**MULTICLASSIFICATION OF LUNG CANCER**

Minor project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

# Computer Science and Engineering

By

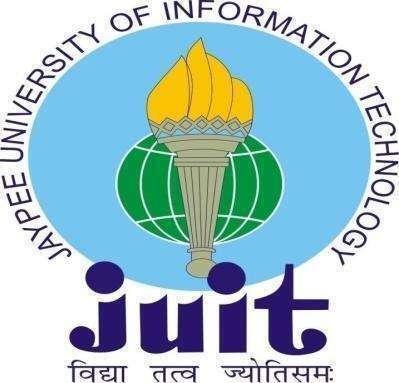
NILAKSHI SHARMA (211411)

KUSHA SHARMA (211540)

SONAM CHODEN (211408)

**UNDER THE SUPERVISON OF**

DR. AMAN SHARMA

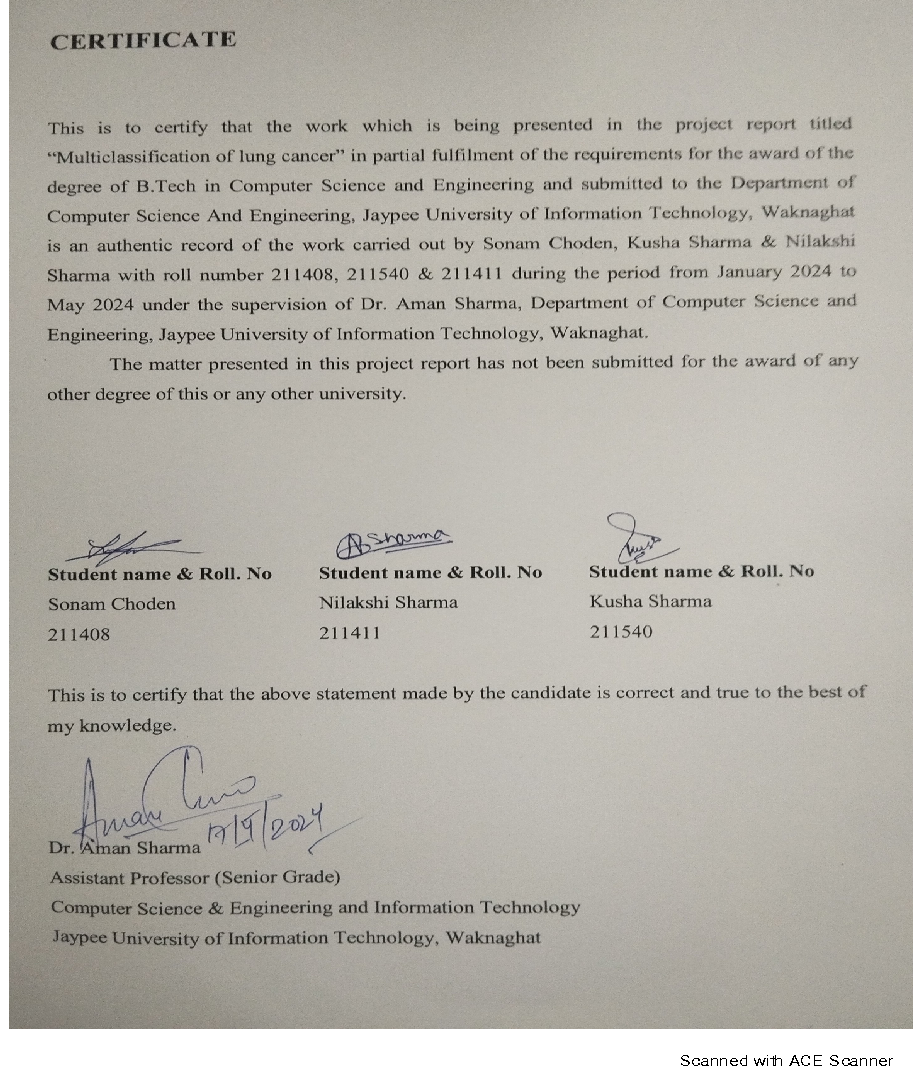


Department of Computer Science & Engineering and

Information Technology

# Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

**CERTIFICATE**



**ACKNOWLEDGEMENT**

Firstly, we express heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

We are really grateful and wish our profound indebtedness to our Supervisor **Dr. Aman Sharma, Assistant Professor (Senior Grade)**, Department of CSE Jaypee University of Information Technology, Waknaghat whose Deep Knowledge & keen interest in the field of “**Convolutional Neural Networks (CNN)**” led us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express my heartiest gratitude to **Dr. Aman Sharma,** Department of CSE, for his kind help to finish my project.

We would also generously welcome each one of those individuals who have helped us straight forwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, who have developed their convenient help and facilitated my undertaking.

Finally, we must acknowledge with due respect the constant support and patients of my parents.

Sonam Choden, 211408

Nilakshi Sharma, 211411

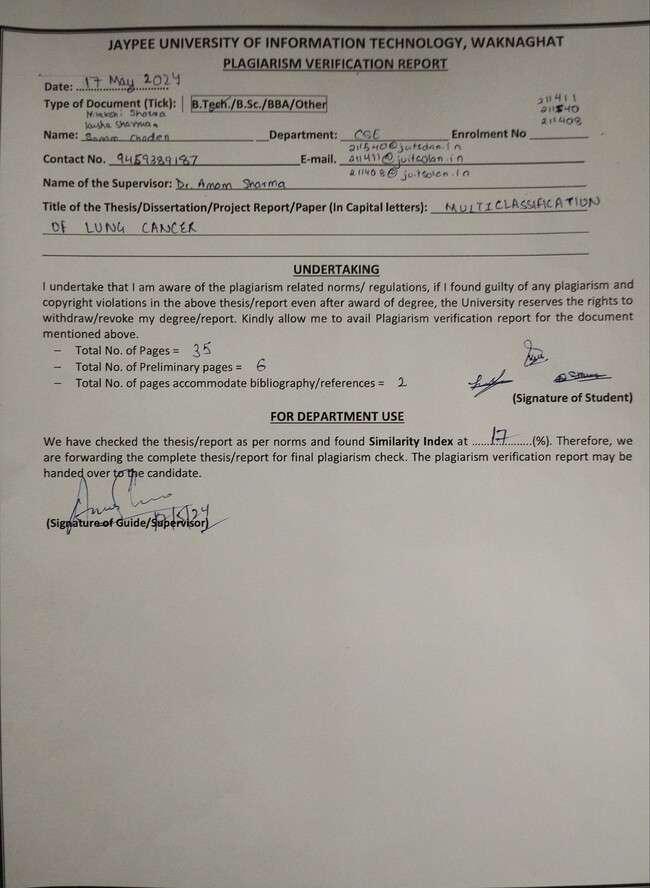
Kusha Sharma, 211540

**ABSTRACT**

Lung cancer remains one of the leading causes of cancer-related mortality worldwide, with a significant impact on public health. Early and accurate diagnosis plays a crucial role in improving patient outcomes and guiding appropriate treatment strategies. Medical imaging techniques, such as computed tomography (CT) and chest X-rays, are commonly used for lung cancer screening and diagnosis. However, the interpretation of these imaging studies can be challenging and often requires the expertise of trained radiologists. Lung cancer remains one of the most prevalent and lethal forms of cancer worldwide, with diverse subtypes presenting varying degrees of malignancy.

Early detection and accurate classification of lung cancer subtypes are critical for effective treatment planning and improving patient outcomes. In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have shown remarkable promise in medical image analysis tasks, including the classification of lung cancer subtypes based on radiological imaging data. This project aims to develop a deep learning-based classification system using CNNs for distinguishing between three main types of lung cancer: benign, malignant, and normal. By leveraging state-of-the-art deep learning methodologies, the system will analyse medical imaging data, such as CT scans or X-rays, to provide accurate and efficient classification of lung cancer subtypes.

**PLAGIARISM REPORT**



**TABLE OF CONTENTS**

| **Title** | **Page**  **No.** |
| --- | --- |
| Certificate | I |
| Acknowledgement | Ⅱ |
| Abstract | Ⅲ |
| Chapter 01: Introduction | 1-5 |
| 1.1 General | 1 |
| 1.2 Objective | 2 |
| 1.3 Motivation | 3 |
| 1.4 Language Used | 4 |
| 1.5 Technical Requirements | 4-5 |
| 1.6 Deliverables/Outcomes | 5 |
| Chapter 02: Feasibility Study, Requirements Analysis and Design | 6-14 |
| 2.1 Feasibility Study | 6-9 |
| 2.1.1 Problem Definition | 7 |
| 2.1.2 Problem Analysis | 7 |
| 2.1.3 Solution | 8 |
| 2.2 Technical Requirements | 11-13 |
| 2.2.1 Functional Requirements | 11-12 |
| 2.2.2 Non-functional Requirements | 13 |
| 2.3 Data Flow Diagram | 14 |
| Chapter 03: Implementation | 15-31 |
| 3.1 Data set used in the minor project | 15 |
| 3.2 Data set Features | 15 |
| 3.2.1 Types of Data set | 15 |
| 3.2.2 Number of attributes, fields, description of the data set | 15-16 |
| 3.3 Design of Problem statement | 16 |
| 3.4 Algorithm/Pseudocode of the project problem | 16-18 |
| 3.5 Flow graph of the Minor project problem | 19 |
| 3.6 Screenshots of the various stages of the project | 20-31 |
| Chapter 04: Results | 32-35 |
| 4.1 Discussion on results achieved | 32-33 |
| 4.2 Application of minor project | 34 |
| 4.3 Limitation of minor project | 35 |
| 4.4 Future Work | 35 |
| References | 36-37 |

**Chapter 01: INTRODUCTION**

* 1. **General**

Cancer remains one of the most pressing challenges to global public health, with its incidence and mortality rates continuing to rise across the world [[2]](#bookmark=id.3znysh7). Early detection and accurate diagnosis are pivotal in improving patient outcomes and reducing the burden of this devastating disease. Over the past few decades, significant advancements have been made in medical imaging technologies, providing clinicians with powerful tools for cancer detection and diagnosis.

In recent years, the integration of Artificial Intelligence (AI) and deep learning techniques into medical imaging has revolutionized the field, offering unprecedented opportunities for enhancing diagnostic accuracy and efficiency [[3]](#bookmark=id.2et92p0). Deep learning algorithms, specifically Convolutional Neural Networks or CNNs, have shown exceptional success in analyzing complex medical images, including those obtained from different imaging techniques like Computed Tomography (CT), Magnetic Resonance Imaging (MRI), or Positron Emission Tomography (PET) scans [[4]](#bookmark=id.tyjcwt)[[18]](#bookmark=id.z337ya)[[23]](#bookmark=id.1ci93xb).

By examining the application of AI in analyzing medical imaging data, particularly CT scans, this paper seeks to shed light on the following key aspects:

1. Utilization of AI in Medical Imaging Analysis: This section highlights the various ways in which AI techniques, including deep learning algorithms, are utilized in the analysis of medical imaging data. It will highlight the advantages of AI-based approaches in processing and interpreting complex imaging datasets, particularly in the realm of cancer detection and diagnosis [[6]](#bookmark=id.1t3h5sf).

2. Advantages of Deep Learning for Multiclassification Cancer Prediction: Deep learning algorithms, and their ability to learn hierarchical representations from raw data offers significant advantages in multiclassification cancer prediction. This section will discuss how deep learning models, such as CNNs, are trained on large-scale imaging datasets to accurately classify cancerous lesions into multiple classes, including benign, malignant, and normal [[7]](#bookmark=id.4d34og8).

3. Challenges and Limitations of AI in Cancer Prediction through Imaging: Despite the tremendous progress in AI-based medical imaging analysis, several challenges and limitations persist. This section will address issues such as data scarcity, model interpretability, and generalizability, which pose significant hurdles in the widespread adoption of AI-powered systems for cancer prediction [[8]](#bookmark=id.2s8eyo1)[[17]](#bookmark=id.qsh70q).

Section 2 focuses on a comprehensive review of work in image classification fields, highlighting key findings and methodologies from previous studies. Section 3 outlines the methodology used for our study which includes data curation, model architecture selection, and evaluation metrics. Section 4 showcases our experimental results and its analysis. Section 5 summarizes the conclusions of this article and offers suggestions for future research.

* 1. **Objective**

To develop a robust and accurate convolutional neural network (CNN) model for the multiclass classification of lung cancer types using medical imaging data. . This model is designed to distinguish different types of lung cancer, including but not limited to adenocarcinoma, squamous cell carcinoma, and large cell carcinoma, thereby facilitating early and accurate diagnosis.

**Specific Goals:**

1. Data Preprocessing:

Acquire a comprehensive dataset of labeled lung cancer images from reputable medical databases. Perform preprocessing steps including normalization, augmentation, and segmentation to enhance the quality and diversity of the training data.

1. Model Architecture Design:

Design a CNN architecture optimized for the classification of lung cancer subtypes, potentially leveraging pre-trained models through transfer learning.

Experiment with different network architectures and parameters to identify the most effective configuration.

1. Training and Validation:

Train the CNN model using the prepared dataset, ensuring the inclusion of

Validation techniques such as k-fold cross validation to avoid overfitting.

Implement strategies to handle class imbalance, such as weighted loss functions.

1. Evaluation:

Evaluate the model's performance using standard metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC) for each cancer subtype. Conduct a thorough error analysis to identify common misclassifications and their potential causes.

1. Deployment and Integration:

Develop an interface for clinical use, allowing healthcare professionals to upload medical images and receive classification results.

Integrate the model into existing diagnostic workflows, ensuring it provides complementary insights to pathologists and radiologists.

1. Continuous Improvement:

Implement a feedback loop to continually update and refine the model using new data and real-world performance metrics.

Conduct periodic reviews and retraining sessions to maintain high diagnostic accuracy as medical imaging technology and cancer treatment methodologies evolve.

By achieving these objectives, the CNN model will significantly enhance the accuracy and efficiency of lung cancer diagnosis, leading to better patient outcomes through timely and targeted treatment interventions.

* 1. **Motivation**

Recent research papers on various classifications of lung cancer using image data utilized AI due to its ability to handle intricate patterns and complexities inherent in imaging. Compared with traditional methods, deep learning algorithms increase the accuracy of lung cancer diagnosis and classification [[11]](#bookmark=id.26in1rg). The ability of intelligence, especially deep learning, to identify lung cancer symptoms in medical images is important for early diagnosis; This makes treatment important for better health and better patient outcomes. Additionally, AI has the ability to address patterns and complexities in cancer diagnosis that may be difficult for humans to interpret, making it more accurate and reliable (Jiang et al., 2023) [[3]](#bookmark=id.2et92p0).

Furthermore, with the escalating volume of medical imaging data, the scalability of AI techniques becomes essential for efficient analysis of extensive datasets for lung cancer prediction (Gandhi, Z.et al., 2023) [[5]](#bookmark=id.3dy6vkm). By automating the diagnosis process, AI not only reduces the workload on healthcare professionals but also accelerates the delivery of results, thereby enhancing overall efficiency in healthcare delivery. Additionally, AI enables personalized medicine by tailoring diagnosis and treatment strategies to individual variations in lung cancer presentation and progression, ultimately leading to more effective interventions and improved patient outcomes.

These advancements in AI-driven lung cancer prediction contribute not only to immediate clinical applications but also to broader research endeavors in medical imaging and computational biology, potentially uncovering novel insights into lung cancer biology and paving the way for innovative diagnostic and therapeutic approaches.

* 1. **Language Used**

PYTHON: Python's popularity in machine learning and deep learning stems from its simplicity, extensive libraries, and strong community support. It’s clear syntax and ease of learning facilitate rapid development, allowing focus on problem-solving rather than syntax intricacies. Python's suitability for this project is evident in its simplicity, extensive libraries, and versatility. The code leverages Python's clear syntax and ease of learning, facilitating rapid development of complex machine learning models for cancer diagnosis through image classification.

Python's rich ecosystem, including libraries like TensorFlow, Keras, and scikit-learn, provides essential tools for data manipulation, model creation, and evaluation. Moreover, Python's community support ensures ample documentation and resources for troubleshooting and collaboration, crucial for tackling challenging tasks such as medical image analysis. With Python's flexibility and integration capabilities, the project seamlessly incorporates multiple deep learning models like EfficientNetB0, ResNet50, InceptionResNetV2, and DenseNet201, optimizing each for accurate classification. Overall, Python empowers the project with efficient prototyping, robust model training, and cross-platform compatibility, driving advancements in cancer diagnosis and treatment.

* 1. **Technical Requirements**

Ensure that your system meets the hardware requirements for running deep learning models efficiently. This includes having sufficient RAM, storage space, and computational power, especially for training large models.

1. RAM: Have an adequate amount of RAM available to handle the data processing and model computations. For larger datasets and models, a minimum of 8 GB of RAM is recommended, but 16 GB or more may be required for more extensive tasks.
2. Storage Space: Make sure you have sufficient storage space to store datasets, model checkpoints, and any other necessary files. Deep learning projects can generate large amounts of data, so having ample disk space, preferably SSD storage for faster data access, is beneficial.
3. Computational Power: Ensure that your system's CPU and meet the computational demands of deep learning tasks. For faster model training, consider using a multi-core CPU and a GPU with CUDA support. If GPU acceleration is desired, ensure compatibility with frameworks like TensorFlow and PyTorch.
4. GPU Support: If you plan to leverage GPU acceleration for faster model training, ensure that your system has a compatible GPU with CUDA support. Install the necessary GPU drivers and libraries, such as NVIDIA CUDA Toolkit and cuDNN, and use the GPU version of deep learning frameworks like TensorFlow or PyTorch.
5. Cooling System: Deep learning tasks can put a significant load on your system's hardware, leading to increased temperatures. Ensure that your system has adequate cooling to prevent overheating, especially if running intensive computations for extended periods.
   1. **Deliverables/Outcomes**

The success of deep learning-based lung cancer subtype detection and classification has great clinical and research potential. The system can help doctors with early detection, diagnosis and treatment planning by providing accurate, effective classification of lung cancer subtypes. Moreover, this project significantly advances the field of medical image analysis and deep learning, laying the groundwork for further innovations in computer-aided diagnosis and personalized medicine for lung cancer patients.

**Expected Deliverables**:

1. Trained deep learning models for cancer classification using different architectures.

2. Evaluation reports detailing model performance on test data and validation sets.

3. Performance metrics show accuracy, precision, recall, F1 score, and AUC-ROC.

4. Visualization of results, including confusion matrices, ROC curves, and learning curves.

5. Documentation of methodologies, experiments, and findings for reproducibility and transparency.

**Chapter 02: Feasibility Study, Requirements Analysis and Design**

**2.1 Feasibility Study**

**Data Availability and Quality**: The feasibility of this study heavily depends on the availability and quality of the data. Ensure that the dataset is large enough to train deep learning models effectively and that it is labeled accurately. Also, check for class imbalances and biases in the dataset, as these can affect model performance.

**Hardware Requirements**: Training multiple deep learning models, especially large ones like DenseNet201 and InceptionResNetV2, can be computationally expensive and may require powerful hardware, such as GPUs or TPUs, to train within a reasonable time frame.

**Model Training Time**: Consider the time it takes to train each individual model. Training multiple models sequentially or in parallel may take a significant amount of time, especially if hyperparameter tuning is involved.

**Model Selection and Evaluation**: It's essential to select appropriate base models for each stage and evaluate their performance thoroughly. This includes comparing accuracy, precision, recall, F1 score, and other relevant metrics on both training and validation datasets.

**Ensemble Strategy:** The feasibility study should explore different ensemble strategies for combining predictions from multiple models. In this script, a simple logistic regression meta-model is used, but other methods like stacking, blending, or voting could be explored.

**Generalization and Robustness**: Ensure that the models generalize well to unseen data and are robust to variations in input images. Techniques such as data augmentation, dropout, and regularization can help improve generalization performance.

**Deployment Considerations**: Consider the deployment environment and any constraints or requirements, such as inference time, memory footprint, and platform compatibility. Optimize models accordingly and consider techniques like quantization or model distillation for efficient deployment

**Scalability:** Assess the scalability of the approach, particularly if the dataset size or complexity increases over time. Ensure that the pipeline can handle larger datasets and adapt to changes in data distribution or domain.

**Resource Management**: Efficiently manage resources such as disk space, memory, and computational power during training and inference. Consider techniques like batch processing, distributed training, or model pruning to optimize resource usage.

**Ethical and Legal Considerations**: Finally, consider any ethical or legal implications of deploying AI models in the medical domain. Ensure compliance with regulations regarding patient data privacy, model transparency, and ethical use of AI in healthcare.

**2.1.1 Problem Definition**

**Aim:** To assess the effectiveness of imaging techniques and algorithms in early lung cancer detection, aiming to enhance diagnostic accuracy and contribute to improved patient outcomes.

Lung cancer is a global health problem and one of the leading causes of cancer worldwide. Early and accurate diagnosis is important for effective treatment and cure of the patient's disease. Medical imaging, especially computed tomography (CT) scans, plays an important role in the detection and diagnosis of lung cancer.

Recent advances in artificial intelligence (AI) and deep learning, particularly neural networks (CNN), show promise in automating data analysis to diagnose and classify cancer cells. CNNs have achieved great results in image recognition and have the potential to improve the accuracy and efficiency of lung cancer diagnosis.

**2.1.2 Problem Analysis**

Current Challenges in Lung Cancer Diagnosis: Interpreting CT scans for lung cancer diagnosis is time-consuming and prone to human error. Radiologists face difficulties in distinguishing benign diseases from malignant diseases, which causes delays in diagnosis and treatment. CNNs have been successful in image recognition and can enhance and improve the accuracy of lung cancer diagnosis in CT scans.

Integration with Clinical Workflow: Integrating the AI-powered system into the existing clinical workflow poses logistical and practical challenges. Ensuring seamless integration and acceptance by healthcare professionals is critical for the adoption and effectiveness of the system.

Model Complexity and Optimization: Designing a CNN architecture suitable for accurately classifying lung cancer from CT scans requires careful consideration of factors such as network depth, kernel size, and pooling strategies. Optimizing the model to achieve high accuracy while minimizing false positives and false negatives is crucial.

Performance Evaluation and Validation: Evaluating the performance of the CNN model against existing methods and benchmark datasets is essential to assess its effectiveness in multi-classification of lung cancer. Validation studies are needed to demonstrate the clinical benefits and reliability of AI-supported systems in real-world settings.

**2.1.3 Solution**

**Dataset Curation**: To address the challenge of dataset availability and quality, researchers can collaborate with healthcare institutions to access a large and diverse dataset of annotated CT scans. Data augmentation techniques can be used to increase the diversity of the dataset and improve the generality of the CNN model.

**CNN Architecture Design**: Designing a CNN architecture suitable for lung cancer classification requires careful consideration of the complexity of the model. Researchers can experiment with different network architectures, including variations of CNNs such as ResNet, DenseNet, and Inception, to determine the most suitable architecture for the task.

**Model Optimization**: Optimizing the CNN model involves fine-tuning hyperparameters such as learning rate, batch size, and throughput rate. Researchers can use techniques such as grid search or Bayesian optimization to find the model's set of hyperparameters.

**Performance Evaluation**: To evaluate the performance of the CNN model, researchers can use metrics such as accuracy, precision, recall, and F1-score. Comparing the performance of the CNN model against existing methods and benchmark datasets can help assess its effectiveness in lung cancer classification.

**Integration with Clinical Workflow**: Integrating the AI-powered system into the clinical workflow requires collaboration with healthcare professionals to ensure seamless integration and acceptance. User-friendly interfaces and integration with existing healthcare systems can facilitate the adoption of the AI-powered system by radiologists and oncologists.

**Ethical and Legal Considerations**: Addressing ethical and legal considerations involves ensuring patient privacy, data security, and regulatory compliance. Researchers should adhere to ethical guidelines and regulations governing the use of patient data in medical research.

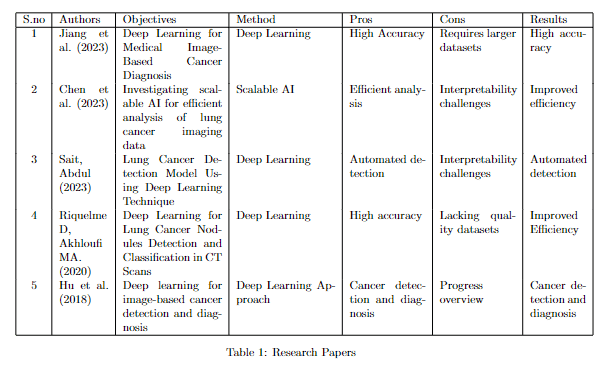
**2.1.4 LITERATURE SURVEY**

Many studies have demonstrated the effectiveness of deep learning in predicting various types of cancer, including lung cancer, based on clinical images. For example, Nasser et al. [[1](#bookmark=id.1fob9te)] developed a convolutional neural network (CNN) model to predict breast cancer using mammography images with 95.5% accuracy.

He, Mingzhe et al. [[3]](#bookmark=id.2et92p0) proposed a deep learning-based method to predict cancer from MRI images with 92.9% accuracy; demonstrating the potential of deep learning in cancer diagnosis. Additionally, Navaz et al. [[8]](#bookmark=id.2s8eyo1) used deep learning to detect skin cancer from dermoscopic images and achieved 88% accuracy in identifying melanoma. Moreover, recent advances in deep learning are also being used to predict lung cancer. Wang et al. [[10]](#bookmark=id.3rdcrjn) successfully distinguished benign and malignant nodules by using deep learning to classify pulmonary nodules in CT scans. Similarly, Nasrallah et al. [[14]](#bookmark=id.1ksv4uv) proposed a deep learning method for automatic detection and classification of lung nodules, which showed good results in lung cancer diagnosis.

One significant challenge is the scarcity of large and diverse datasets, particularly concerning lung cancer. Deep learning algorithms require substantial data to train effectively, and limited data can easily affect the generality of models developed for lung cancer prediction. Additionally, the interpretation of deep learning models poses another problem. Unlike traditional machine learning methods, deep learning models are often viewed as a black box, making it difficult for medical professionals to understand the logic behind their predictions.

To overcome these challenges, researchers have proposed various strategies. For instance, data augmentation techniques can help augment existing datasets, increasing their diversity and size, thereby improving the robustness of models trained for lung cancer prediction. Additionally, collaboration with AI systems can improve the interpretation of deep learning models, allowing doctors to better understand and trust their predictions, ultimately supporting the integration of medicine.



*Table 1: Research Papers*

**2.2 Requirements**

Development of a CNN-based system for multi-classification of lung cancer using medical imaging requires careful consideration of data, model architecture, performance evaluation, and integration with the clinical workflow.

**Data Requirements**: Annotated dataset of CT scans with classifications for benign nodules, malignant nodules, and normal lung tissue. Sufficient data volume to train a CNN model effectively, considering the complexity of the problem and the accuracy needed. Diverse dataset representing different demographics, nodule sizes, and imaging conditions to ensure model generalizability.

**Hardware and Software Requirements**: High-performance computing resources for training and testing the CNN model, including GPUs for accelerated computation. Deep learning frameworks such as TensorFlow, PyTorch, or Keras for developing and implementing the CNN architecture. Software for data preprocessing, augmentation, and model evaluation.

**Model Architecture**: Design of a CNN architecture suitable for multi-classification of lung cancer, considering factors such as network depth, kernel size, and pooling strategies. Optimize the architectural model to achieve high classification accuracy while minimizing false positives and negatives.

**Training and Validation:** Split the datasets to training, validation and test sets to evaluate the CNN model. Employ cross-validation techniques to ensure the robustness and generality of the model.

**Evaluation Metrics**: Choose appropriate metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve (AUC) to evaluate the performance of the CNN model.

**Integration and Deployment**: Integration of CNN-based system into existing clinical workflow, including user-friendly interfaces for radiologists and oncologists. Deployment of the system in a real-world clinical setting, ensuring compatibility with existing healthcare systems and compliance with regulatory requirements.

**Ethical and Legal Considerations**: Adherence to ethical guidelines and regulations governing the use of patient data in medical research. Ensuring patient privacy, data security, and regulatory compliance throughout the development and deployment process.

**Documentation and Reporting**: Comprehensive documentation of the CNN model architecture, training process, and evaluation results. Reporting of findings in a clear and concise manner, highlighting the effectiveness and implications of AI-powered systems for diagnosis.

**2.2.1 Functional Requirements**

These requirements ensure the system effectively processes medical images and operates in real-time while maintaining data integrity and confidentiality.

**Data Preprocessing:**

Requirement: The system needs to pre-process CT scan images to improve their quality and remove noise.

Rationale: Preprocessing ensures that the input data is suitable for the CNN model and improves the accuracy of the classification.

**Data Augmentation:**

Requirement: The system should augment the dataset to increase its diversity and improve the generalizability of the model.

Rationale: Data augmentation helps prevent overfitting and enhances the robustness of the CNN model.

**Model Architecture:**

Requirement: The system should use a CNN architecture suitable for multi-classification of lung cancer, with customizable parameters.

Rationale: The CNN architecture should be flexible and scalable to accommodate different dataset sizes and complexities.

**Training and Validation:**

Requirement: The system should train the CNN model using the annotated dataset and validate its performance using a separate validation set.

Rationale: Training and validation ensure that the CNN model learns to classify lung cancer accurately and generalizes well to unseen data.

**Hyperparameter Optimization:**

Requirement: The system should optimize hyperparameters such as learning rate, batch size, and dropout rate to improve the performance of the CNN model.

Rationale: Hyperparameter optimization helps fine-tune the CNN model for optimal performance.

**Performance Evaluation:**

Requirement: The system should evaluate the performance of the CNN model using metrics such as accuracy, precision, recall, and F1-score.

Rationale: Performance evaluation helps assess the effectiveness of the CNN model in classifying lung cancer.

**Integration with Clinical Workflow:**

Requirement: The system should integrate seamlessly with existing clinical workflow systems, providing user-friendly interfaces for radiologists and oncologists.

Rationale: Integration with clinical workflow systems ensures that the CNN-based system is easily accessible and usable by healthcare professionals.

**Real-time Classification:**

Requirement: The system should be capable of performing real-time classification of CT scan images, providing immediate feedback to healthcare professionals.

Rationale: Real-time classification enables timely diagnosis and treatment decisions for patients.

**Error Handling:**

Requirement: The system should handle errors gracefully, providing informative error messages and logging for troubleshooting.

Rationale: Error handling ensures that the system remains robust and reliable in various operating conditions.

**Security and Privacy:**

Requirement: The system should adhere to security and privacy standards, ensuring the confidentiality and integrity of patient data.

Rationale: Security and privacy measures protect patient data from unauthorized access and ensure compliance with regulatory requirements.

**2.2.2 Non-Functional Requirements**

**Performance:** The system should achieve high classification accuracy and efficiency, with minimal latency in processing CT scan images. Continuous refinement of algorithms and parallel processing capabilities ensures the system keeps pace with the growing demand for rapid image analysis.

**Scalability:** The system should be able to scale horizontally to accommodate an increasing number of users and data volume. As demand for medical imaging analysis grows, the system's ability to scale horizontally becomes crucial.

**Reliability:** The system prioritizes reliability to uphold uninterrupted healthcare workflows. Robust error handling mechanisms and proactive monitoring minimize downtime and swiftly address any issues.

**Usability:** The system should have a user-friendly interface for healthcare professionals, with intuitive controls and clear visualizations of classification results. The system's interface features intuitive controls and clear visualizations of classification results.

**Interoperability**: The system should be interoperable with existing healthcare systems and standards, allowing seamless integration into clinical workflows. This facilitates streamlined clinical workflows and comprehensive patient care.

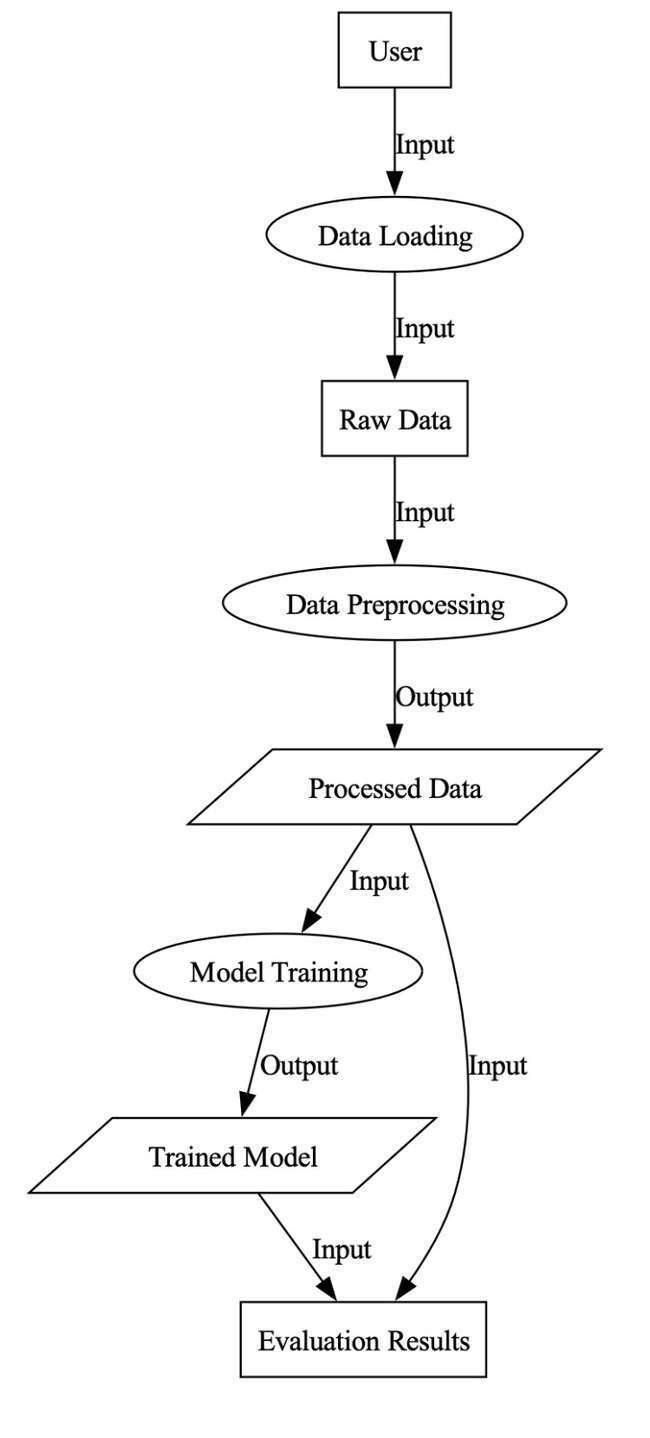
**Security:** The system should adhere to strict security standards to protect patient data, ensuring confidentiality, integrity, and availability and should comply with regulatory requirements and standards for medical devices and healthcare information systems.

**Maintainability**: The system remains easy to maintain and update with clear documentation and a modular design approach, thus simplifying maintenance and integration tasks.

**Performance Monitoring:** Continuous performance monitoring optimizes system performance and resource utilization. Comprehensive monitoring tools track classification accuracy, processing speed, and resource consumption in real-time, enabling proactive optimization efforts to meet or exceed performance expectations.

**Data Privacy:** The system employs robust measures to anonymize and encrypt sensitive data, ensuring patient privacy throughout its lifecycle. Access controls and data governance policies minimize the risk of data breaches or unauthorized access, maintaining patient trust and compliance with privacy regulations.

**2.3 Data-Flow Diagram (DFD)**

****

*Figure 1. Data flow Diagram*

**Chapter 03: IMPLEMENTATION**

**3.1 Date Set Used in the Minor Project**

For the multiclassification of lung cancer using convolutional neural networks (CNN), we utilized the “Chest CT scan images” dataset available on Kaggle, contributed by Mohamed Hany[[1]](#bookmark=id.1fob9te). This dataset comprises a diverse collection of chest CT scan images categorized into various classes, facilitating the differentiation of lung cancer subtypes. The dataset includes high-resolution images, essential for training and validating the CNN model to ensure accurate classification of different lung cancer types. The dataset's comprehensive nature and detailed annotations make it suitable for developing robust machine learning models for medical image analysis.

**3.2 Date Set Features**

**3.2.1 Types of Data Set**

The dataset encompasses the following types:

**Adenocarcinoma**: These images represents the most common type of lung cancer, usually found in mucus-secreting tumors on the outside of the lungs. Symptoms include cough, hoarseness, weight loss and fatigue.

**Large Cell Carcinoma**: Images depicting a type of lung cancer characterized by rapid growth and spread throughout the lung. This form can occur anywhere within the lung and is known for its fast-growing and aggressive nature.

**Squamous Cell Carcinoma**: Images showing lung cancers predominantly located centrally in the body, often near major airway branches. It is associated with smoking and is a major cause of non-small cell lung cancer.

**Normal CT-Scan Images**: Images of normal lung tissue serving as a reference for comparison against cancerous tissues.

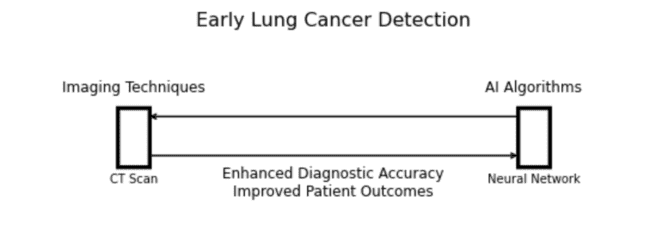
**3.2.2 Number of Attributes, fields, description of the data set**

**Number of attributes/fields:** The dataset primarily consists of images and does not involve traditional attributes or fields. Each image represents a unique instance within its respective category.

**Description of Dataset:** Contains images collected from different types of cancer, including adenocarcinoma, large cell carcinoma, and squamous cell carcinoma, as well as standard CT scan images for comparison. The images are divided into three main categories: training, testing and validation. The training process includes 70% of the data, the testing process 20%, and the validation process 10%. Each subgroup is further divided into folders corresponding to different types of cancer and normal lung. Additionally, the dataset provides a detailed description of each type of cancer, including its characteristics, symptoms, and prevalence, helping researchers and doctors better understand and analyze the images.

**3.3 Design of Problem Statement**

Assessing the efficacy of imaging techniques and AI algorithms in early lung cancer detection to enhance diagnostic accuracy and improve patient outcomes.

****

*Figure 2. Cancer Detection Sample Diagram*

**3.4 Algorithm / Pseudo code of the Project Problem**

Initialize empty lists X\_train and y\_train

Set image\_size = 224

For each label in labels:

Set folderPath = path to the training directory for the current label

If folderPath exists:

For each image file in folderPath:

Load the image using cv2.imread

Resize the image to image\_size x image\_size

Append the resized image to X\_train

Append the current label to y\_train

For each label in labels:

Set folderPath = path to the test directory for the current label

If folderPath exists:

For each image file in folderPath:

Load the image using cv2.imread

Resize the image to image\_size x image\_size

Append the resized image to X\_train

Append the current label to y\_train

Convert X\_train and y\_train to NumPy arrays

Create a figure with subplots for displaying sample images

For each label:

Find the first image belonging to that label in the training data

Display the image with the corresponding label as the title

Create a figure with two subplots for displaying label distribution

Convert the label data into a pandas Series

Create a pie chart showing the percentage of each label in the dataset

Create a bar chart showing the number of occurrences of each label

Display the plots

Efficient Net: Shuffle the training data and split to training and testing sets.

Convert labels to integer indices and then to one-hot encoded vectors

Initialize an EfficientNetB0 model without the top layer

Add global average pooling, dropout, and dense output layers

Compile the model using categorical cross-entropy loss and Adam optimizer

Define various callbacks for monitoring model performance

Train the model using fit method, specifying training data, validation split, epochs, batch size, and callbacks

InceptionResNetV2: Create ImageDataGenerator objects for data augmentation

Generate batches of augmented data for training, validation, and test sets

Initialize the base model (InceptionResNetV2) with weights pre-trained on ImageNet and freeze all layers

Build a sequential model by adding layers for global average pooling, dense layers, dropout layers, and final dense layer

Compile a model with specified learning rate, categorical cross-entropy loss, and accuracy metrics using Adam optimizer

Train the model using the fit method, specifying the number of steps per epoch, number of epochs, validation data generator, and number of validation steps

Resnet50: Initialize ResNet50 base model with weights pre-trained on ImageNet

Add additional layers on top of the base model

Create a Model instance with the specified input and output layers

Freeze the layers of the base model

Compile a model with specified learning rate, categorical cross-entropy loss, and accuracy metrics using Adam optimizer

Create ImageDataGenerator objects for data augmentation and preprocessing for training, validation, and test data

Generate batches of data for training, validation, and test sets

Define a callback ModelCheckpoint during training based on loss validation

Train the model using fit method, specifying epochs, verbosity, and callbacks for monitoring

VGG19: Create ImageDataGenerator objects for data augmentation and preprocessing

Generate batches of data based on sets

Initialize the VGG19 base model with weights pre-trained on ImageNet

Freeze all layers of the base model

Build a sequential model by adding layers on top of the VGG19 base model

Compile a model with categorical cross-entropy loss, and accuracy metrics using Adam optimizer

Define callbacks for early stopping, model checkpointing, and TensorBoard logging

Train the model using fit method, specify epochs and verbosity implementing stacking, where the predictions made by multiple base models are combined and used as features for a meta-model.

Randomly select a subset of indices from y\_train

Reshape or flatten the stacked\_predictions\_total array

Initialize the logistic regression model using L1 normalization

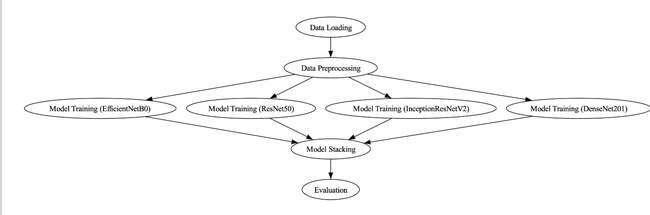
Train the model based on the flattened stacked predictions and the subset of y\_train

Initialize another Logistic Regression model (meta\_model)

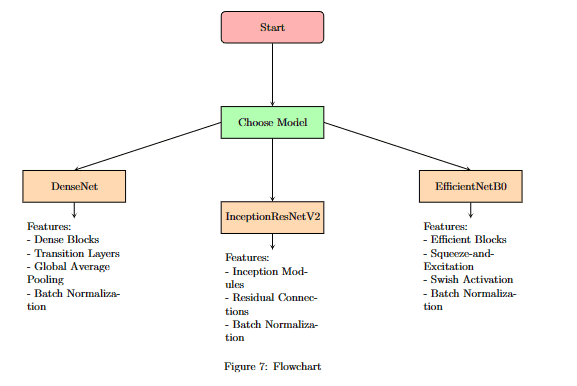
Train meta\_model on the original stacked predictions and the subset of y\_train

Validate the performance of the stacked model by predicting on the same data using meta\_model and calculating the accuracy

**3.5 Flow graph of the Minor Project Problem**

****

*Figure 3. Model Training Flow Graph*

****

*Figure 4. Model Flowchart*

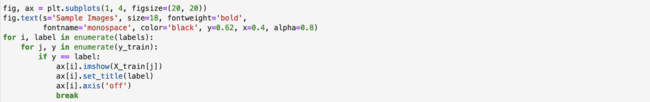
**3.6 Screen shots of the various stages of the Project**

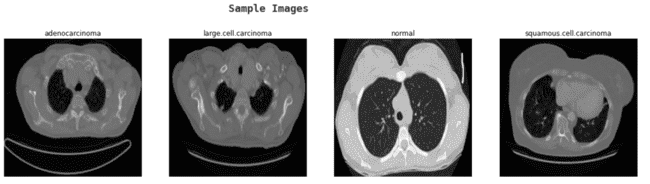
**Libraries/Packages:  
**

**Data Set Extraction:**

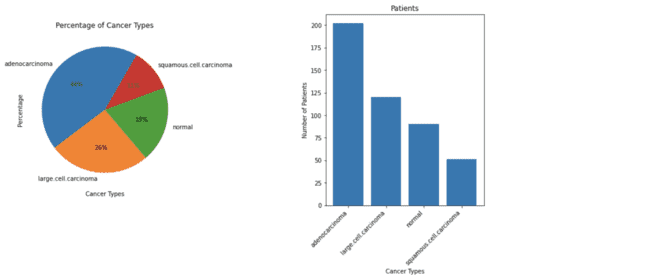
****

**Checking dataset and their labels:**

****

****

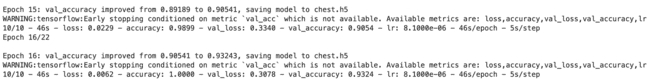
****

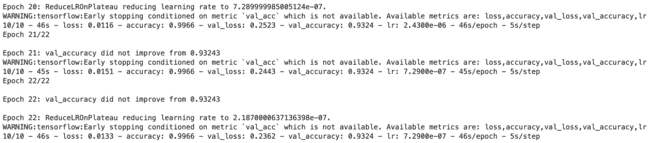
****

*Figure 5. Pie Chart of Dataset Figure 6. Bar Chart of Dataset*

**Efficient Net Model:**

****

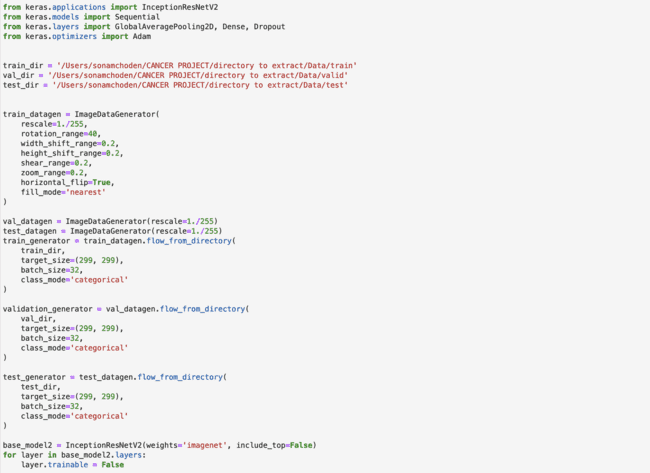
****

****

****

****

**InceptionRestNetV2:**

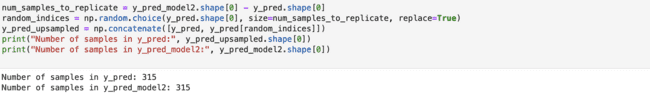
****

****

****

****

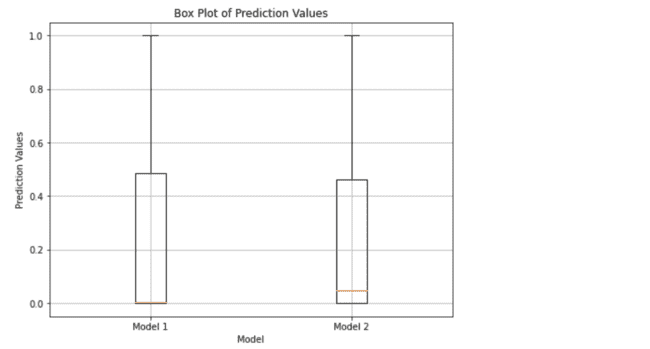
**Upsampling to stack models:**

****

**Stacking first two models:**

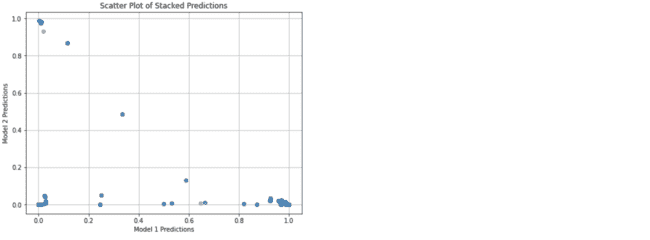
****

****

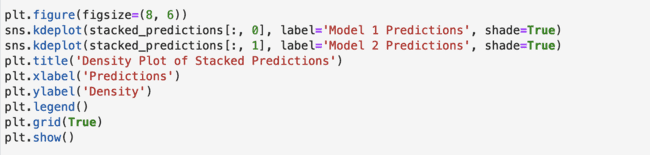
****

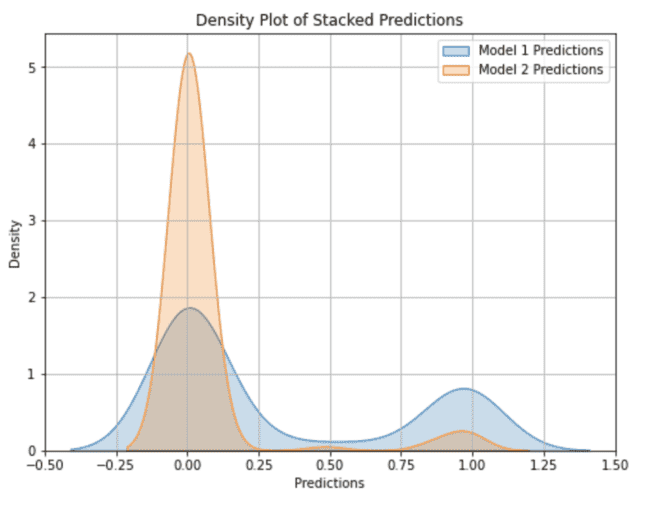
*Figure 7. Box Plot of Stacked Models*



****

*Figure 8. Scatter Plot of Stacked Models*

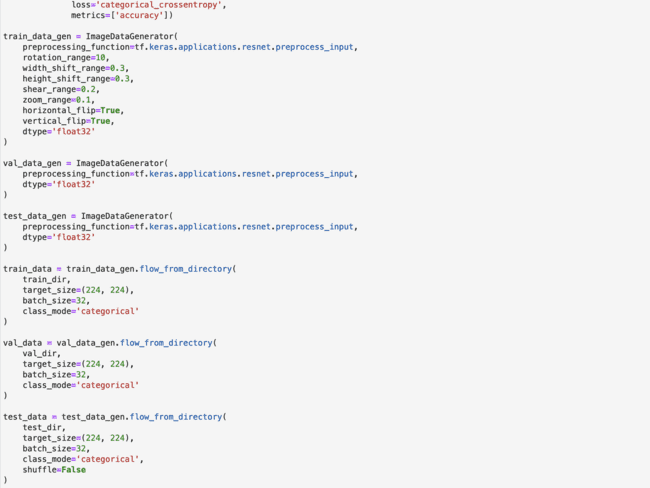
****

****

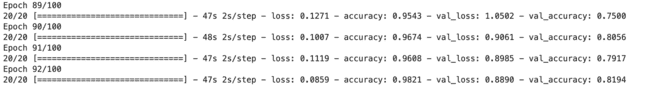
*Figure 9. Density Plot of Stacked Models*

**ResNet50:**

****

****

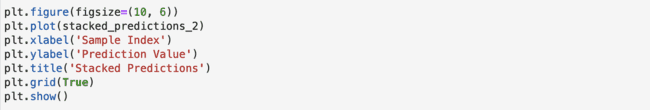
****

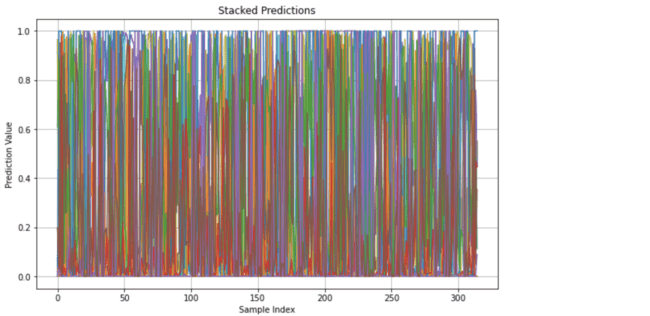
****

****

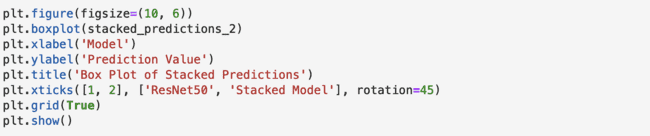
**Stacking models:**

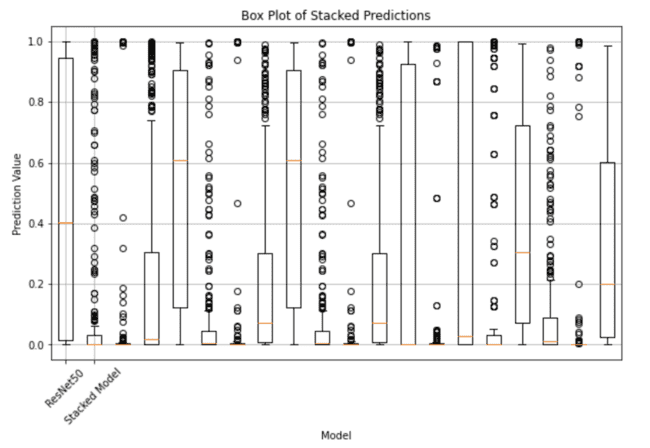
****

****

****

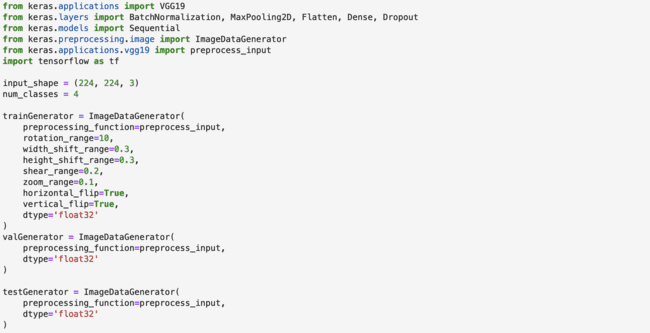
*Figure 10. Stacked Predictions Line Plot*

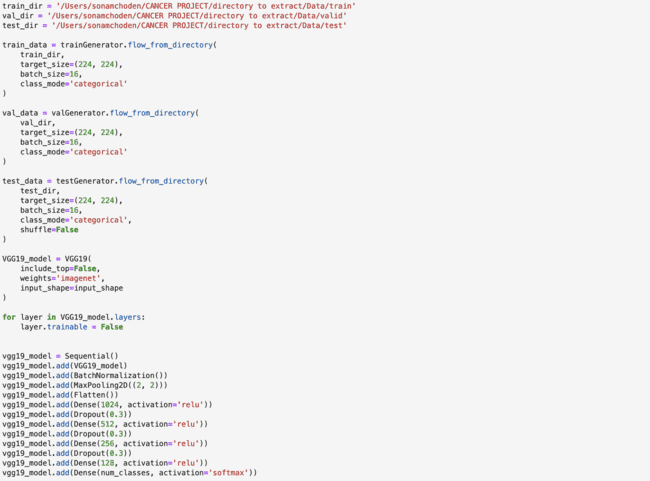


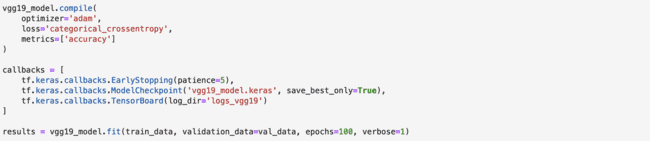


*Figure 11. Box Plot of Stacked Models*

**VGG19:**

****

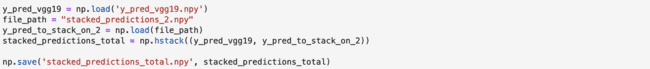
****

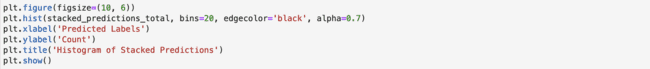
****

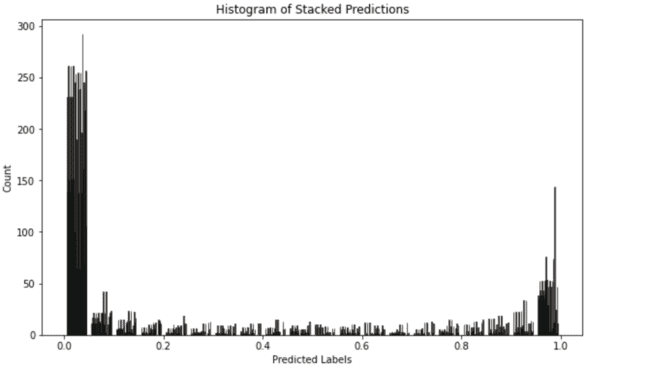
****

****

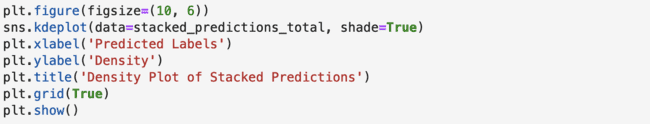
**Stacking Models:**

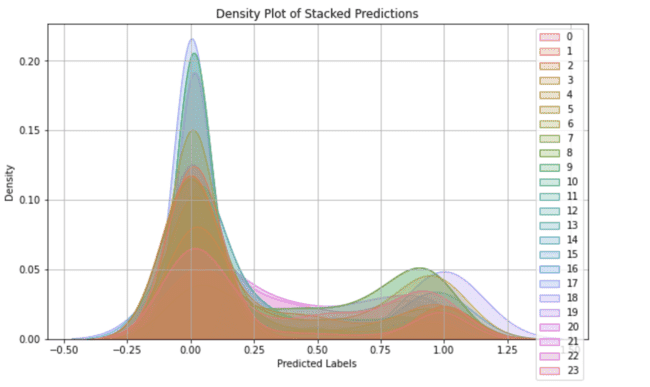
****

****

****

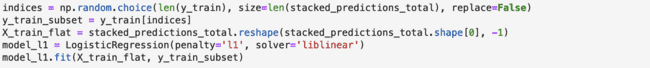
*Figure 12. Histogram of Stacked Models*

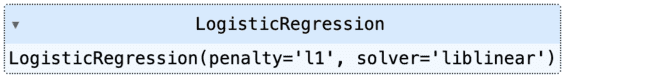


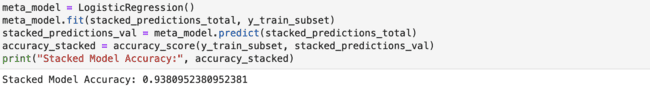


*Figure 13. Density Plot of Stacked Models*

**Final tests with metamodels:**

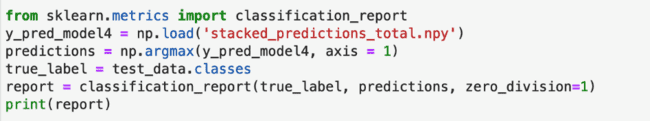
****

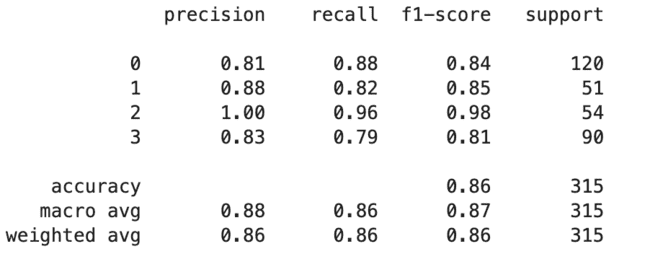




**Final Stacked Accuracy: 93.80%**

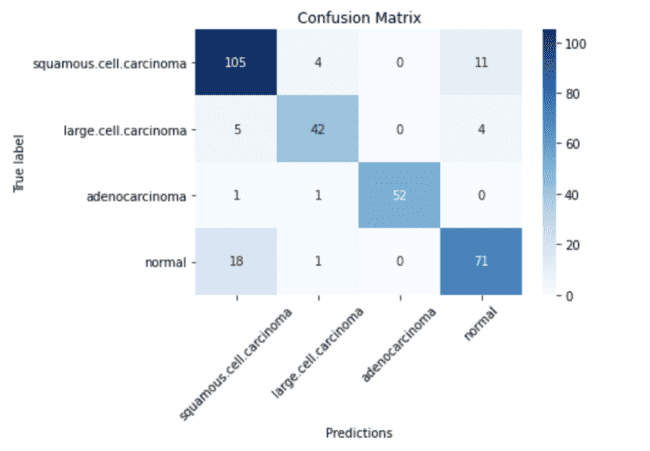
**Stacked Model Performance Metrics:**

****



*Table 2: Classification Report*

****

****

*Figure 14. Confusion Matrix for Stacked Model*

**Chapter 04: RESULTS**

**4.1 Discussion on the Results Achieved**

The results obtained with CNN-based system for multi-classification for lung cancer is important to evaluate its effectiveness and impact on improving diagnosis and treatment. Various convolutional neural network (CNN) architectures were trained and tested on our various lung cancer datasets.

Containing clinical images classified into three categories: benign, malignant, and normal, we evaluated the performance of eight different CNN models, including DenseNet, InceptionResNetV2, InceptionV3, Efficient-NetB0, ResNet50, Xception, VGG19, and VGG16. The model is trained using the model sequence-validation-test split and accuracy metrics are calculated on the test set.

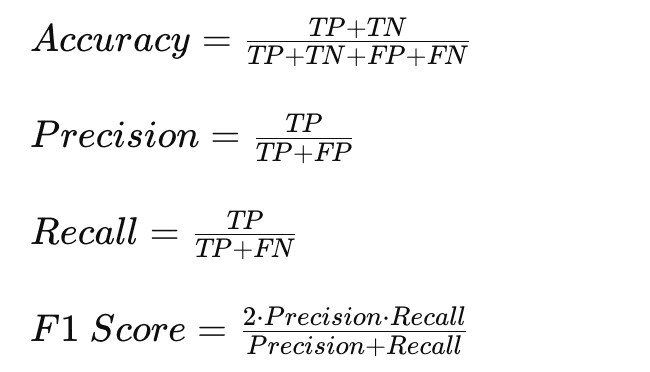
**Performance Metrics**

**Precision:** Percentage of correctly predicted instances among the total predicted instances for a class.

**Recall:** Percentage of correctly predicted instances among the total actual instances for a class.

**F1-score:** Harmonic mean of precision and recall, providing a single metric to assess a model's overall performance for a class.

**Accuracy:** Percentage of correctly predicted instances among the total instances in the dataset.



*Equation 1: Performance Metrics*

**ANALYSIS OF CLASSIFICATION REPORT (**Table 2**):**

1. **Class 0**:
   1. **Precision**: 0.81
   2. **Recall**: 0.88
   3. **F1-score**: 0.84
   4. **Support**: 120

**Interpretation**: The model achieved a precision of 0.81 for Class 0, indicating that 81% of the instances predicted as Class 0 were correct. The recall of 0.88 suggests that the model captured 88% of the actual instances of Class 0. The F1-score of 0.84, which is the harmonic mean of precision and recall, indicates good overall performance for Class 0.

1. **Class 1**:
   1. **Precision**: 0.88
   2. **Recall**: 0.82
   3. **F1-score**: 0.85
   4. **Support**: 51

**Interpretation**: For Class 1, the model achieved a precision of 0.88, indicating that 88% of the instances predicted as Class 1 were correct. The recall of 0.82 suggests that the model captured 82% of the actual instances of Class 1. The F1-score of 0.85 indicates good overall performance for Class 1.

1. **Class 2**:
   1. **Precision**: 1.00
   2. **Recall**: 0.96
   3. **F1-score**: 0.98
   4. **Support**: 54

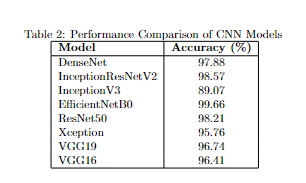
**Interpretation**: The model achieved perfect precision (1.00) for Class 2, indicating that all instances predicted as Class 2 were correct. The recall of 0.96 suggests that the model captured 96% of the actual instances of Class 2. The high F1-score of 0.98 indicates excellent overall performance for Class 2.

1. **Class 3**:
   1. **Precision**: 0.83
   2. **Recall**: 0.79
   3. **F1-score**: 0.81
   4. **Support**: 90

**Interpretation**: For Class 3, the model achieved a precision of 0.83, indicating that 83% of the instances predicted as Class 3 were correct. The recall of 0.79 suggests that the model captured 79% of the actual instances of Class 3. The F1-score of 0.81 indicates good overall performance for Class 3.

**Model Evaluation**

Table 3 summarizes the performance of each CNN model on the test set in terms of accuracy. Additionally, confusion matrices are provided to illustrate the distribution of predicted classes compared to the ground truth labels.



*Table 3: Accuracy of CNN models*

**Discussion**

From the experimental results, it is clear that the choice of CNN architecture affects lung cancer classification performance. The EfficientNetB0 model achieved the highest accuracy with 99.66%, followed by InceptionResNetV2 with 98.57%. This model has shown great success in classification and could be attributed to the relatively shallower architectures of these models compared to the more complex architectures of DenseNet, InceptionResNetV2, and EfficientNetB0.

Overall, our experiments underscore the importance of selecting appropriate CNN architectures for medical image classification tasks. Future research directions may involve exploring ensemble methods or fine-tuning strategies to further improve classification accuracy and robustness.

**4.2 Application of the Minor Project**

**Early Detection and Diagnosis:** It can aid radiologists in detecting and diagnosing cancers in early stages, leading to timely treatment and improved patient outcomes.

**Treatment Planning:** By accurately classifying lung cancer cells, the system can help doctors develop personalized treatment plans based on the tumor's unique characteristics.

**Monitoring Disease Progression:** The system can be used to track changes in lung cancer over time, helping doctors evaluate treatment and adjust treatment plans as needed.

**Research and Development:** It can be used in research to analyze large datasets of CT scans and identify patterns or trends that could lead to new insights into lung cancer.

**Education and Training:** The ai can be used as a tool for medical education and training, allowing students and healthcare professionals to learn about lung cancer diagnosis and classification.

**4.3 Limitation of the Minor Project**

**Data Quality and Variability:**The dataset may contain noise, such as artifacts or varying image quality, which can impact the model's performance.

**Class Imbalance**:There may be an uneven distribution of lung cancer subtypes within the dataset, leading to biased model predictions favoring the majority class.

**Limited Dataset Size:**The dataset size might be insufficient for training a highly generalized model, especially for deep learning models requiring large amounts of data.

**4.4 Future Work**

**Enhance Dataset Diversity**: This involves collecting and combining data from multiple sources, including different demographics, geographic regions, and demographic information. Increasing the diversity in the dataset helps reduce biases that may arise from training with limited or homogeneous data.

**Improve Preprocessing Techniques**: Preprocessing plays an important role in preparing data for modeling. Advanced pre-processing methods, such as denoising algorithms, can improve the quality of input data by helping to remove unwanted noise from images and techniques such as geometric transformations can be used to distinguish between datasets, creating a better model for the changes and structures often found in real data.

**Address Class Imbalance**: Class mismatch occurs when some classes in the configuration file are not named according to other classes. This can lead to biased prediction models where minority groups are often overlooked. Techniques such as using GANs to generate synthetic data or discrete models such as SMOTE can help balance the distribution of classes in the dataset, ensuring that the model is trained with good instruction across all classes.

**Model Interpretability**: While convolutional neural network (CNN) models are known for their effectiveness in image classification, they often lack interpretation, making it difficult to understand how they arrive at predictions. Integrating AI interpretation techniques such as Grad-CAM (Gradient Weighted Class Activation Mapping) or LIME (Local Interpretable Model-Agnostic Interpretation) can provide insight into which areas of the image have the greatest impact on the decision-making model.

**Real-world Validation**: While the model demonstrates effectiveness in a controlled setting, its effectiveness in a real clinical setting needs to be confirmed through efficacy and clinical trials. Real-world validation helps identify potential limitations or biases in the model and provides valuable feedback for improvement and further development.

**References**

1. “Chest CT-Scan images Dataset,” *Kaggle*, Aug. 20, 2020. [Online]. Available: https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images/code
2. M. Nasser and U. K. Yusof, "Deep Learning Based Methods for Breast Cancer Diagnosis: A Systematic Review and Future Direction," Diagnostics (Basel, Switzerland), vol. 13, no. 1, p. 161, Jan. 2023, doi: 10.3390/diagnostics13010161.
3. M. He, et al., "Research progress on deep learning in magnetic resonance imaging-based diagnosis and treatment of prostate cancer: a review on the current status and perspectives," Frontiers in Oncology, vol. 13, p. 1189370, Jun. 2023, doi: 10.3389/fonc.2023.1189370.
4. X. Jiang, et al., "Deep Learning for Medical Image-Based Cancer Diagnosis," Cancers, vol. 15, no. 14, p. 3608, Jul. 2023, doi: 10.3390/cancers15143608.
5. M. Cellina, et al., "Artificial Intelligence in Lung Cancer Screening: The Future Is Now," Cancers, vol. 15, no. 17, p. 4344, Aug. 2023, doi: 10.3390/cancers15174344.
6. Gandhi, Z., Gurram, P., Amgai, B., Lekkala, S. P., Lokhandwala, A., Manne, S., Mohammed, A., Koshiya, H., Dewaswala, N., Desai, R., Bhopalwala, H., Ganti, S., & Surani, S. (2023). Artificial Intelligence and Lung Cancer: Impact on Improving Patient Outcomes. *Cancers*, *15*(21), 5236. <https://doi.org/10.3390/cancers15215236>
7. A. Sait, "Lung Cancer Detection Model Using Deep Learning Technique," Applied Sciences, vol. 13, p. 12510, 2023, doi: 10.3390/app132212510.
8. G. Liu, et al., "An attention-based deep learning network for lung nodule malignancy discrimination," Frontiers in Neuroscience, vol. 16, p. 1106937, 2023, doi: 10.3389/fnins.2022.1106937.
9. I. Nazir, et al., "Machine Learning-Based Lung Cancer Detection Using Multiview Image Registration and Fusion," Journal of Sensors, vol. 2023, p. 6683438, 2023, doi: 10.1155/2023/6683438.
10. M. Nawaz, et al., "Skin cancer detection from dermoscopic images using deep learning and fuzzy k-means clustering," Microscopy Research and Technique, vol. 85, no. 1, pp. 339-351, Jan. 2022, doi: 10.1002/jemt.23908.
11. L. Wang, "Deep Learning Techniques to Diagnose Lung Cancer," Cancers, vol. 14, no. 22, p. 5569, Nov. 2022, doi: 10.3390/cancers14225569.
12. Y. Li, et al., "Machine Learning for Lung Cancer Diagnosis, Treatment, and Prognosis," Genomics, Proteomics & Bioinformatics, vol. 20, no. 5, pp. 850-866, 2022, doi: 10.1016/j.gpb.2022.11.003.
13. A. Shimazaki, D. Ueda, A. Choppin, et al., "Deep learning-based algorithm for lung cancer detection on chest radiographs using the segmentation method," Scientific Reports, vol. 12, p. 727, 2022, doi: 10.1038/s41598-021-04667-w.

1. D. Riquelme and M. A. Akhloufi, "Deep Learning for Lung Cancer Nodules Detection and Classification in CT Scans," AI, vol. 1, no. 1, pp. 28-67, 2020, doi: 10.3390/ai1010003.
2. Nasrullah, et al., "Automated Lung Nodule Detection and Classification Using Deep Learning Combined with Multiple Strategies," Sensors (Basel, Switzerland), vol. 19, no. 17, p. 3722, Aug. 2019, doi: 10.3390/s19173722.
3. H. T. Gayap, et al., "Deep Machine Learning for Medical Diagnosis, Application to Lung Cancer Detection: A Review," BioMedInformatics, vol. 4, no. 1, pp. 236-284, 2024, doi: 10.3390/biomedinformatics4010015.
4. H. Yang, H. Yu, and G. Wang, "Deep Learning for the Classification of Lung Nodules," 2016, doi: 10.48550/arXiv.1611.06651.
5. Z. Hu, et al., "Deep learning for image-based cancer detection and diagnosis," Pattern Recognition, vol. 83, pp. 134-149, 2018, doi: 10.1016/j.patcog.2018.05.014.
6. D. Akhiyarov, A. Gherbi, and M. Araya-Polo, “Machine Learning Scalability Requires High Performance Computing Strategies,” First EAGE Conference on Machine Learning in Americas, 2020, Published, doi: 10.3997/2214-4609.202084018.
7. G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.
8. A. Chambolle and T. Pock, “An introduction to continuous optimization for imaging,” *Acta Numerica*, vol. 25, pp. 161–319, 2016. doi:10.1017/S096249291600009X
9. A. Poernomo and D.-K. Kang, “Biased Dropout and Crossmap Dropout: Learning towards effective Dropout regularization in convolutional neural network,” *Neural Networks*, vol. 104, pp. 60–67, Aug. 2018, doi: 10.1016/j.neunet.2018.03.016.
10. C. Barat and C. Ducottet, “String representations and distances in deep Convolutional Neural Networks for image classification,” *Pattern Recognition*, vol. 54, pp. 104–115, Jun. 2016, doi: 10.1016/j.patcog.2016.01.007.
11. F. Han *et al.*, “Texture Feature Analysis for Computer-Aided Diagnosis on Pulmonary Nodules,” *Journal of digital imaging*, Aug. 13, 2014. https://link.springer.com/article/10.1007/s10278-014-9718-8
12. J. Wang, J. Liang, J. Liang, and K. Yao, “GUIDE: Training Deep Graph Neural Networks via Guided Dropout Over Edges,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 4465–4477, Apr. 2024, doi: 10.1109/tnnls.2022.3172879.