layers that work together to extract hierarchical features from input data. :
1. **Convolutional Layer:** The convolutional layer is the cornerstone of CNNs. It applies convolutional operations to the input data using filters or kernels, which are small learnable matrices. This process captures spatial hierarchies and extracts features, enabling the network to recognize patterns.
2. **Pooling Layer:** Following convolution, the pooling layer is employed to down-sample the spatial dimensions of the feature maps. Pooling reduces computational complexity, focuses on essential information, and enhances the network's translational invariance, making it less sensitive to spatial variations.
3. **Fully Connected Layer:** The fully connected layer, also known as the dense layer, integrates the features learned by the convolutional and pooling layers. The fully connected layer performs high-level abstraction, capturing complex relationships between features and facilitating the final classification or regression output.

RESULTS

This Tuberculosis diagnosis project yields promising results, showcasing the efficacy of the Convolutional Neural Network model in accurately identifying tuberculosis patterns in chest X-ray images. Through extensive training on a diverse dataset and leveraging advanced preprocessing techniques, the model demonstrates high sensitivity and specificity. The real-time testing with new X-ray inputs consistently produces reliable outcomes, facilitating swift and accurate diagnosis. The user-friendly interface, accessible to both physicians and patients, further enhances the project's practicality. Overall, the results underscore the potential of artificial intelligence to revolutionize tuberculosis detection, paving the way for improved efficiency and accessibility in healthcare diagnostics.

Tuberculosis Detection

Choose File No file chosen Predict

Prediction: Tuberculosis

Figure 1: Diagnosis Result of X-ray.

STANDARDS AND POLICIES

Google Colab: Google Colab, short for Colaboratory, is an online platform provided by Google that offers a cloud-based environment for running Python code, particularly popular for its integration with Jupyter Notebooks. Unlike local environments, Google Colab operates entirely in the cloud, enabling collaborative coding and resource-intensive tasks without the need for extensive hardware specifications.
Visual Studio Code: Visual Studio Code is a source-code editor developed by Microsoft, known for its versatility and support for various programming languages. While it doesn't adhere to a specific ISO/IEC standard, Microsoft, the developer of Visual Studio Code, maintains a commitment to security and privacy in its software development practices.

CONCLUSIONS

The project represents a significant leap forward in the intersection of healthcare and artificial intelligence. The integration of CNN within the proposed system offers a transformative solution to the longstanding challenges associated with manual chest X-ray analysis. The project's success lies in its ability to automate and expedite the diagnostic process, providing an objective, accurate, and scalable approach to tuberculosis detection. By harnessing the power of deep learning, the system not only enhances the efficiency of diagnosis but also contributes to timely interventions, reducing the risk of disease progression and transmission. The adaptability of the model to diverse populations and healthcare settings enhances its practicality and potential global impact.
The incorporation of user-friendly interfaces, accessible to both physicians and patients, fosters a collaborative healthcare environment. The website's intuitive design allows users to effortlessly upload X-ray images, facilitating a seamless integration of technology into routine clinical practices. This project not only addresses the immediate need for improved tuberculosis diagnostics but also sets a precedent for the integration of artificial intelligence in healthcare, promising enhanced accessibility, efficiency, and accuracy.

ACKNOWLEDGEMENT

1. Dr. K. ANTONY KUMAR.,ME,PhD / Assistant Professor
98406 44980
antonykumar@veltech.edu.in

Figure 9.1: **Poster**

References

[1] Rahman, T., Khandakar, A., Kadir, M.A., Islam, K.R., Islam, K.F., Mazhar, R., Hamid, T., Islam, M.T., Kashem, S., Mahbub, Z.B. and Ayari, M.A., 2020. Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access, 8, pp.191586-191601.

[2] Hooda, R., Sofat, S., Kaur, S., Mittal, A. and Meriaudeau, F., 2017, September. Deep-learning: A potential method for tuberculosis detection using chest radiography. In 2017 IEEE international conference on signal and image processing applications (ICSIPA) (pp. 497-502). IEEE.

[3] Kant, S. and Srivastava, M.M., 2018, November. Towards automated tuberculosis detection using deep learning. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1250-1253). IEEE.

[4] Yadav, O., Passi, K. and Jain, C.K., 2018, December. Using deep learning to classify X-ray images of potential tuberculosis patients. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 2368-2375). IEEE.

[5] Rajaraman, S. and Antani, S.K., 2020. Modality-specific deep learning model ensembles toward improving TB detection in chest radiographs. IEEE Access, 8, pp.27318-27326.

[6] Liu, C., Cao, Y., Alcantara, M., Liu, B., Brunette, M., Peinado, J. and Curioso, W., 2017, September. TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network. In 2017 IEEE international conference on image processing (ICIP) (pp. 2314-2318). IEEE.

[7] Nguyen, Q.H., Nguyen, B.P., Dao, S.D., Unnikrishnan, B., Dhingra, R., Ravichandran, S.R., Satpathy, S., Raja, P.N. and Chua, M.C., 2019, April. Deep learning models for tuberculosis detection from chest X-ray images. In 2019 26th international conference on telecommunications (ICT) (pp. 381-385). IEEE.

[8] Stirenko, S., Kochura, Y., Alienin, O., Rokovyi, O., Gordienko, Y., Gang, P. and Zeng, W., 2018, April. Chest X-ray analysis of tuberculosis by deep learning with segmentation and augmentation. In 2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO) (pp. 422-428). IEEE.

[9] Ho, T.K.K., Gwak, J., Prakash, O., Song, J.I. and Park, C.M., 2019. Utilizing pretrained deep learning models for automated pulmonary tuberculosis detection using chest radiography. In Intelligent Information and Database Systems: 11th Asian Conference, ACIIDS 2019, Yogyakarta, Indonesia, April 8–11, 2019, Proceedings, Part II 11 (pp. 395-403). Springer International Publishing.

[10] Norval, M., Wang, Z. and Sun, Y., 2019, December. Pulmonary tuberculosis detection using deep learning convolutional neural networks. In Proceedings of the 3rd International Conference on Video and Image Processing (pp. 47-51).