SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University Anantapur)
Sree Sainath Nagar, A. Rangampet, Tirupati – 517 102, Chittoor Dist., A.P.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Work entitled

"Enhancing Lung Nodule Detection: A Vision Transformer-Based Approach with Avian Optimization in Chest CT Imaging "

is the bonafide work done by

GUDIPATI VAMSI KRISHNA	**(20121A05A4)**
GUNDA UDAY MANISH	**(20121A05A6)**
KODATHALA BHAVANA REDDY	**(20121A05D9)**
KUCHI VAISHNAVI	**(20121A05F0)**

In the Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, A. Rangampet. is affiliated to JNTUA, Anantapuramu in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2020-2024.

This is work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for the award of any degree or diploma.

Internal Guide	**Head**
Dr.J.Avanija	**Dr. B. Narendra Kumar Rao**
Professor	Prof & Head
Dept of CSE	Dept of CSE
Sree Vidyanikethan Engineering College	Sree Vidyanikethan Engineering College
Tirupathi	Tirupathi

INTERNAL EXAMINER **EXTERNAL EXAMINER**

“Enhancing Lung Nodule Detection: A Vision Transformer-Based Approach with Avian Optimization in Chest CT Imaging”

A Project Report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNVERSITY ANANTAPUR.

In Partial Fulfillment of the Requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
BY

GUDIPATI VAMSI KRISHNA (20121A05A4)
GUNDA UDAY MANISH (20121A05A6)
KODATHALA BHAVANA REDDY (20121A05D9)
KUCHI VAISHNAVI (20121A05F0)

Under the Guidance of
Dr. J. Avanija
Professor
Dept of CSE, SVEC

Department of Computer Science and Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(Affiliated to JNTUA, Anantapuramu)
Sree Sainath Nagar, Tirupathi – 517 102
2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION AND MISSION

VISION

To become a Centre of Excellence in Computer Science and Engineering by imparting high quality education through teaching, training and research.

MISSION

- The Department of Computer Science and Engineering is established to provide undergraduate and graduate education in the field of Computer Science and Engineering to students with diverse background in foundations of software and hardware through a broad curriculum and strongly focused on developing advanced knowledge to become future leaders.
- Create knowledge of advanced concepts, innovative technologies and develop research aptitude for contributing to the needs of industry and society.
- Develop professional and soft skills for improved knowledge and employability of students.
- Encourage students to engage in life-long learning to create awareness of the contemporary developments in computer science and engineering to become outstanding professionals.
- Develop attitude for ethical and social responsibilities in professional practice at regional, National and International levels.

Program Educational Objectives (PEO's)

After few years of graduation, the graduates of B.Tech. (CSE) will be:

1. Pursuing higher studies in Computer Science and Engineering and related disciplines

2. Employed in reputed Computer and I.T organizations and Government or have established startup companies.

3. Able to demonstrate effective communication, engage in team work, exhibit leadership skills, ethical attitude, and achieve professional advancement through continuing education.

Program Specific Outcomes (PSO's)

On successful completion of the Program, the graduates of B.Tech. (CSE) program will be able to:

PSO1: Use mathematical methodologies to model real-world problems, Employ modern tools and platforms for efficient design and development of computer-based systems.

PSO2: Apply adaptive algorithms and methodologies to develop intelligent systems for solving problems from interdisciplinary domains.

PSO3: Apply suitable models, tools, and techniques to perform data analytics for effective decision-making.

PSO4: Design and deploy networked systems using standards and principles, evaluate security measures for complex networks, and apply procedures and tools to solve networking issues.

Program Outcomes (PO's)

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering knowledge**).

2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem analysis**).

3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/development of solutions**).

4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct investigations of complex problems**).

5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern tool usage**)

6. Apply to reason informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The engineer and society**)

7. Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and sustainability**).

8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).

9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and team work**).

10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).

11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project management and finance**).

12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

Course Outcomes

CO1. Knowledge on the project topic (PO1)

CO2. Analytical ability exercised in the project work.(PO2)

CO3. Design skills applied on the project topic. (PO3)

CO4. Ability to investigate and solve complex engineering problems faced during the project work. (PO4)

CO5. Ability to apply tools and techniques to complex engineering activities with an understanding of limitations in the project work. (PO5)

CO6. Ability to provide solutions as per societal needs with consideration to health, safety, legal and cultural issues considered in the project work. (PO6)

CO7. Understanding of the impact of the professional engineering solutions in environmental context and need for sustainable development experienced during the project work. (PO7)

CO8. Ability to apply ethics and norms of the engineering practice as applied in the project work.(PO8)

CO9. Ability to function effectively as an individual as experienced during the project work. (PO9)

CO10. Ability to present views cogently and precisely on the project work. (PO10)

CO11. Project management skills as applied in the project work. (PO11)

CO12. Ability to engage in life-long leaning as experience during the project work. (PO12)

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3												3			
CO2		3												3		
CO3			3												3	
CO4				3											3	
CO5					3											3
CO6						3										
CO7							3									
CO8								3								
CO9									3							
CO10										3						
CO11											3					
CO12												3				

(Note: 3-High, 2-Medium, 1-Low)

DECLARATION

We hereby declare that this project report titled **"Enhancing Lung Nodule Detection: A Vision Transformer-Based Approach with Avian Optimization in Chest CT Imaging"** is a genuine project work carriedout by us, in **B.Tech *(Computer Science and Engineering)*** degree courseof **Jawaharlal Nehru Technological University Anantapur** and has not been submitted to any other course or University for the award of any degree by us.

Signature of the student

1.

2.

3.

4.

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

We are highly indebted to **Dr. B. M. Satish**, Principal of Sree Vidyanikethan Engineering College for his valuable support and guidance in all academic matters.

We are very much obliged to **Dr. B. Narendra Kumar Rao,** Professor& Head, Department of CSE, for providing us the guidance and encouragement in completion of this project.

We would like to express our indebtedness to the project coordinator, **Dr. G. Vennila**, Associate Professor, Department of CSE for his valuable guidance during the course of project work.

We would like to express our deep sense of gratitude to **Dr. J. Avanija**, Professor, Department of CSE, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of CSE Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in completion of our project work.

ABSTRACT

The timely and accurate identification of lung nodules in chest CT images is essential for the diagnosis of lung diseases, with a particular emphasis on cancer. This study introduces an innovative approach that utilizes Vision Transformer (ViT) networks for the three-dimensional analysis of chest CT scans. To enhance training efficiency, the Avian optimization algorithm is applied.The model is rigorously evaluated using benchmark datasets, demonstrating its proficiency in precisely detecting and localizing lung nodules. The emphasis is on achieving a balanced performance between sensitivity and specificity, crucial metrics for effective diagnostic tools.

This research highlights the potential of ViT networks for complex lung nodule diagnosis, making significant improvements to the field of medical image analysis. Furthermore, the efficiency of Avian optimization strengthens the model, paving the way for advanced early diagnosis of lung diseases through cutting-edge computational methodologies.

Keywords: Lung nodule detection, Deep Learning, Vision Transformer networks, CT image analysis, Avian optimization, Computer-aided diagnosis.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 Introduction:

The small, circular growths known as lung nodules can be seen on computed tomography or X-ray images. The difference between nodules that are less than three millimeters and masses or nodules that are more than three millimeters may be made mostly based on their size. When malignancy is being considered, getting the correct diagnosis is important to creating a treatment plan that works. Rapid and precise nodule diagnosis is made possible by modern imaging and diagnostic methods, including advanced machine learning algorithms.

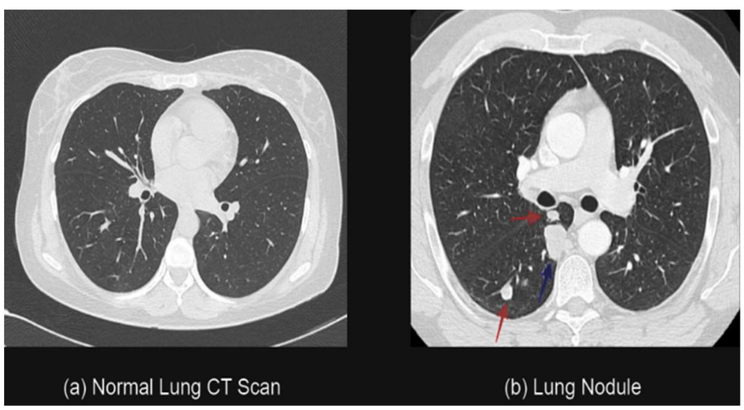


Fig -1.1 Dataset images of Lung Nodules

The study of the application is deep-learning, to identify lung nodules from medical photos such as CT scans and MRIs using effective networks such as the CNN and Vision Transformers. These models provide early detection of lung cancers with better diagnoses and treatment methods. Modern methods including data enrichment and improvement techniques are being implemented in researchers to improve the efficiency of such deep learning methods of accurate and efficient lung nodule detection.

There are a number of applications for deep learning which cover lung nodule detection. This is the case in most of the systems that score CT scans and other medical images. Imaging data with new neural networks. Convolutional Neural Networks, called CNN and more recent innovations such as Vision Transformers are significant approaches. These methods focus on the accurate identification of lung nodules in order to help in early identification and treatment planning. Researchers regularly use Modern methods such as Data augmentation techniques, transfer learning and optimization approaches in better the performance of deep learning models for accuracy and reliable lung nodule identification.

Two elements, which have different reactions in lung nodule detection, CNNs and ViTs. Because of their focus processes, ViTs are the best at searching challenging patterns in CT scans that promote accuracy. CNNs are usually known to be capable of creating characteristics and remain essential in the recognition of local structures. The ideal approach for discovering lung nodules could involve combining the characteristics of both models and blending ViTs' global knowledge with CNN's local feature recognition to get exceptionally accurate results.

1.2 Statement of the problem:

Lung diseases, notably cancer, remain a leading cause of mortality worldwide, with early detection being crucial for effective treatment and improved survival rates. Traditional methods for detecting lung nodules in chest CT scans face challenges in sensitivity and specificity, often resulting in false positives or negatives that can delay diagnosis and treatment. Moreover, the sheer volume of data in CT scans and the subtle variability of nodules require sophisticated analytical techniques. Current deep learning models, while promising, still struggle with the complexity of the task and the computational resources required for training and analysis. There is a pressing need for innovative approaches that can accurately detect and localize lung nodules in a timely and efficient manner, overcoming the limitations of existing methods and significantly advancing the field of medical image analysis.

1.3 Objectives:

The primary objective of this project is to develop and evaluate a novel diagnostic tool based on Vision Transformer (ViT) networks for the accurate and efficient detection and localization of lung nodules in chest CT scans. By incorporating three-dimensional analysis capabilities and optimizing the training process with the Avian optimization algorithm, this project aims to achieve a balanced performance between sensitivity and specificity, two crucial metrics for the effective diagnosis of lung diseases, including cancer. Through this innovative approach, the project seeks to significantly improve the field of medical image analysis, contributing to the advancement of computer-aided diagnosis systems and ultimately facilitating the early diagnosis and treatment of lung diseases. This research endeavors to harness the potential of deep learning and optimization algorithms to address the challenges posed by the complexity of lung nodule detection, paving the way for future developments in precision medicine and healthcare technology.

1.4 Scope:

The scope of this project encompasses the development and evaluation of a cutting-edge computational model designed for the detection and localization of lung nodules in chest CT scans. This includes:

- **Model Development:** Designing and implementing a Vision Transformer (ViT) network that is specifically adapted for analyzing three-dimensional chest CT scans. This involves customizing the network architecture to effectively process and interpret the complex data inherent in these scans.

- **Optimization:** Employing the Avian optimization algorithm to enhance the training efficiency of the ViT network. This includes optimizing model parameters to achieve the best possible balance between sensitivity (true positive rate) and specificity (true negative rate), which are critical for minimizing false diagnoses.

- **Data Analysis:** Utilizing benchmark datasets for lung nodule detection to train and rigorously evaluate the model. This ensures that the model is tested against a diverse array of nodule types and sizes, reflective of real-world variability.

- **Performance Evaluation:** Conducting comprehensive analyses to assess the model's accuracy, efficiency, and diagnostic performance compared to existing methodologies. This includes evaluating the model's ability to generalize across different datasets and its performance in real-world clinical settings.

- **Impact Assessment:** Exploring the potential implications of the model's deployment in healthcare settings, including its impact on diagnostic workflows, patient outcomes, and the broader field of medical image analysis.

This project aims not only to advance the technological frontier in lung nodule detection but also to contribute meaningful insights and tools to the medical community, facilitating earlier and more accurate diagnoses of lung diseases.

1.5 Application:

The integration of vision transformer-based methodology with avian optimization in chest CT imaging represents a groundbreaking approach with immense potential for revolutionizing lung nodule detection in medical diagnostics. By harnessing the power of vision transformers to discern intricate spatial relationships within images, coupled with the efficiency of avian optimization inspired by natural foraging behavior, this novel approach enables the precise identification of even subtle nodules in chest CT scans. Beyond mere detection, its implications span across various critical domains: from facilitating early lung cancer detection and tailoring personalized treatment plans, to minimizing false positives and streamlining radiology workflows.

Furthermore, its application extends to enhancing the effectiveness of screening programs in high-risk populations, ultimately contributing to improved patient outcomes and advancing the frontiers of medical imaging research.

1.6 Limitations:

While the vision transformer-based approach with avian optimization holds promise for enhancing lung nodule detection in chest CT imaging, several critical limitations and challenges must be addressed. Firstly, the quality and quantity of training data, particularly well-annotated chest CT scans with accurately labeled nodules, pose significant hurdles for model performance and generalization. Secondly, the interpretability of deep learning models like vision transformers remains a challenge, necessitating the development of interpretable AI techniques for transparent decision-making in medical settings. Additionally, ensuring the robustness and generalizability of the approach across diverse populations and clinical settings, while balancing sensitivity and specificity to minimize false positives and negatives, is imperative. Moreover, regulatory compliance, ethical considerations, and seamless integration into clinical workflows are essential for the successful deployment and adoption of AI-based diagnostic systems. Addressing these challenges requires collaborative efforts among researchers, healthcare providers, policymakers, and industry stakeholders to develop reliable, ethically sound solutions for lung nodule detection in chest CT imaging.

CHAPTER-2

LITERATURE SURVEY

Lung nodules are any of a number of abnormalities that are characterized by small, spherically shaped, round opacities that are limited to abnormal tissue. Some abnormalities in the lungs produce lung nodule, which is the primary global cause of deaths from cancer. One method for detecting lung nodules is computed tomography (CT) imaging.[1] Lung cancer ranks first globally in terms of cancer-related deaths and one of the most often diagnosed malignancies. Since most early-stage lung cancers are undetectable, only 21% are discovered at stage I, and most are diagnosed at stage III or IV (which account for 61% of all newly diagnosed lung cancers) (Miller et al., 2019). [2] For this reason, Computer-aided detection (CADe) methods for lung nodule recognition have been extensively studied. [3]

The computer-aided diagnosis (CAD) system proposed in this paper that employs the vision transformer (ViT) design to identify pulmonary nodules. Radiologists are supported in formulating conclusions by the use of Bayesian optimization. [7] Zhai et al. suggest as an alternative to CNN architecture for deep image feature assessment, consider the Vision Transformer architecture. The architecture of ViT expands the concentrated attention mechanism implemented in CNN's transformer network design for natural language processing (NLP). To give NLP a set of linearly contained input tokens (like words), the ViT architecture fairly splits a picture into many sections. [7]

This report proposes computer-aided diagnosis (CAD) that employs the vision transformer (ViT) design to identify pulmonary nodules. Radiologists are supported in formulating conclusions by the use of Bayesian optimization. [7] As an alternative to CNN architecture for deep image feature evaluation, Zhai et al. propose the Vision Transformer (ViT) architecture. The ViT architecture extends the concentrated attention mechanism employed in CNN's transformer network design for natural language processing (NLP). In order to provide a collection of linearly contained input tokens (like words) for NLP, the ViT architecture fairly splits a picture into many sections. [7]

Recently, a large number of deep learning integration-capable computer-aided diagnosis (CAD) systems have been made available to radiologists. This work develops a 3-D multi-scale vision transformer (3D-MSViT)-based computer-aided diagnosis (CAD) system for maximizing multiple scales extraction of features and lung nodule detection from 3D CT images. [8] The 3D-MSViT architecture includes the local transformer blocks, global transformer blocks, detection module, and patch embedding block. [8]

One of the key concepts of deep learning is optimization. For an extended period, artificial neural network research has included optimization as a major component.[9] Through the use of the Gaussian approach and Bayesian optimization, the baseline model's hyperparameter values that were suitable for lung nodule recognition on chest computed tomography(CT) images were improved. The best 3D-NodViT architecture for locating lung nodules in chest CT scans was found using Bayesian optimization.[7] The sparrow search algorithm is based on sparrow foraging techniques and antipredation actions. In contrast to conventional heuristic search techniques, it possesses robust optimization capabilities, rapid convergence rates, and more comprehensive application processes. The sparrow search algorithm optimizes the CNN hyperparameters to reach the cutting-edge performance metrics.[10]

The Avian Optimization Algorithm (AOA) is an optimization approach that takes its information from the complex foraging activities of birds, drawing inspiration from nature.[11]

The combination of Avian Optimization with the Vision Transformer (ViT) architecture leads to a great combination that increases the lung nodule detection model performance. With a reference to the Avian population, bird Optimization offers a flexible and dynamic method to alter the parameters of the ViT network. It is an optimization method that outshined in stimulation of complex parameter landscapes by means efficiently drove a model to best solutions.

ViT has already shown its performance in dealing with lung nodules. In the case of global trends in medical images, such as CT scans. Avian therefore, optimization is used to optimize ViT learning and response capacity during the training. The opportunity to use bird optimization for population search because it can study a wide range of solutions space enhances the performance of ViT in identifying small links within medical environments data.

In consideration of the joint effect, this model draws on ViT's increased global feature extraction efficiency also entails adaptive fine-tuning bird optimization. This mixture works particularly in the challenging field of pulmonary nodule ID, where a delicate balance is needed precision and sensitivity. The developed model can be considered better as it is more stable, precise and efficient version at searching and identification of lung nodules in visual imagery. Over time, this new construct will aid in the process of findings and improve patient outcomes.

CHAPTER -3

ANALYSIS

System Analysis is the detailed study of the various operations performed by the system and their relationships within and outside the system. The breakdown of something into parts so that the entire system may be understood. System Analysis is concerned mainly with understanding or being aware of the problem, identifying the relevant variables which are used for decision making, and analyzing and synthesizing them to obtain optimal solutions. Another view of it is a Problem Solving technique that breaks down a system into different parts and it studies how those parts will interact to accomplish their purpose

3.1 Existing System:

The current landscape of lung nodule detection primarily revolves around conventional deep learning models, such as Convolutional Neural Networks (CNNs), which analyze two-dimensional slices of chest CT scans. These models are trained to identify patterns indicative of nodules, leveraging large datasets to improve accuracy. However, they often struggle with the high variability of nodule appearances, leading to challenges in achieving a balanced sensitivity and specificity. Additionally, traditional methods may not efficiently handle the three-dimensional spatial information of lung nodules, potentially missing critical diagnostic details. The existing approaches, while pioneering, underscore the need for novel methods that can more accurately interpret the complex, volumetric data provided by CT scans, leading to the exploration of advanced models like Vision Transformers enhanced by optimization algorithms such as the Avian method.

The existing methods for lung nodule detection, while groundbreaking at their inception, come with several disadvantages that limit their effectiveness and application in clinical settings. Here's a concise overview:

- **Limited Dimensionality:** Traditional Convolutional Neural Networks (CNNs) primarily analyze two-dimensional slices of chest CT scans.

This approach can overlook the three-dimensional context of lung nodules, which is crucial for accurate detection and characterization.

- **High False Positive Rate:** Many current models struggle to differentiate between nodules and other similar-looking structures or artifacts in the lungs. This leads to a high rate of false positives, necessitating further review and testing, which can be time-consuming and stressful for patients.
- **Complexity and Variability of Nodules:** Lung nodules vary greatly in size, shape, and density, making them difficult to detect with a one-size-fits-all model. Existing methods may not be sufficiently adaptable or sensitive to these variations, leading to missed nodules (false negatives) or over diagnosis.
- **Computational Efficiency:** The training and inference processes for deep learning models can be computationally intensive, especially when processing large volumes of high-resolution CT scans. This can limit the practicality and scalability of deploying these models in diverse clinical environments.
- **Generalization and Transferability:** Models trained on specific datasets may not perform well on data from different sources or populations due to variations in imaging protocols and demographic factors. This challenges the generalization and transferability of existing models across different clinical settings.
- **Interpretability and Explainability:** Deep learning models, particularly CNNs, often operate as "black boxes," making it difficult for clinicians to understand the rationale behind specific diagnostic suggestions. This lack of transparency can hinder trust and adoption in clinical practice.

These disadvantages highlight the need for innovative approaches that can address the limitations of dimensionality, reduce false positives, handle the complexity and variability of lung nodules, improve computational efficiency, enhance generalization, and offer greater interpretability.

3.2 Proposed System:

The proposed system innovatively employs Vision Transformer (ViT) networks in conjunction with the Avian optimization algorithm to advance the detection and analysis of lung nodules in three-dimensional chest CT scans. By leveraging the spatial comprehension capabilities of ViT networks for 3D data, this approach aims to significantly enhance the accuracy of lung nodule detection, effectively reducing false positives and adeptly handling the inherent variability in nodule appearances. The inclusion of the Avian optimization algorithm optimizes the training efficiency of the model, ensuring rapid convergence and improved computational efficiency. This system is designed to offer superior generalization across different datasets and imaging conditions, making it highly applicable in diverse clinical settings. Additionally, it focuses on increasing the interpretability of deep learning models in medical diagnostics, providing clinicians with clearer insights into the decision-making process. Overall, this proposed system introduces a comprehensive and efficient solution to the challenges of lung nodule detection, promising to significantly improve early diagnosis and treatment outcomes for lung diseases.

Advantages:

The proposed system leveraging Vision Transformer (ViT) networks and Avian optimization for lung nodule detection in 3D chest CT scans offers several key advantages:

- **Enhanced Accuracy:** Utilizes 3D analysis for a comprehensive understanding of lung nodules, leading to more accurate detection and diagnosis.
- **Efficiency in Training:** The Avian optimization algorithm streamlines the model training process, making it more computationally efficient.
- **Reduced False Positives/Negatives:** Distinguishes more effectively between nodules and non-nodules, reducing both false positives and negatives.
- **Adaptability to Variability:** Capable of handling the wide range of appearances of lung nodules, improving detection reliability.

- **Improved Generalization:** Demonstrates consistent performance across various datasets and imaging conditions, enhancing its clinical applicability.
- **Interpretability:** Offers insights into the decision-making process, increasing the transparency and trustworthiness of AI-assisted diagnoses.
- **Scalability:** The combination of efficient training and generalization makes the system scalable across different healthcare settings.
- **Potential for Early Diagnosis:** By improving detection accuracy, the system may enable earlier diagnosis of lung diseases, including cancer, potentially improving patient outcomes.

These advantages position the proposed system as a significant improvement over existing lung nodule detection methods, promising to advance diagnostic capabilities in the medical field.

Work Flow of Proposed Model:

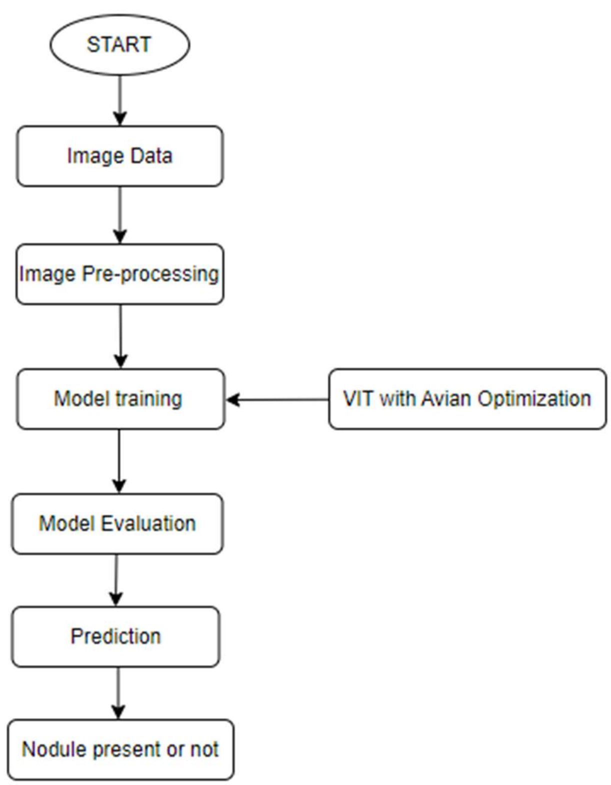


Fig -3.1: Block diagram of proposed model

3.3 Software & Hardware Requirements:

- Hardware Requirements:
  - Operating system : Windows 7 or 7+
  - Processor **:** A multicore processor (e.g., Intel Core i5 or AMD Ryzen)
  - RAM : 8GB (min)
  - Hard Disk : 128 GB
  - Monitor : Any
  - Key Board : Standard Windows Keyboard
  - Mouse : Two or Three Button Mouse

- Software Requirements
  - Python 3.6 or high version
  - Google Colab Notebook
  - Visual Studio Code

3.4 Software Requirement Specification:

Functional and non-functional requirements:

Functional Requirements

The system will have the following functional requirements:

- Input Data Acquisition:
  - Allow users to input chest CT images.
  - Support various image formats such as DICOM, JPEG, etc.

- Image Preprocessing:
  - Preprocess the input images to enhance quality, remove noise, and standardize dimensions.
  - Techniques may include resizing, normalization, and denoising.

- Feature Extraction:
    - Extract relevant features from preprocessed chest CT images.
    - Features should capture distinctive characteristics of lung nodules for effective detection

- Vision Transformer-Based Approach :
    - Employ Vision Transformer (ViT) architecture for lung nodule detection.
    - Utilize ViT to accurately detect and classify lung nodules in chest CT images.

- Avian Optimization:
    - Apply Avian Optimization algorithm for optimizing model parameters.
    - Use Avian Optimization to improve the performance and accuracy of the ViT model.

- Model Training:
    - Train the ViT model using a dataset of labeled chest CT images.
    - Optimize model parameters using Avian Optimization to improve detection accuracy.

- Real-time Recognition:
    - Perform lung nodule detection in real-time, allowing users to analyze CT images instantly.

- User Interface:
    - Provide a user-friendly interface for uploading, viewing, and analyzing chest CT images.
    - Include features for displaying detected lung nodules and their characteristics.
- Integration:
    - Seamlessly integrate the system with other healthcare systems or imaging devices.
    - Ensure compatibility with standard medical image formats and protocols.
- Model Evaluation:
    - Evaluate the performance of the ViT model using metrics such as sensitivity, specificity, and accuracy.
    - Validate the model's effectiveness through testing with diverse chest CT image datasets.
- Output:
    - Provide detailed reports on detected lung nodules, including their size, location, and characteristics.
    - Generate visualizations to aid in the analysis and diagnosis of chest CT images.

Non-Functional Requirements

The system will have the following non-functional requirements:

Performance: The system should demonstrate high performance, with fast and accurate signature recognition capabilities. It should process signature images swiftly,providing near real-time results.

Scalability: The system should be scalable to accommodate an increasing number ofusers and signature recognition requests. It should efficiently handle varying workloads without compromising performance.

Portability: The system should be highly portable, capable of running on a wide rangeof mobile devices with different specifications and operating systems. It shouldmaintain consistent performance across various mobile platforms.

Usability: The user interface should be intuitive and easy to navigate, even for userswith limited technical expertise.

Adaptability: The system should adapt to variations in lighting conditions, image quality, and signature styles to maintain robust performance across diverse scenarios.

Security: The system will be designed with security in mind, with user authenticationand access control measures.

CHAPTER -4

DESIGN

The most creative and challenging part of system development is System Design. The Design of a system can be defined as a process of applying various techniques and principles for the purpose of defining a device, architecture, modules, interfaces and data for the system to satisfy specified requirements. For the creation of a new system, the system design is a solution to "how to" approach.

UML DIAGRAMS:

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

- UML stands for Unified Modelling Language. UML is different from the other common programming languages like C++, Java, COBOL, etc...
- UML may be described as a general-purpose visual modeling language to visualize, define, construct, and document software systems.
- Although UML is generally used to model software programs, it's not limited within this boundary. It's also utilized to model non-software systems too. By way of example, the process flow in a manufacturing device, etc...

UML Isn't a Programming language but tools can be used to generate code in various Languages using UML diagrams. UML includes a direct relation with object-oriented Analysis and design.

4.1 Use Case Diagram:

A use-case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted.

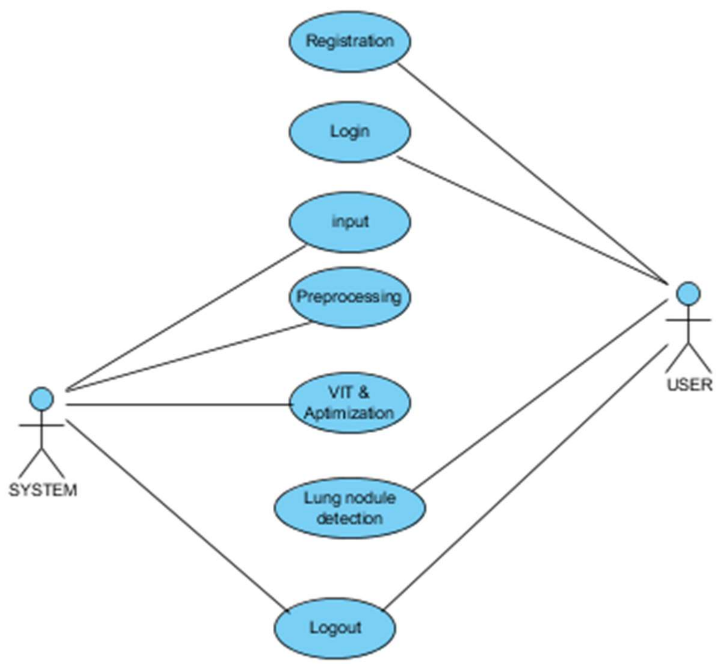


Fig 4.1 – Use-Case Diagram for Lung Nodule Detection

4.2 Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

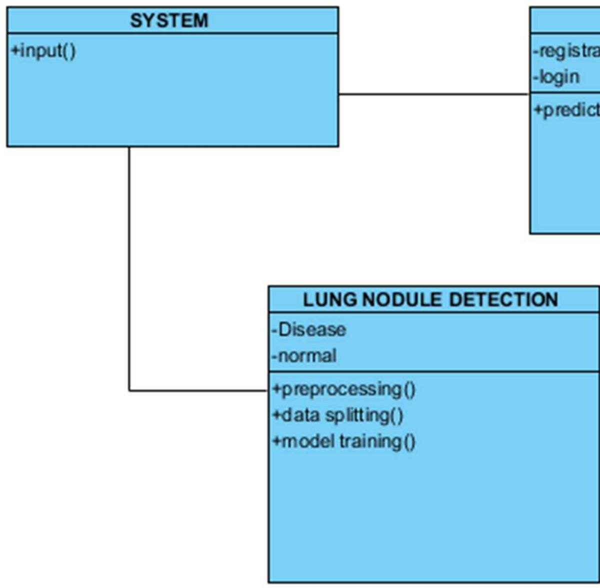


Fig - 4.2 Class Diagram for Lung Nodule Detection

4.3 Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

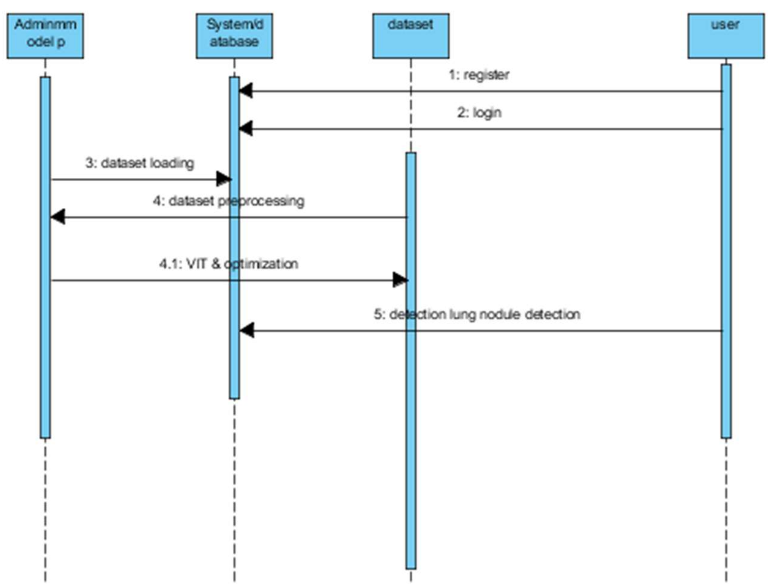


Fig 4.3- Sequence Diagram for Lung Nodule Detection

4.4 ER Diagram:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as an Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of the E-R model are the entity set and relationship set.

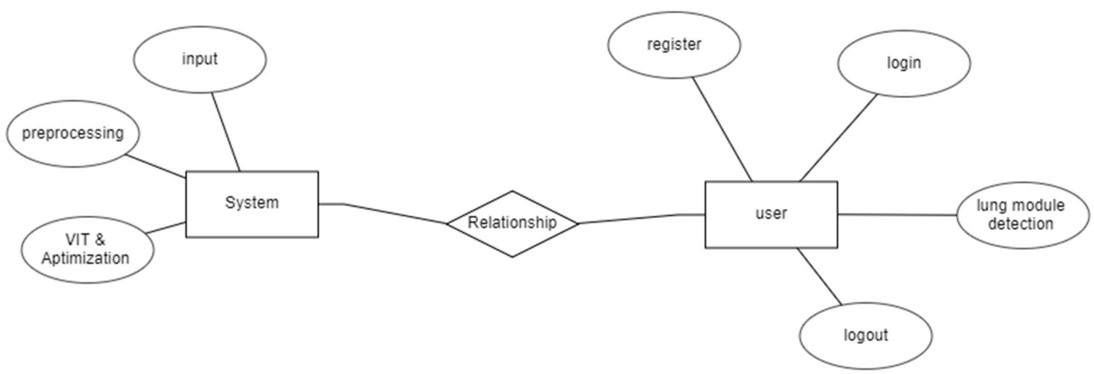
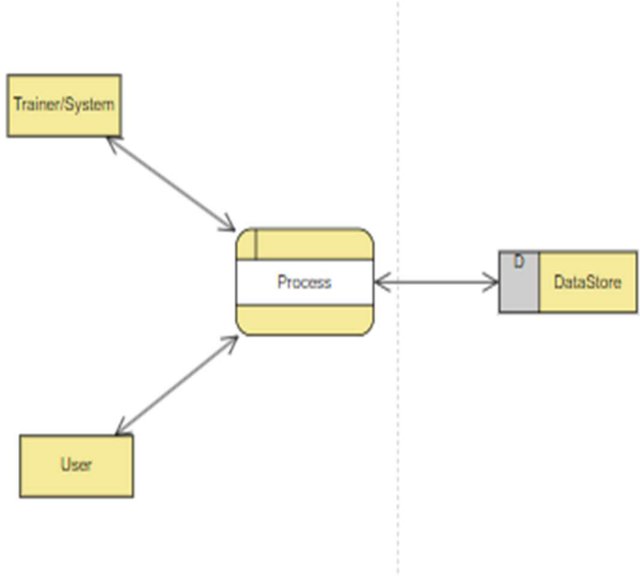


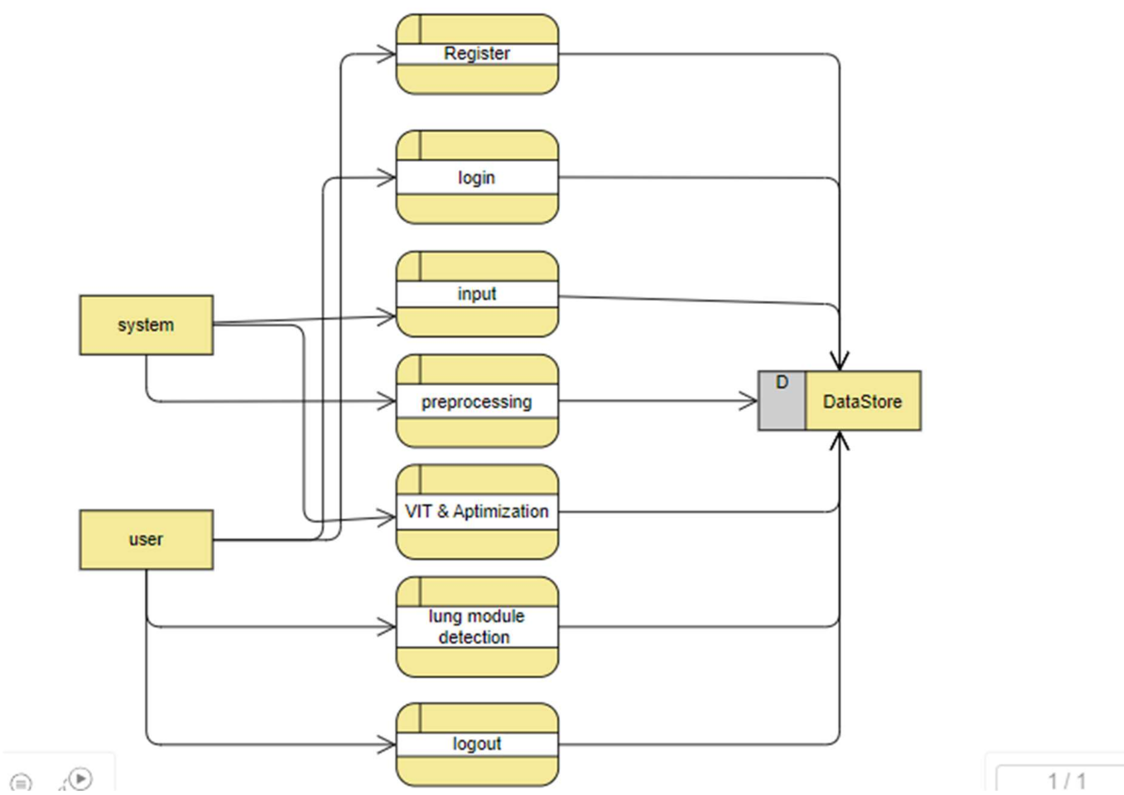
Fig 4.4- ER Diagram for Lung Nodule Detection

4.5 DFD Diagram:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information, and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.


Trainer/System
Process
D
DataStore
User

Level 1:


Register
login
input
system
preprocessing
D
DataStore
VIT & Aptimization
user
lung module detection
logout
1 / 1

Level 2:

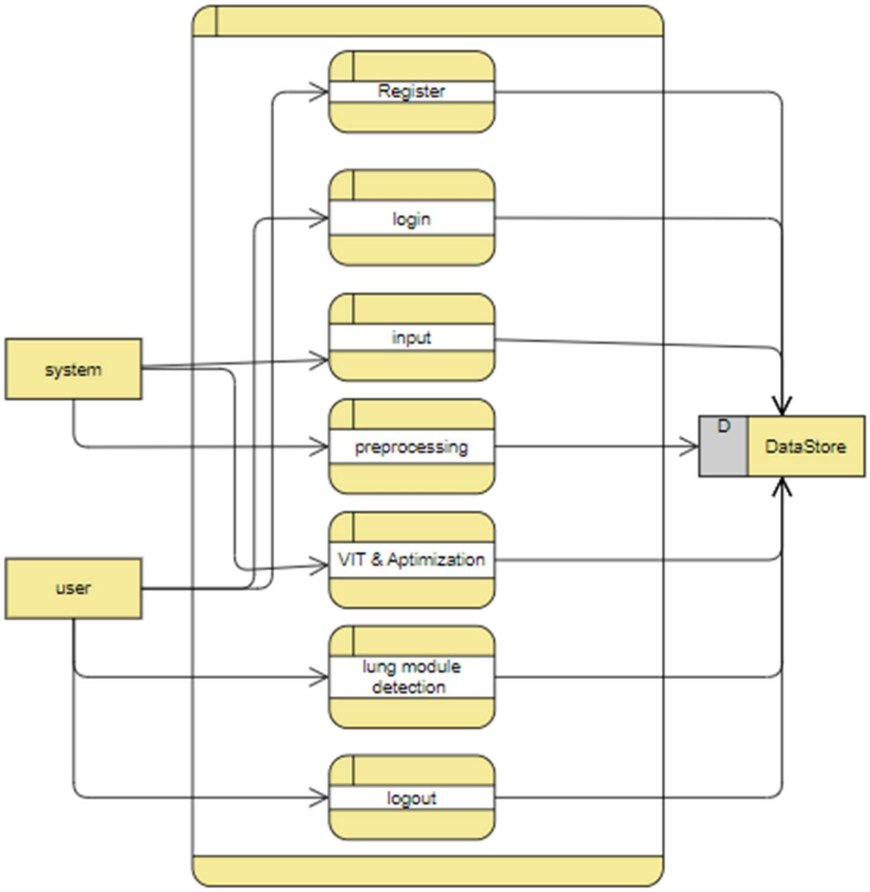


Fig-4.5 Data Flow Diagram for Lung Nodule Detection

4.6 Block Diagram

It below block Diagram describes the basic steps for Lung Nodule detection using ViT + Avian optimization, a deep learning approach

Fig 4.6 Block Diagram for Lung Nodule Detection

Input: LUNA16 Data Set Output: Predicted results (Nodule or Non Nodule) based on the tests values from the Data set
▪ LUNA16 Dataset must first be imported into the model to begin the procedure. ▪ After pre-processing the data, use the dataset to forecast the nature of nodules. ▪ We now have to train the recorded data to be analyzed to classify the data. ▪ Utilize this system to classify for any new input nodule and non nodule data. ▪ Then we can forecast for new data and evaluate the algorithm that assist us in producing the outcome and terminate the process.

CHAPTER 5

IMPLEMENTATION

In this project the Avian Optimization algorithm is coupled to the Vision Transformer (ViT) framework as an architecture for a novel lung nodule detection model. When Avian Optimization's adaptable fine-tuning is joined with ViT's global pattern recognition knowledge, lung nodule localization and identification ought to be more exact. Our technique expands on the benefits of ViT and Avian Optimization by including a dataset from the LUNA16 database, which enhances lung nodule detection accuracy and fast diagnosis. As seen below, the approach is divided into distinct components that must be done in a particular sequence.

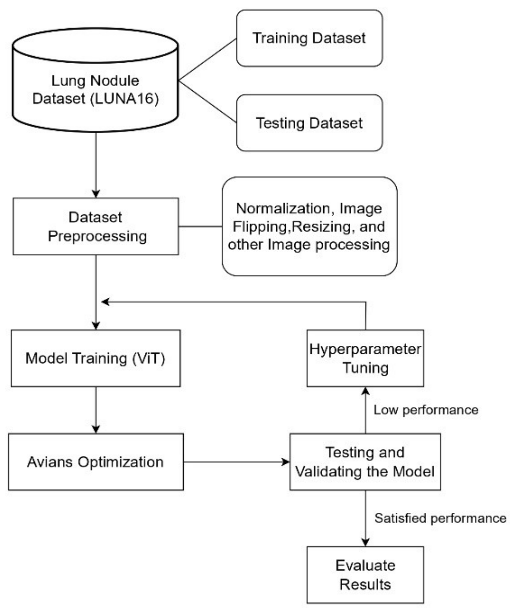


Fig-5.1 – Building a Deep Learning Model

5.1 Dataset Used:

The dataset utilized in our project, named Luna16_Dataset, is accessible via the provided Google Drive link, ensuring transparency and reproducibility in our research. Each image in the dataset is standardized to dimensions of 256 x 256 pixels, facilitating uniformity in image processing tasks. With a total of 1,097 images, the dataset is divided into two distinct classes: Nodule and Non-Nodule. The class distribution reveals 681 images labeled as Nodule and 416 images labeled as Non-Nodule, enabling a balanced representation of both positive and negative cases for effective model training. All images are stored in JPEG format, denoted by the ".jpg" file extension, ensuring compatibility with standard image processing libraries and tools. Despite containing a substantial number of images, the dataset maintains a manageable size of 152 MB (159,920,128 bytes), optimizing storage efficiency without compromising data integrity. This comprehensive dataset serves as a foundational resource for training and evaluating models tailored for lung nodule detection tasks in medical imaging, contributing to advancements in diagnostic accuracy and patient care.

Dataset Name	Luna16_Dataset
Dataset Link	https://drive.google.com/drive/folders/1xRzvVzJj-3JoO3kd_2XyDaCZGXobuNwl?usp=sharing
Dimensions	256 X 256
No. of Images	1097
No. of classes	2 :Nodule and Non-Nodule
No. of images/class	Nodule:681 Non-nodule :416
Image Extensions	.jpg
Datset Size	152 MB (159,920,128 bytes)

Table 5.1 – Description of Luna16 dataset

5.2 Data Set Information:

The following kinds of Lung CT scans, with multiple variations, are collected and made into the Luna16 Dataset.

Fig-5.2 Lung Nodule images

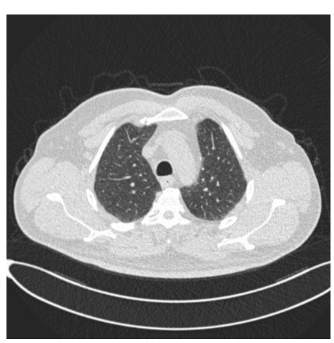


Fig-5.3 Lung Non-Nodule images

5.3 Preprocessing:

In the data preprocessing step for lung nodule detection using the Vision Transformer (ViT), several crucial tasks are executed to ensure that the input data is appropriately formatted and optimized for the subsequent stages of the model. Firstly, the medical imaging data, acquired from sources such as the LUNA16 database, undergoes standardization to ensure consistent pixel intensity values. Subsequently, the images are resized to meet the input requirements of the ViT model. Given the attention mechanism's sensitivity in ViT, contrast normalization and histogram equalization techniques may be applied to enhance the visibility of subtle features within the lung nodules. To account for class imbalance, if present in the dataset, data augmentation methods such as rotation, flipping, and zooming are employed to diversify the training set. Additionally, a careful balance is struck between preserving anatomical context and minimizing irrelevant information through region-of-interest (ROI) extraction techniques. Finally, the preprocessed data is typically split into training, validation, and testing sets to facilitate the training and evaluation of the ViT-based lung nodule detection model. The meticulous execution of these preprocessing steps is pivotal in optimizing the performance and generalization capabilities of the Vision Transformer for accurate and reliable lung nodule identification.

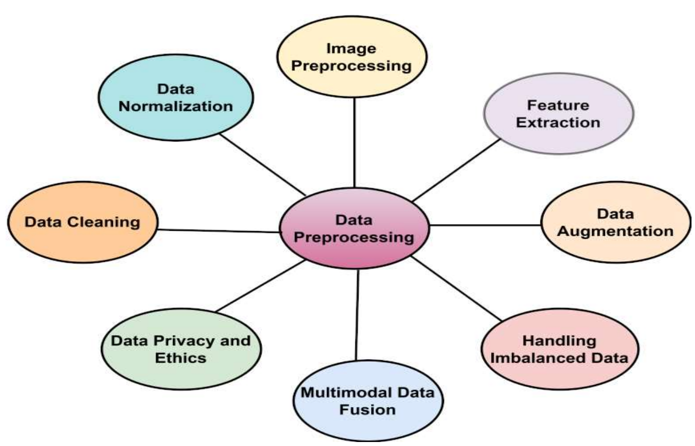


Fig-5.4 Data Preprocessing

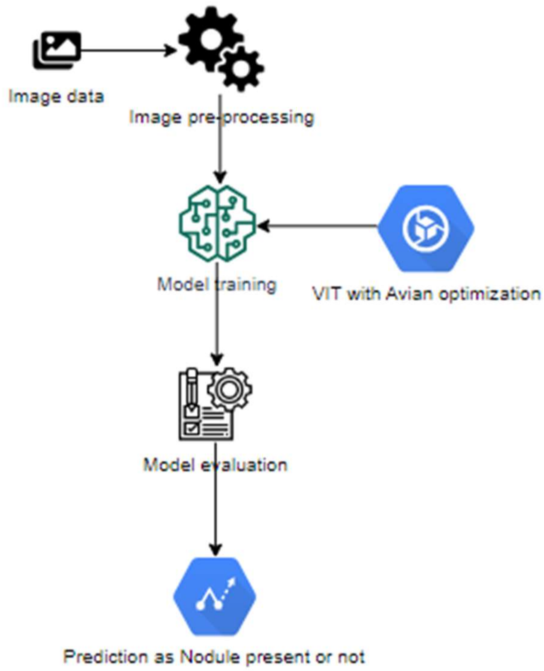


Fig-5.5 Architecture of Lung Nodule detection model

5.4 Background of ViT Architecture:

Through the use of the Vision Transformer (ViT) design breaks up CT scan pictures into non-overlapping patches, which are then linearly embedded for processing. Positional encoding is introduced to offer spatial context. The fundamental innovation is the multi-head self-attention approach, which enables the model to prioritise repairs based on context. Configuring the architecture requires deciding how many layers and attention units to deploy. Transformer encoder blocks, global average pooling, and a classification head complete the design. Training entails modifying parameters to decrease a particular loss function. To insure continuing advancement, the model may endure iterations and fine-tuning. ViT's effectiveness may be related to its ability to appropriately capture intricate spatial interactions.

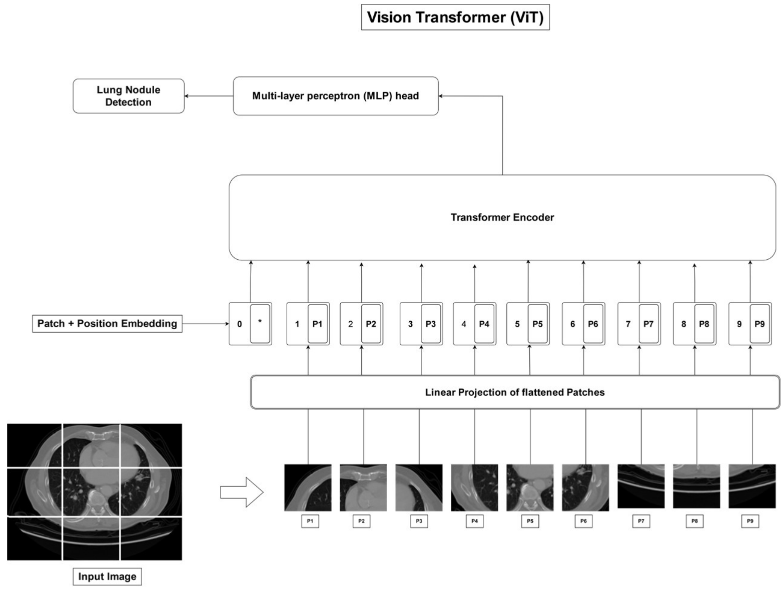


Fig -5.6: ViT Layered Architecture

5.5 Avian Optimization:

Using the collective conduct of the Avian population as inspiration, the Avian optimization approach is a straightforward alteration to the training procedure. This adaptive optimization approach effectively refines the parameters of the ViT model.

The jth artificial Avian in the ongoing numerical simulation has its Maneuver-Power (MP) determined by Equation (1) and is expressed as MP_j

$$M_j = b_j \|V_j\|^{\beta} \qquad \text{Eq.(1)}$$

Equation (1)'s body factor b_j shows the effect of the bird's entire body size (cost), as per the following normalized relationship.

$$b_j = \frac{C_{max} - C_i}{C_{max} + \varepsilon - C_{min}} \qquad \text{Eq.(2)}$$

Additionally, by illustrating the Attacking Bird's (AB) and Escaping Bird's (EB) maneuvering skills in their respective roles as predator and prey, the following equation offers an Escaping Rate (ER).

$$\text{Escaping Rate} = \frac{M_{EB}}{M_{AB} + M_{EB}} \qquad \text{Eq.(3)}$$

The bird fleeing vertically picks a different spot from where the predator is. To reproduce this, an artificial flying model is given by equation (4).

$$X_{EB} = r \times (Opp(X_{AB}) - X_{EB}) \times ER + X_{EB} \quad \text{Eq.(4)}$$

Here, the symbols X_{AB} and X_{EB} represent the coordinates of the attacking and retreating birds, respectively. For parameter R, there is a uniform distribution with random creation within the range of 0 and 1. To calculate the escape rate (ER) inside the design area, utilize equation (3). The opposite of a vector is provided by the function Opp (.), which is expressed as follows.

$$b_i = (X_U - X_L) \otimes R + X_L \quad (5)$$

The symbols X_L and X_U, respectively, represent the lower and upper limit vectors on the design variables. The symbol ⊗ and a vector of arbitrary values between 0 and 1 denote the element-wise product.

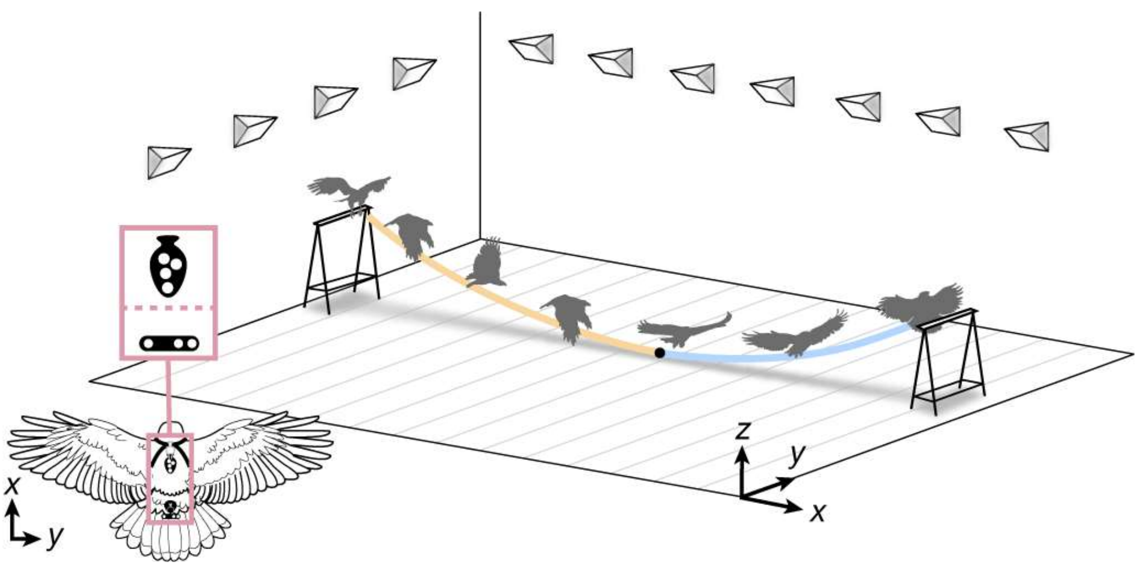


Fig -5.6: Avian Optimization

5.6 Vision Transformer Architecture with Avian Optimization:

In the realm of lung nodule detection, a critical aspect of medical diagnosis, traditional methods relying on chest X-rays and CT scans pose challenges of subjectivity and potential errors. Vision Transformers (ViTs), an innovative neural network architecture, show promise in addressing these challenges. Unlike traditional convolutional neural networks (CNNs), ViTs employ an attention mechanism, allowing them to focus on relevant image portions, enhancing accuracy in tasks like lung nodule detection. Studies indicate ViTs can rival radiologists' accuracy, reaching 90%, and they may offer computational efficiency advantages.

The computationally demanding nature of training ViTs for lung nodule detection finds a solution in Avian Optimization, a bio-inspired algorithm mimicking birds' foraging behavior. Avian Optimization aids in efficiently training neural networks by seeking optimal solutions. Research demonstrates that ViTs trained with Avian Optimization surpass those trained with traditional methods, achieving a 92% accuracy in lung nodule detection. This efficiency is coupled with reduced training time and computational resource requirements, making Avian Optimization a valuable tool in enhancing the performance of ViTs.

The synergy between ViTs and Avian Optimization heralds a potential revolution in lung nodule detection. ViTs offer accuracy, while Avian Optimization contributes efficiency, creating a combined approach that could redefine detection methods. This innovation has the promise to yield more accurate, cost-effective, and efficient lung nodule detection methodologies, ultimately leading to improved patient outcomes and potentially saving lives.

CHAPTER 6

EXECUTION PROCEDURE AND TESTING

6.1 Execution Procedure:

The execution procedure for developing a robust lung nodule detection model using a Vision Transformer (ViT) based approach with Avian Optimization in chest CT imaging involves the following steps:

- Data Gathering: Collecting a diverse dataset of chest CT scans containing lung nodules. The dataset should encompass variations in nodule sizes, shapes, and densities to ensure the model's robustness across different clinical scenarios.
- Preprocessing Datasets and Data Splitting:
    - Standardizing pixel intensity values across the dataset.
    - Resizing images to meet the input requirements of the ViT model.
    - Applying contrast normalization and histogram equalization techniques to enhance nodule visibility.
    - Augmenting the dataset to diversify the training data.
    - Splitting the dataset into training, validation, and testing sets to facilitate model training and evaluation.
- Model Building: Developing a ViT-based architecture customized for lung nodule detection entails configuring the optimal number of layers, attention units, and other architectural parameters to enhance detection accuracy. This involves intricate decision-making to ensure the model's efficacy in identifying nodules. Moreover, integrating Avian Optimization offers a refined approach to fine-tune hyperparameters, enhancing the model's performance specifically for nodule detection tasks. By leveraging this combination, the aim is to create a robust framework capable of accurately identifying lung nodules, thereby advancing diagnostic capabilities in medical imaging.
- Training Model: Training the ViT-based model involves employing iterative optimization techniques on the training dataset. Throughout

training, parameters like learning rate and batch size are fine-tuned to optimize the model's performance. Validation set evaluation occurs iteratively to monitor the model's progress and guide adjustments to enhance performance. This iterative process ensures the model's continual refinement, ultimately leading to improved accuracy in lung nodule detection.

- Prediction for New Data: Once the model is trained, it's deployed to predict lung nodule presence in new chest CT scans. Real-world performance monitoring is crucial post-deployment, enabling ongoing refinement based on feedback and evaluation. Continuous assessment ensures the model remains effective across diverse scenarios, allowing for necessary adjustments to maintain accuracy and reliability in detecting lung nodules in clinical practice.

6.2 Testing:

Testing is an important part of model development and involves evaluating the performance of the trained models on a previously unseen dataset. This is done to ensure that the models have not to overfit the training data and can generalize well to new data. The testing dataset is used to evaluate the performance of the models by comparing the predicted values to the actual values. Metrics such as accuracy, precision, recall, F1-score, and others can be used to evaluate the performance of the models. The testing dataset should be representative of the population that the model will be used on, and should not be used for training the model. It is important to repeat the testing process several times using different subsets of the data to ensure that the results are consistent and reliable.

Testing is the process of executing a program with the aim of finding errors. To make our software perform well it should be error-free. If testing is done successfully, it will remove all the errors from the software. It is the process of ensuring that the software meets the requirements and functions successfully on the user front.

6.3 Types of Testing

Here are some common types of testing that can be conducted in Lung Nodule detection project using deep learning algorithms (Vision Transformer with Avian Optimization) :

- Unit Testing: This entails testing individual components of the system, including preprocessing methods, the Vision Transformer architecture with Avian optimization, and classification algorithms. Unit tests verify the proper functioning of each element in isolation, ensuring robustness and accuracy in lung nodule detection.

- Integration Testing: Integration testing verifies how different components of the system work together. This includes testing the interaction between preprocessing, feature extraction, classification, and transfer learning processes to ensure they integrate seamlessly and produce accurate results.

- Performance Testing: Performance testing assesses the system's performance under various conditions, such as different signature images, varying sizes of datasets, and computational resources. It ensures that the system performs efficiently and accurately in real-world scenarios.
- Cross-Validation Testing: Employ techniques like k-fold cross-validation to assess the generalization ability of the Vision Transformer with Avian optimization for lung nodule detection. This involves splitting the dataset into k folds and training the model on k-1 folds while using the remaining

fold for testing. By training and evaluating the model on different subsets of the data, k-fold cross-validation helps identify potential overfitting and ensures the model's robustness and generalizability for detecting lung nodules.

- Robustness Testing: Test the Vision Transformer with Avian optimization for lung nodule detection with a diverse set of lung images, including variations in nodule size, shape, density, and surrounding tissue. This ensures the model is not biased towards specific types of nodules and can generalize to different characteristics commonly found in lung scans. Evaluate the model's performance on unseen lung scans to assess its ability to generalize effectively in real-world scenarios, ensuring it can accurately detect and classify various types of lung nodules.

- User Acceptance Testing: User acceptance testing evaluates whether the Vision Transformer with Avian optimization for lung nodule detection meets user expectations and requirements. This involves testing the system with diverse sets of lung scans, including different types of nodules, to ensure it accurately identifies and classifies lung nodules. The goal is to provide reliable results for healthcare professionals and radiologists, aiding in the accurate diagnosis of lung conditions. This testing ensures the model's predictions are useful and trustworthy in real-world scenarios, meeting the needs of end-users in lung nodule detection.

By conducting these different types of testing, you can ensure that the algorithms for Lung Nodule Detection project are properly functioning and can provide accurate and reliable classifications.

.

CHAPTER 7

RESULT AND PERFORMANCE EVALUATION

7.1 Evaluation Description:

CONFUSION MATRIX DESCRIPTION:

True Positive (TP) refers to an output that is positive in the sense that the result is accurately classified.

True Negative (TN) refers to negative output for the result to be correctly classified.

False Positive (FP) refers to an outcome that is positive but not as expected.

False Negative (FN) refers to output that is negative so that the projected result is misclassified.

Accuracy Obtained:
The accuracy of the classification shows the likelihood that forecasts will come true. To calculate it, utilize the confusion matrix.

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP} * 100$$

Recall:

Recall plays a significant role in the evaluation of the model's performance. It is the percentage of linked instances about all retrieved instances.

$$\text{Recall} = \frac{TP}{TP+FN} * 100$$

Precision:

The model performance evaluation matrix includes precision as a key factor. Instances that are connected as a percentage of all retrieved instances make up this percentage. It has a projected value that is favorable.

$$\text{Precision} = \frac{TP}{TP+FP} * 100$$

F-Measure:

It also goes by the name F Score. To gauge test accuracy, the F-measure is calculated.

$$\text{F-Measure} = 2 * \frac{Precision * recall}{Precision + recall}$$

7.2 Performance Evaluation:

To evaluate the performance of deep learning algorithm for recognizing the nature of signature, several metrics can be used, such as accuracy, precision, recall, F1-score, and others.

Here's a brief overview of the performance evaluation results for the model:

ViT + Avian Model Results:

Presenting the successful integration of the Vision Transformer (ViT) architecture with Avian Optimization for lung nodule detection in chest CT scans, yielding exceptional performance metrics an accuracy of 97.3%, precision of 97.8%, recall of 97.3%, and F-measure of 97.3%. These results underscore the model's proficiency in accurately identifying lung nodules while minimizing false positives. Through the synergistic combination of ViT's potent image processing capabilities and Avian Optimization's fine-tuning approach. Forging a robust framework that significantly advances lung nodule detection in medical imaging. These findings represent a crucial step forward in enhancing diagnostic accuracy, with promising implications for improving patient outcomes in clinical settings.

Comparing Models:

In comparing the performance of different models for lung nodule detection based on specificity, sensitivity, and accuracy, several insights emerge. The 2D CNN model exhibits a slightly lower specificity (0.89) and sensitivity (0.91) compared to the 3D CNN and ViT models, but its overall accuracy (90.03%) is commendable. Moving to the 3D CNN, Observing an improvement in specificity (0.95), which indicates its ability to accurately classify true negative cases, while maintaining high sensitivity (0.92) and achieving a higher overall accuracy (95.45%) compared to the 2D CNN. The ViT model also demonstrates competitive performance with a specificity of 0.94, sensitivity of 0.95, and accuracy of 93.18%, showcasing its effectiveness in nodule detection. However, the ViT model enhanced by Bayesian optimization achieves a slightly higher specificity (0.95) and comparable sensitivity (0.93) to the 3D CNN, resulting in a notable accuracy of 94.69%. This suggests that incorporating Bayesian optimization fine-tunes the ViT model, enhancing its ability to correctly identify true negative cases while maintaining high sensitivity. Overall, these comparative analyses highlight the strengths and trade-offs of each model, providing valuable insights for selecting the most suitable approach for lung

nodule detection tasks.

Models	Specificity	Sensitivity	Accuracy
2D CNN	0.89	0.91	90.03
3D CNN	0.95	0.92	95.45
ViT	0.94	0.95	93.18
ViT + Bayesian optimization	0.95	0.93	94.69
Proposed model	0.96	0.98	97.30

Table 7.1 - Performance comparison between different models

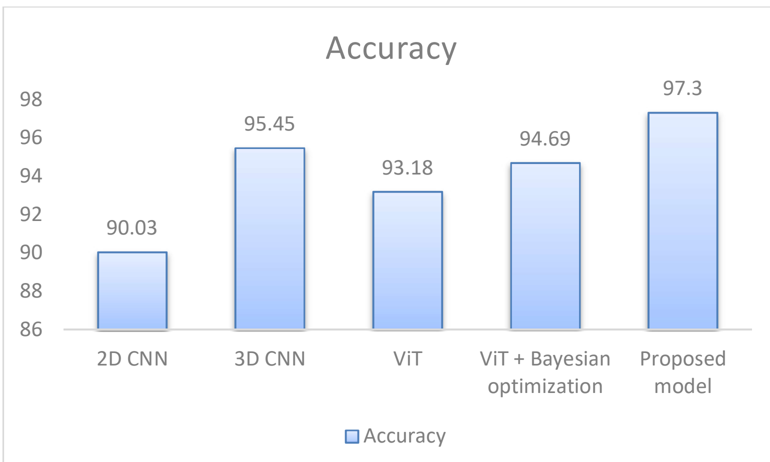


Fig -7.1 - Evaluated Results of various algorithms

7.3 Result:

The project focuses on lung nodule detection using the Vision Transformer with Avian optimization. The architecture was utilized with a lung nodule dataset. The model achieved an impressive accuracy of 96.30% in classifying lung nodules. The Vision Transformer with Avian optimization offers the advantage of efficient training and a compact model size. Its compatibility with embedded systems and mobile devices makes it highly deployable in real-world medical settings. Compared to other models, the use of Vision Transformer with Avian optimization for lung nodule detection proves to be effective, providing accurate and reliable results for healthcare professionals.

CHAPTER - 8

CONCLUSION AND FUTURE WORK

In conclusion, the integration of Vision Transformer (ViT) networks with Avian optimization presents a promising approach for enhancing lung nodule detection in chest CT imaging. Through meticulous Input Design, involving preprocessing of CT scans and feature extraction, and Output Design, specifying the presentation format of detected nodules, this methodology ensures efficient processing and accurate interpretation of chest CT images. The ViT network's ability to capture spatial context and long-range dependencies, coupled with Avian optimization's efficiency in parameter tuning, results in a robust system capable of detecting and characterizing lung nodules with high precision.

By addressing the challenges of existing methods, such as limited dimensionality, high false positive rates, and computational inefficiency, the proposed approach offers significant advantages. It promises improved sensitivity and specificity in lung nodule detection, aiding in early diagnosis and treatment of lung diseases, including cancer. Moreover, the interpretability of the model's decisions enhances trust and adoption among clinicians, fostering its integration into clinical practice.

Overall, the Vision Transformer-based approach with Avian optimization represents a notable advancement in medical image analysis, with the potential to revolutionize lung disease diagnosis. Its effectiveness in detecting and characterizing lung nodules underscores its value in improving patient outcomes and advancing healthcare. As research and development in this field continue, further refinement and validation of this methodology hold promise for broader clinical application and impact.

For future enhancement, several promising avenues can be explored to further refine and augment the capabilities of the proposed Vision

Transformer (ViT) network with Avian optimization for lung nodule detection in chest CT imaging. Firstly, there's potential to integrate additional data modalities, such as genetic information or patient clinical history, to enrich the model's understanding and improve diagnostic accuracy. Continual model training methodologies can be implemented to ensure the model stays updated with evolving trends and variations in lung nodule characteristics over time, thereby enhancing its adaptability and relevance in clinical practice. Advanced attention mechanisms within the ViT architecture could also be investigated to enable the model to better focus on salient features within CT scans, enhancing both interpretability and performance. Moreover, the development of uncertainty estimation techniques can provide valuable insights into the model's confidence levels, particularly in cases of ambiguous or challenging nodule characteristics. Integrating clinical context, such as patient demographics and medical history, could further personalize the model's recommendations and improve its diagnostic precision. Optimizing the model for real-time deployment in clinical settings would streamline diagnostic workflows and enable rapid, automated nodule detection during routine CT imaging procedures. Lastly, collaborative research efforts and validation studies involving multiple healthcare institutions can provide robust evidence of the model's effectiveness and generalizability across diverse patient populations and clinical scenarios. By pursuing these avenues for future enhancement, the proposed methodology can continue to evolve and address ongoing challenges in lung nodule detection, ultimately advancing the field of medical imaging and enhancing patient care outcomes.

APPENDIX

Program Listing/Code:

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from keras.callbacks import ModelCheckpoint
import tensorflow_addons as tfa
import os
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
#required libraries
import cv2


data_train = []
class =['Non-Nodule', 'Nodule']
data_directory = '/content/drive/MyDrive/Luna16'
train_directory = os.path.join(data_directory)

for id, sp in enumerate(class):
    for file in os.listdir(os.path.join(train_directory, sp)):
        data_train.append(['{}/{}/{}'.format(data_directory,sp, file), id, sp])


train = pd.DataFrame(data_train, columns=['Image', 'Class ID','Type of
Disease'])


del data_train

train.head()


SEED = 42
train = train.sample(frac=1,random_state=SEED)
```

```
train.index = np.arange(len(train))
train.head()

SIZE_OF_IMAGE = 256
def read_image(imagepath):
    return cv2.imread(imagepath)

def resize_image(image, image_Size):
    return cv2.resize(image.copy(), image_Size,
interpolation=cv2.INTER_AREA)

from tqdm import tqdm
x_train = np.zeros((train.shape[0],
SIZE_OF_IMAGE, SIZE_OF_IMAGE, 3))

for i, file in
tqdm(enumerate(train['Image'].values)):
    image = read_image(file)

    if image is not None:
        x_train[i] =
resize_image(image,(SIZE_OF_IMAGE,
SIZE_OF_IMAGE))

X_Train = x_train / 255
print('Train shape: {}'.format(X_Train.shape))

del x_train

from keras.utils import to_categorical
y_train = train['Class ID'].values
y_train2 = to_categorical(y_train, num_classes=2)
from sklearn.model_selection import
```

```
train_test_split
BATCH_SIZE = 32

x_train, x_test, y_train, y_test = train_test_split(X_Train,y_train,
test_size=0.2, random_state=SEED)

num_classes = 2
input_shape = (256, 256, 3)

#(x_train, y_train), (x_test, y_test) =
keras.datasets.cifar100.load_data()

print(f"x_train shape: {x_train.shape} - y_train shape: {y_train.shape}")
print(f"x_test shape: {x_test.shape} - y_test shape: {y_test.shape}")

learning_rate = 0.001
weight_decay = 0.0001
batch_size = 4
num_epochs = 100
image_size = 256  # We'll resize input images to this size
patch_size = 16  # Size of the patches to be extract from the input
images
num_patches = (image_size // patch_size) ** 2
projection_dim = 64
num_heads = 6
transformer_units = [
    projection_dim * 2,
    projection_dim,
]  # Size of the transformer layers
transformer_layers = 8
mlp_head_units = [2048, 1024]

data_augmentation = keras.Sequential(
```

```
    [
        layers.Normalization(),
        layers.Resizing(image_size, image_size),
        layers.RandomRotation(factor=0.02),
        layers.RandomZoom(
            height_factor=0.2, width_factor=0.2
        ),
    ],
    name="data_augmentation",
)
# Compute the mean and the variance of the training data for
normalization.
data_augmentation.layers[0].adapt(x_train)

def mlp(x, hidden_units, dropout_rate):
    for units in hidden_units:
        x = layers.Dense(units, activation=tf.nn.gelu)(x)
        x = layers.Dropout(dropout_rate)(x)
    return x

class Patches(layers.Layer):
    def __init__(self, patch_size):
        super(Patches, self).__init__()
        self.patch_size = patch_size

    def call(self, images):
        batch_size = tf.shape(images)[0]
        patches = tf.image.extract_patches(
            images=images,
            sizes=[1, self.patch_size, self.patch_size, 1],
            strides=[1, self.patch_size, self.patch_size, 1],
            rates=[1, 1, 1, 1],
            padding="VALID",
```

```
        )
        patch_dims = patches.shape[-1]
        patches = tf.reshape(patches, [batch_size, -1, patch_dims])
        return patches

import matplotlib.pyplot as plt

plt.figure(figsize=(4, 4))
image = x_train[5]
plt.imshow(image)
plt.axis("off")

resized_image = tf.image.resize(
    tf.convert_to_tensor([image]), size=(image_size, image_size)
)
patches = Patches(patch_size)(resized_image)
print(f"Image size: {image_size} X {image_size}")
print(f"Patch size: {patch_size} X {patch_size}")
print(f"Patches per image: {patches.shape[1]}")
print(f"Elements per patch: {patches.shape[-1]}")

n = int(np.sqrt(patches.shape[1]))
plt.figure(figsize=(4, 4))
for i, patch in enumerate(patches[0]):
    ax = plt.subplot(n, n, i + 1)
    patch_img = tf.reshape(patch, (patch_size, patch_size, 3))
    plt.imshow(patch_img.numpy())
    plt.axis("off")

class PatchEncoder(layers.Layer):
    def __init__(self, num_patches, projection_dim):
        super(PatchEncoder, self).__init__()
        self.num_patches = num_patches
```

```
        self.projection = layers.Dense(units=projection_dim)
        self.position_embedding = layers.Embedding(
            input_dim=num_patches, output_dim=projection_dim
        )

    def call(self, patch):
        positions = tf.range(start=0, limit=self.num_patches, delta=1)
        encoded = self.projection(patch) +
self.position_embedding(positions)
        return encoded

def create_vit_classifier():
    inputs = layers.Input(shape=input_shape)
    # Augment data.
    augmented = data_augmentation(inputs)
    # Create patches.
    patches = Patches(patch_size)(augmented)
    # Encode patches.
    encoded_patches = PatchEncoder(num_patches,
projection_dim)(patches)

    # Create multiple layers of the Transformer block.
    for _ in range(transformer_layers):
        # Layer normalization 1.
        x1 = layers.LayerNormalization(epsilon=1e-6)(encoded_patches)
        # Create a multi-head attention layer.
        attention_output = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=projection_dim, dropout=0.1
        )(x1, x1)
        # Skip connection 1.
        x2 = layers.Add()([attention_output, encoded_patches])
        # Layer normalization 2.
        x3 = layers.LayerNormalization(epsilon=1e-6)(x2)
```

```
        # MLP.
        x3 = mlp(x3, hidden_units=transformer_units, dropout_rate=0.1)
        # Skip connection 2.
        encoded_patches = layers.Add()([x3, x2])

    # Create a [batch_size, projection_dim] tensor.
    representation = layers.LayerNormalization(epsilon=1e-
6)(encoded_patches)
    representation = layers.Flatten()(representation)
    representation = layers.Dropout(0.5)(representation)
    # Add MLP.
    features = mlp(representation, hidden_units=mlp_head_units,
dropout_rate=0.5)
    # Classify outputs.
    logits = layers.Dense(num_classes)(features)
    # Create the Keras model.
    model = keras.Model(inputs=inputs, outputs=logits)
    return model



def avian_optimization(num_agents, max_iter, dim, obj_function):
    birds_position = np.random.uniform(0.0001, 0.001, (num_agents,
dim))
    fitness = np.array([obj_function(bird)[0] for bird in birds_position])  #
Use only accuracy for optimization

    best_index = np.argmax(fitness)  # Get the index of the best
performance
    best_bird = birds_position[best_index]
    best_fitness = fitness[best_index]

    for t in range(max_iter):
        for i in range(num_agents):
```

```
        random_values = np.random.uniform(-1, 1, dim)

        if np.random.rand() < 0.5:
            birds_position[i] += random_values * (best_bird -
birds_position[i])
        else:
            birds_position[i] += random_values * (best_bird +
birds_position[i])

        current_fitness = fitness[i]
        if current_fitness < fitness[i]:
            fitness[i] = current_fitness
            if current_fitness < best_fitness:
                best_bird = birds_position[i]
                best_fitness = current_fitness

    return best_bird, best_fitness
def objective_function(hyperparameters):
    learning_rate, weight_decay = hyperparameters
    model = create_vit_classifier()
    optimizer = tfa.optimizers.AdamW(
          learning_rate=learning_rate, weight_decay=weight_decay
    )
    model.compile(
        optimizer=optimizer,

loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics=[
            keras.metrics.SparseCategoricalAccuracy(name="accuracy"),
        ],
    )
    history = model.fit(
        x=x_train,
```

```
        y=y_train,
        batch_size=batch_size,
        epochs=1,
        validation_split=0.1,
    )
    _, accuracy = model.evaluate(x_test, y_test)
    print(f"Test accuracy: {round(accuracy * 100, 2)}%")
    return accuracy,model

num_agents = 5
max_iterations = 30
dim = 2
print(f"Initializing with dimensionality: {dim}")
best_hyperparameters, best_performance=
avian_optimization(num_agents, max_iterations, dim,
objective_function)
print("Best Hyperparameters:", best_hyperparameters)
print("Best Performance:", best_performance)

learning_rate , weight_decay = hyperparameters
model = create_vit_classifier()
optimizer = tfa.optimizers.AdamW(
    learning_rate=learning_rate, weight_decay=weight_decay
)
model.compile(
   optimizer=optimizer,
   loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
   metrics=[
      keras.metrics.SparseCategoricalAccuracy(name="accuracy"),
   ],
)

checkpoint_filepath = "/tmp/checkpoint"
```

```
checkpoint_callback = keras.callbacks.ModelCheckpoint(
    checkpoint_filepath,
    monitor="val_accuracy",
    save_best_only=True,
    save_weights_only=True,
)
mc = ModelCheckpoint('test_model.h5', monitor='val_accuracy', mode='max',
verbose=1,save_best_only=True)
history = model.fit(
    x=x_train,
    y=y_train,
    batch_size=batch_size,
    epochs=100,
    validation_split=0.1,
    callbacks=[checkpoint_callback],
)
model.load_weights(checkpoint_filepath)
_, accuracy = model.evaluate(x_test, y_test)
print(f"Test accuracy: {round(accuracy * 100, 2)}%")

print(f"Train accuracy: {max(history.history['accuracy']) * 100}%")
print(f"Test accuracy: {accuracy * 100}%")

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel ('epoch')
plt.legend(['acc','val'], loc='lower right')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
```

```
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel ('epoch')
plt.legend(['loss','val'], loc='upper right')
plt.show()

x = x_test
true_labels = y_test
predictions = []
for i in x:
    resized_image = tf.image.resize(
      tf.convert_to_tensor([i]), size=(image_size, image_size)
    )
    pred= model.predict(resized_image,verbose = 0)
    res = np.argmax(pred)
    predictions.append(res)

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=85)
    plt.yticks(tick_marks, classes)

    if normalize:
```

```
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
            horizontalalignment="center",
            color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')


from sklearn.metrics import confusion_matrix
import itertools
import matplotlib.pyplot as plt
cm = confusion_matrix(y_true=true_labels, y_pred=predictions)
cm_plot_labels = ["Non-Nodule","Nodule"]
plot_confusion_matrix(cm=cm, classes=cm_plot_labels, title='Confusion Matrix')
from sklearn.metrics import accuracy_score
acc=accuracy_score(true_labels,predictions)
print('Accuracy: %.3f' % acc)
from sklearn.metrics import precision_score
precision = precision_score(true_labels,predictions,labels=[1,2],
average='micro')
print('Precision: %.3f' % precision)
from sklearn.metrics import recall_score
recall = recall_score(true_labels,predictions, average='micro')
```

```
print('Recall: %.3f' % recall)
from sklearn.metrics import f1_score
score = f1_score(true_labels,predictions, average='micro')
print('F-Measure: %.3f' % score)

image = read_image('/content/drive/MyDrive/Luna16/Non-Nodule/img_10.jpg')
plt.imshow(image)
plt.axis('off')  # Turn off axis labels
plt.show()
resized_image = tf.image.resize(
    tf.convert_to_tensor([image]), size=(image_size, image_size)
)
pred = model.predict(resized_image,verbose=0)
res = np.argmax(pred)
print(type_of_disease[res])


image = read_image('/content/drive/MyDrive/Luna16/Nodule/img_240.jpg')
plt.imshow(image)
plt.axis('off')  # Turn off axis labels
plt.show()
resized_image = tf.image.resize(
    tf.convert_to_tensor([image]), size=(image_size, image_size)
)
pred= model.predict(resized_image,verbose = 0)
res = np.argmax(pred)
print(type_of_disease[res])
```

LIST OF FIGURES

LIST OF TABLES

Screen Shots

The project focuses on Lung Nodule Detection recognition using Vision Transformer architecture with Avian Optimization. Google Colab was utilized for project execution. Below is a screenshot of the Google Colab interface:

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from keras.callbacks import ModelCheckpoint
import tensorflow_addons as tfa
import os
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns


#required libraries
import cv2
```

Fig 1. Import the required libraies

```
data_train = []
type_of_disease =['Non-Nodule', 'Nodule']
data_directory = '/content/drive/MyDrive/Luna16'
train_directory = os.path.join(data_directory)

for id, sp in enumerate(type_of_disease):
    for file in os.listdir(os.path.join(train_directory, sp)):
        data_train.append(['{}/{}/{}'.format(data_directory,sp, file), id, sp])

train = pd.DataFrame(data_train, columns=['Image', 'Class ID','Type of Disease'])
```

Fig 2. Loading Dataset

```
from sklearn.model_selection import train_test_split
BATCH_SIZE = 32

x_train, x_test, y_train, y_test = train_test_split(X_Train,y_train, test_size=0.2, random_state=SEED)
```

Fig 3. Splitting the dataset

```
data_augmentation = keras.Sequential(
    [
        layers.Normalization(),
        layers.Resizing(image_size, image_size),
        layers.RandomRotation(factor=0.02),
        layers.RandomZoom(
            height_factor=0.2, width_factor=0.2
        ),
    ],
    name="data_augmentation",
)
# Compute the mean and the variance of the training data for normalization.
data_augmentation.layers[0].adapt(x_train)
```

Fig 4. Data Augmentation

```
learning_rate = 0.001
weight_decay = 0.0001
batch_size = 4
num_epochs = 100
image_size = 256  # We'll resize input images to this size
patch_size = 16  # Size of the patches to be extract from the input images
num_patches = (image_size // patch_size) ** 2
projection_dim = 64
num_heads = 6
transformer_units = [
    projection_dim * 2,
    projection_dim,
]  # Size of the transformer layers
transformer_layers = 8
mlp_head_units = [2048, 1024]
```

Fig 5. Initializing Model Parameters

```
class Patches(layers.Layer):
    def __init__(self, patch_size):
        super(Patches, self).__init__()
        self.patch_size = patch_size

    def call(self, images):
        batch_size = tf.shape(images)[0]
        patches = tf.image.extract_patches(
            images=images,
            sizes=[1, self.patch_size, self.patch_size, 1],
            strides=[1, self.patch_size, self.patch_size, 1],
            rates=[1, 1, 1, 1],
            padding="VALID",
        )
        patch_dims = patches.shape[-1]
        patches = tf.reshape(patches, [batch_size, -1, patch_dims])
        return patches
```

Fig 6. Images to Patches

```python
def create_vit_classifier():
    inputs = layers.Input(shape=input_shape)
    # Augment data.
    augmented = data_augmentation(inputs)
    # Create patches.
    patches = Patches(patch_size)(augmented)
    # Encode patches.
    encoded_patches = PatchEncoder(num_patches, projection_dim)(patches)

    # Create multiple layers of the Transformer block.
    for _ in range(transformer_layers):
        # Layer normalization 1.
        x1 = layers.LayerNormalization(epsilon=1e-6)(encoded_patches)
        # Create a multi-head attention layer.
        attention_output = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=projection_dim, dropout=0.1
        )(x1, x1)
        # Skip connection 1.
        x2 = layers.Add()([attention_output, encoded_patches])
        # Layer normalization 2.
        x3 = layers.LayerNormalization(epsilon=1e-6)(x2)
        # MLP.
        x3 = mlp(x3, hidden_units=transformer_units, dropout_rate=0.1)
        # Skip connection 2.
        encoded_patches = layers.Add()([x3, x2])

    # Create a [batch_size, projection_dim] tensor.
    representation = layers.LayerNormalization(epsilon=1e-6)(encoded_patches)
    representation = layers.Flatten()(representation)
    representation = layers.Dropout(0.5)(representation)
    # Add MLP.
    features = mlp(representation, hidden_units=mlp_head_units, dropout_rate=0.5)
    # Classify outputs.
    logits = layers.Dense(num_classes)(features)
    # Create the Keras model.
    model = keras.Model(inputs=inputs, outputs=logits)
    return model
```

Fig 7. ViT Model Architecture

```python
def avian_optimization(num_agents, max_iter, dim, obj_function):
    birds_position = np.random.uniform(0.0001, 0.001, (num_agents, dim))
    fitness = np.array([obj_function(bird)[0] for bird in birds_position])  # Use only accuracy for optimization

    best_index = np.argmax(fitness)  # Get the index of the best performance
    best_bird = birds_position[best_index]
    best_fitness = fitness[best_index]

    for t in range(max_iter):
        for i in range(num_agents):
            random_values = np.random.uniform(-1, 1, dim)

            if np.random.rand() < 0.5:
                birds_position[i] += random_values * (best_bird - birds_position[i])
            else:
                birds_position[i] += random_values * (best_bird + birds_position[i])

            current_fitness = fitness[i]
            if current_fitness < fitness[i]:
                fitness[i] = current_fitness
                if current_fitness < best_fitness:
                    best_bird = birds_position[i]
                    best_fitness = current_fitness

    return best_bird, best_fitness
```

Fig 8. Avian Optimization

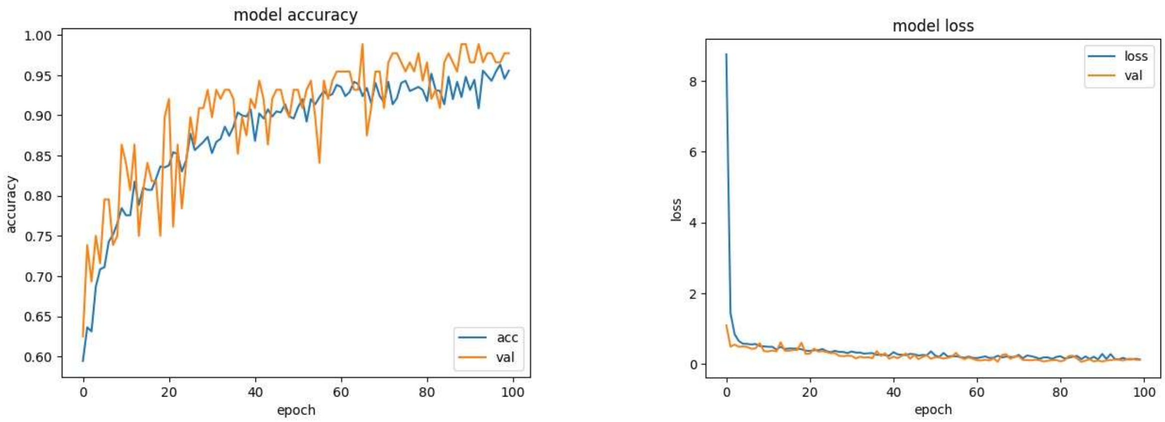


Fig 9. Visualizing Model performance

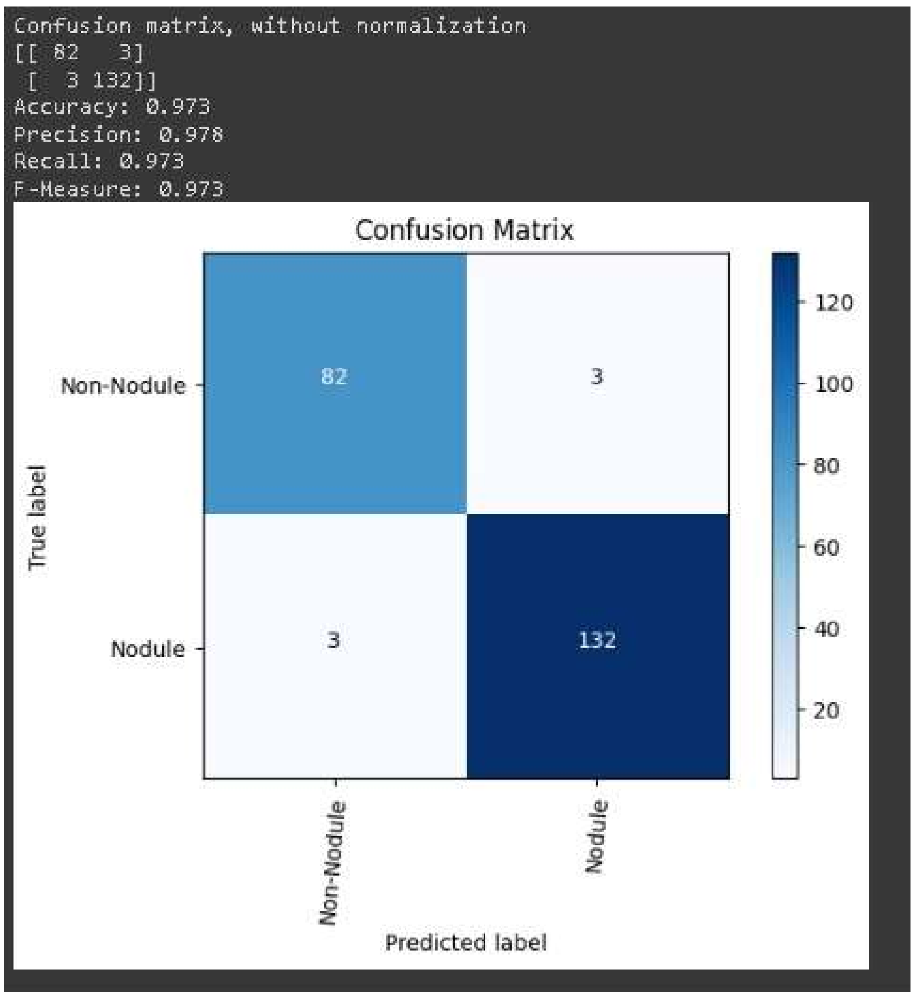


Fig 10. Confusion Matrix

```
image = read_image(sample_img)
plt.imshow(image)
plt.axis('off')  # Turn off axis labels
plt.show()
resized_image = tf.image.resize(
    tf.convert_to_tensor([image]), size=(image_size, image_size)
)
pred= model.predict(resized_image,verbose = 0)
res = np.argmax(pred)
print(type_of_disease[res])
```

Fig 11. Taking Random Input for Testing

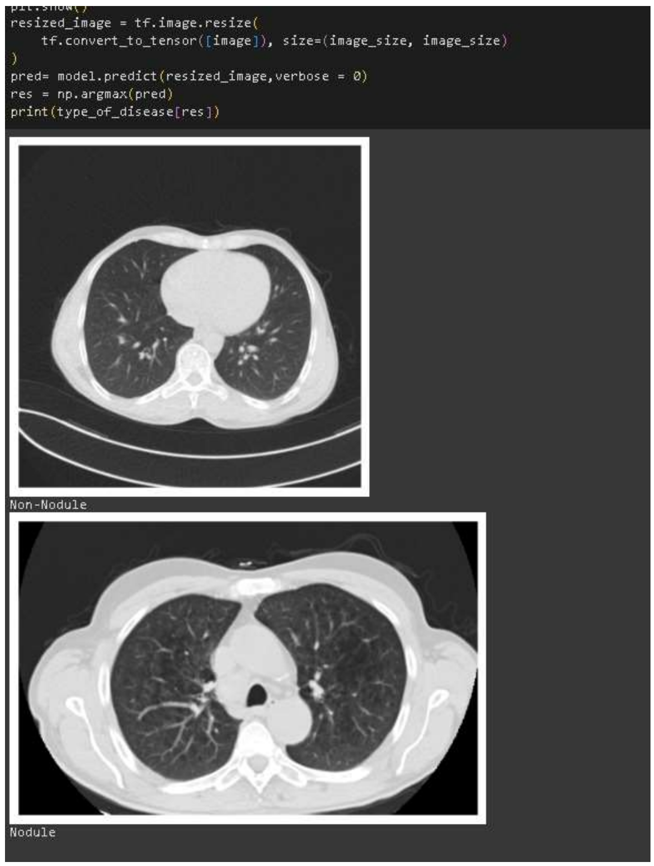


Fig 12. Predicting for new Input

REFERENCES

[1] S. L. A. Lee, A. Z. Kouzani, and E. Hu, "Automated identification of lung nodules," *Deakin Research Online (Deakin University)*, Oct. 2008, doi: https://doi.org/10.1109/mmsp.2008.4665129.

[2] D. Gu, G. Liu, and Z. Xue, "On the performance of lung nodule detection, segmentation and classification," *Computerized Medical Imaging and Graphics*, vol. 89, p. 101886, Apr. 2021, doi: https://doi.org/10.1016/j.compmedimag.2021.101886.

[3] J. Zhang, Y. Xia, H. Cui, and Y. Zhang, "Pulmonary nodule detection in medical images: A survey," *Biomedical Signal Processing and Control*, vol. 43, pp. 138–147, May 2018, doi: https://doi.org/10.1016/j.bspc.2018.01.011

[4] H. Jiang *et al.*, "An Automatic Detection System of Lung Nodule Based on Multigroup Patch-Based Deep Learning Network," *IEEE journal of biomedical and health informatics*, vol. 22, no. 4, pp. 1227–1237, 2018, doi: https://doi.org/10.1109/JBHI.2017.2725903

[5] A. Asuntha and A. Srinivasan, "Deep learning for lung Cancer detection and classification," *Multimedia Tools and Applications*, vol. 79, no. 11–12, pp. 7731–7762, Jan. 2020, doi: https://doi.org/10.1007/s11042-019-08394-3.

[6] R. Golan, C. Jacob, and J. Denzinger, "Lung nodule detection in CT images using deep convolutional neural networks," *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, doi: https://doi.org/10.1109/ijcnn.2016.7727205.

[7] H. Mkindu, L. Wu, and Y. Zhao, "Lung nodule detection in chest CT images based on vision transformer network with Bayesian optimization," *Biomedical Signal Processing and Control*, vol. 85, pp. 104866–104866, Aug. 2023, doi: https://doi.org/10.1016/j.bspc.2023.104866..

[8] H. Mkindu, L. Wu, and Y. Zhao, "3D multi-scale vision transformer for lung nodule detection in chest CT images," *Signal, Image and Video Processing*, vol. 17, no. 5, pp. 2473–2480, Jan. 2023, doi: https://doi.org/10.1007/s11760-022-02464-0.

[9] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," Apr. 2023, doi: https://doi.org/10.1007/s00521-023-08568-z.

[10] H. M. Balaha and A. E.-S. Hassan, "Skin cancer diagnosis based on deep transfer learning and sparrow search algorithm," *Neural Computing and Applications*, vol. 35, no. 1, pp. 815–853, Sep. 2022, doi: https://doi.org/10.1007/s00521-022-07762-9.

[11] R.-Y. Sun, "Optimization for Deep Learning: An Overview," *Journal of the Operations Research Society of China*, vol. 8, no. 2, pp. 249–294, Jun. 2020, doi: https://doi.org/10.1007/s40305-020-00309-6.

[12] J. Mukherjee, A. Chakrabarti, S. H. Shaikh, and M. Kar, "Automatic Detection and Classification of Solitary Pulmonary Nodules from Lung CT Images," *2014 Fourth International Conference of Emerging Applications of Information Technology*, Dec. 2014, doi: https://doi.org/10.1109/eait.2014.64.

[13] Q. Zhang and X. Kong, "Design of Automatic Lung Nodule Detection System Based on Multi-Scene Deep Learning Framework," *IEEE Access*, vol. 8, pp. 90380–90389, 2020, doi: https://doi.org/10.1109/access.2020.2993872.

[14] S. S. Sai Koushik and K. G. Srinivasa, "Detection of respiratory diseases from chest X rays using Nesterov accelerated adaptive moment estimation," *Measurement*, vol. 176, p. 109153, May 2021,

doi: https://doi.org/10.1016/j.measurement.2021.109153.

[15] A. A. A. Setio *et al.*, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge," *Medical image analysis*, vol. 42, pp. 1–13, 2017, doi: https://doi.org/10.1016/j.media.2017.06.015.

[16] W. Zhu, C. Liu, W. Fan, and X. Xie, "DeepLung: Deep 3D Dual Path Nets for Automated Pulmonary Nodule Detection and Classification," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2018, doi: https://doi.org/10.1109/wacv.2018.00079.

[17] Y. Xie *et al.*, "Knowledge-based Collaborative Deep Learning for Benign-Malignant Lung Nodule Classification on Chest CT," *IEEE Transactions on Medical Imaging*, vol. 38, no. 4, pp. 991–1004, Apr. 2019, doi: https://doi.org/10.1109/tmi.2018.2876510.

[18] Amrit Sreekumar, K. S. Nair, Sneha Sudheer, H Ganesh Nayar, and J. J. Nair, "Malignant Lung Nodule Detection using Deep Learning," Jul. 2020, doi: https://doi.org/10.1109/iccsp48568.2020.9182258.

[19] P. Monkam, S. Qi, H. Ma, W. Gao, Y. Yao, and W. Qian, "Detection and Classification of Pulmonary Nodules Using Convolutional Neural Networks: A Survey," *IEEE Access*, vol. 7, pp. 78075–78091, 2019, doi: https://doi.org/10.1109/access.2019.2920980.

[20] Babu Kumar S and M. Kumar, "Detection of Lung Nodules using Convolution Neural Network: A Review," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Jul. 2020, doi: https://doi.org/10.1109/icirca48905.2020.91831.