**LUNG AND COLON CANCERS DETECTION**

**ABSTRACT**

Lung and colon cancers are among the most common types of cancer worldwide, and accurate detection and classification of these cancers is crucial for effective treatment and it improves patient outcomes. In this project Median Filtering for Preprocessing and **Convolutional Neural Network** models like **Sequential model** which achieved 98% **and ResNet models** which achieved 97% for the detection of cancer through **histopathological images**. Moreover, there are **machine learning models** like **SVM, Decision tree** which can classify the cancers, but those are not much accurate. This project demonstrates that the system outperforms existing methods in terms of accuracy. The proposed algorithm has the potential to significantly enhance the efficiency and accuracy of cancer detection and reduces the **diagnosing time**, ultimately contributing to improved treatment quality and patient outcomes**.**

**Keywords:** Convolutional Neural Network,Median Filtering,Sequential model, Histopathological Images.

**CHAPTER 1**

**INTRODUCTION**

**­­­­­­­­­­­­­­**

**INTRODUCTION**

Lung and colon cancers are among the most common types of cancer worldwide, with cancer rates of 11.4% and 10% respectively in 2020. Although these cancers are widespread, there is a possibility of synchronous occurrence of lung and colon cancers (LCC), resulting in high mortality rates of 18% and 9.4% respectively for each cancer. Therefore, precise detection of these cancer sub-types is necessary to improve therapeutic strategies in the early stages of cancer.

Non-invasive diagnostic methods for lung cancer include computed tomography (CT) images and radiography, while CT colonoscopy and flexible sigmoidoscopy are common for colon cancer. However, these non-invasive techniques may not always reliably differentiate specific sub-categories of these cancers, necessitating slightly invasive methods like histopathology for accurate detection and better treatment quality. Manual grading of histopathological images (HI) can be challenging, time-consuming, and prone to error. Thus, automatic image processing approaches for LCC images have proven beneficial for easing the burden on pathologists.

In recent years, advancements in biomedical applications have expanded opportunities for the diagnosis and treatment of various types of cancer. Utilizing artificial intelligence (AI) technology, including deep learning (DL) techniques, offers the potential to accelerate decision-making and enhance diagnostic accuracy. This project aims to analyze data containing the HI of LCCs, employing a Convolutional Neural Network (CNN) approach for precise cancer detection and classification. Additionally, diagnosing cancer can be a lengthy process that relies on the judgments of multiple physicians, especially in the early stages.

AI applications have made significant contributions to the medical field, such as early diagnosis of biomedical images, disease prediction, and medical emergencies. DL techniques can extract latent features in medical images, providing timely cancer identification and differentiating between various stages.

* Development of a new LCCD algorithm consisting of median filtering (MF)-based preprocessing and CNN classification, designed specifically for LCC detection. To our knowledge, this LCCD approach is novel in the literature.
* Implementation of MobileNet for extracting relevant features from HIs, critical for precise cancer detection. Additionally, the CNN Sequential model excels at identifying occurrences of LCC in HIs, improving detection accuracy and contributing to better treatment outcomes.

**CHAPTER 2**

**LITERATURE SURVEY**

**­­­­­­­­­­­­­­­**

**LITERATURE SURVEY**

[1] Mizuho et al. employed machine learning algorithms for diagnosing three types of lung cancer. They extracted features using homology-based image and texture analysis methods, finding that machine learning with homology-based image features performed better than texture analysis. This demonstrates the value of different feature extraction techniques in improving diagnosis.

[2] Vinod et al. designed a method to identify pulmonary nodules using the watershed algorithm and Gabor filter. Features extracted from these methods were then categorized using SVM. This approach underscores the potential of combining traditional image processing techniques with machine learning for accurate diagnosis.

[3] Mesut et al. developed the DarkNet-19 model for training lung and colon cancer datasets from scratch. They applied the Equilibrium algorithm to select efficient features from the data, which were then classified using Support Vector Machines (SVM). This approach emphasizes the importance of feature selection in achieving better classification results.

[4] Mehedi et al. proposed a deep learning model for diagnosing five classes of lung and colon cancer, incorporating 2D Fourier and 2D wavelet features. This innovative approach achieved an accuracy of 96.33%, demonstrating the effectiveness of these features in cancer diagnosis.

[5] Sanidhya et al. presented a CNN Pre-Trained Diagnostic Network for diagnosing lung and colon cancer. Their approach uses a shallow CNN architecture trained on histological images, yielding impressive accuracy rates of 96% and 97% for diagnosing colon and lung cancer, respectively. The shallow CNN architecture leverages pre-trained models to improve accuracy in diagnosing these types of cancers.

[6] Mumtaz et al. introduced a capsule network with multiple inputs for diagnosing abnormal cell carcinoma of the lung and colon. This approach combines convolutional layers with the capsule network to improve detection capabilities. The use of capsule networks, which retain spatial relationships between features, adds value in the accurate diagnosis of cancers.

[7] SHAHID et al. optimized AlexNet, a deep learning model, for diagnosing lung and colon cancer cells. By modifying its four essential layers and training it on the dataset, they achieved an accuracy of 89%. This demonstrates the flexibility of AlexNet and its potential in cancer diagnosis.

[8] Dipanjan et al. generated a 1D CNN network to classify small cell lung tumors. By combining hybrid features with clinical features, their approach exceeded traditional machine learning techniques, highlighting the potential for hybrid models in lung cancer diagnosis.

Overall, these studies showcase the breadth and depth of approaches being explored in the diagnosis of lung and colon cancers. By combining different deep learning architectures, feature extraction methods, and machine learning algorithms, researchers are achieving high accuracy rates and advancing the field of medical diagnostics.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**­­­­­­­­­­­­­­­**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

The existing system likely includes **traditional methods of cancer detection**, such as **manual grading of histopathological images by pathologists**, as well as **non-invasive techniques** like computed **tomography (CT) and radiography for lung cancer, and CT colonoscopy** for colon cancer. However, the limitations of these existing approaches, including the **time-consuming nature** of manual grading, the potential for error, and the inability of non-invasive techniques to accurately identify sub-categories of these cancers. Moreover, there are **machine learning models** which can classify the cancers, but those are not much accurate. So, there is a need for enhanced algorithms and techniques to address the challenges associated with the existing system, particularly in the context of accurately detecting and classifying lung and colon cancers using of complex histopathological images.

**DISADVANTAGES:**

* The early detection and classification of lung and colon cancers using histopathological images (HI) and deep learning techniques hold great promise; however, there are several challenges and disadvantages to consider:
* Manual Grading of HI: The manual grading of histopathological images can be a time-consuming and labor-intensive process for pathologists. This method is also subject to human error, leading to inconsistencies in diagnosis.
* Non-Invasive Techniques: Although non-invasive diagnostic techniques, such as computed tomography (CT) and radiography, can provide valuable information, they may not always accurately identify specific sub-categories of lung and colon cancers.
* Limited Machine Learning Accuracy: While machine learning models have been employed to classify cancers, their performance may vary and may not always achieve the level of accuracy required for reliable diagnosis. These models may struggle with complex histopathological images.
* Lack of Precision in Sub-Category Detection: Traditional methods and existing machine learning models may lack the precision needed to accurately classify the different sub-categories of lung and colon cancers, which is crucial for targeted treatment plans.
* Dependency on High-Quality Data: The effectiveness of machine learning models depends heavily on the quality of available data. Inconsistencies in data quality or lack of sufficient annotated datasets can hinder model performance.

**3.2 PROPOSED SYSTEM**

In the proposed system, the Convolutional Neural Network for Lung and Colon Cancer Detection method, aims to enhance the accuracy and efficiency of cancer detection and classification in histopathological images. This new approach incorporates advanced components such as the median filtering (MF) approach for noise removal, a deep convolutional neural network (DCNN) based models like Sequential and ResNet model for cancer detection, the proposed system seeks the grading of histopathological images and accurately identify sub-categories of lung and colon cancers. This system represents a significant advancement in the field of medical imaging and demonstrates superior performance compared to existing approaches.

**ADVANTAGES:**

* Enhanced Noise Removal: The median filtering (MF) approach effectively removes noise from histopathological images, leading to clearer and more accurate image analysis.
* Sequential CNN Model: Employing a CNN Sequential model for cancer detection allows the system to accurately classify and grade histopathological images. This approach provides reliable identification of sub-categories of lung and colon cancers.
* ResNet CNN Model: Employing a CNN ResNet model for cancer detection allows the system to accurately classify and grade histopathological images. This approach provides reliable identification of sub-categories of lung and colon cancers
* Improved Accuracy: The combination of advanced feature extraction and noise reduction techniques enhances the system's overall accuracy in cancer detection, outperforming existing methods.
* Efficiency in Diagnosis: By automating the grading and classification of histopathological images, the proposed system significantly reduces the time and effort required by pathologists, leading to faster diagnosis and improved patient outcomes.
* Superior Performance: This system demonstrates superior performance compared to traditional approaches, as it leverages advanced deep learning components to analyse complex images and detect cancer more effectively.
* Potential for Early Detection: The proposed system's ability to accurately grade and classify histopathological images can facilitate earlier detection of lung and colon cancers, enabling timely intervention and treatment.

**CHAPTER 4**

**SOFTWARE REQUIREMENTS AND SPECIFICATIONS**

**­­­­­­­­­­­­­­­**

**SOFTWARE REQUIREMENTS AND SPECIFICATIONS**

**4.1 HARDWARE REQUIREMENTS:**

* Processor : Intel Core i5 or higher or AMD Ryzen 5 or higher.
* RAM : minimum of 8 GB

**4.2 SOFTWARE REQUIREMENTS:**

* Operating System : Windows 7/8/10/11, MacOS
* IDE : VS Code, Google Colab
* Front-End : HTML, CSS, JAVASCRIPT
* Framework : Flask

**4.3 LIBRARIES USED:**

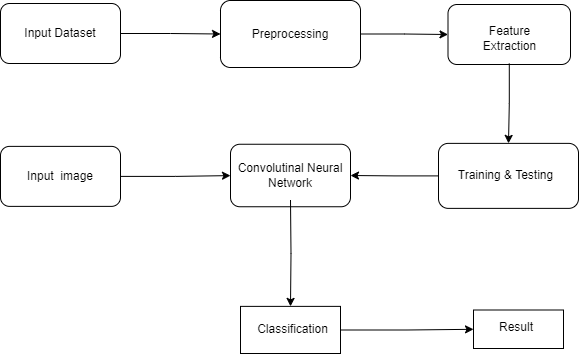
* **OpenCV (Open-Source Computer Vision Library):** OpenCV is a popular open-source library for computer vision and image processing tasks. It provides a wide range of functions and algorithms for tasks such as image manipulation, feature detection, object tracking, and facial recognition. With its extensive documentation and cross-platform support, OpenCV is widely used in various applications, including robotics, augmented reality, and medical imaging.
* **TensorFlow (2.15.0):** TensorFlow is a popular open-source deep learning framework developed by Google. It allows developers to create, train, and deploy machine learning models for a variety of tasks, such as image and speech recognition, natural language processing, and more. TensorFlow 2.15.0 includes enhancements for ease of use and performance improvements.
* **scikit-learn (1.2.2):** Scikit-learn is a popular open-source machine learning library that provides tools for data preprocessing, model training, and evaluation. It supports a variety of algorithms such as classification, regression, clustering, and dimensionality reduction. Version 1.2.2 includes improvements and bug fixes.
* **Pandas (2.0.3):** Pandas is a powerful open-source data manipulation and analysis library for Python. It provides data structures such as DataFrames and Series for handling tabular data efficiently. Version 2.0.3 includes new features and improvements in performance and memory usage.
* **Gradio (4.26.0):** Gradio is an open-source library for creating interactive web interfaces for machine learning models. It allows developers to quickly build user-friendly applications for model inference and testing. Version 4.26.0 includes enhancements and new features.
* **Keras (2.15.0):** Keras is a high-level neural network library that provides an easy-to-use interface for building, training, and deploying deep learning models. Keras can be used as a frontend to TensorFlow. Version 2.15.0 includes compatibility with the latest TensorFlow release and performance improvements.
* **Matplotlib (3.7.1):** Matplotlib is a widely used open-source plotting library for Python. It allows users to create a variety of static, animated, and interactive plots and visualizations. Version 3.7.1 offers various enhancements and new features for plotting data.
* **Flask (2.2.5):** Flask is a lightweight web framework for Python that allows developers to create web applications quickly and efficiently. Version 2.2.5 includes performance improvements and bug fixes.
* **NumPy (1.25.2):** NumPy is an open-source numerical computing library for Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these data structures. Version 1.25.2 includes performance optimizations and new features.

**CHAPTER 5**

**SYSTEM DESIGN**

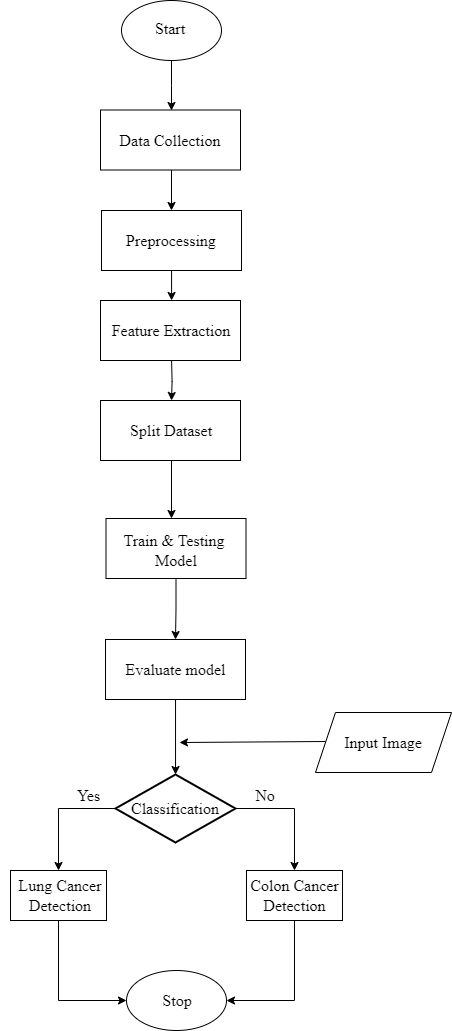
**SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE:**

****

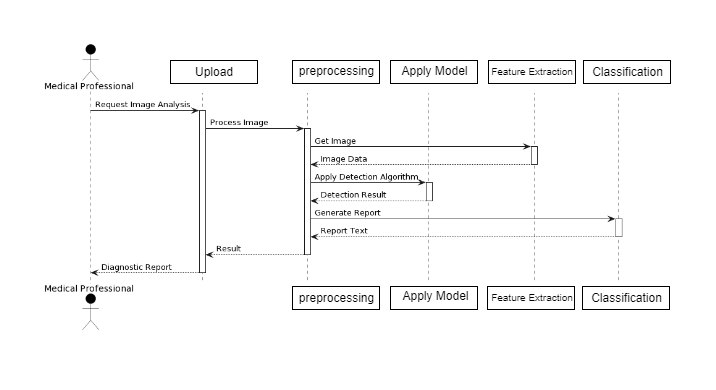
**Fig 5.1: System Architecture**

**5.2 SYSTEM FLOWCHART:**

****

**Fig 5.2: System Flow Chart**

**5.3 SYSTEM UML SEQUENCE DIAGRAM:**

****

**Fig 5.3: UML Sequence Diagram**

**CHAPTER 6**

**IMPLEMENTATION**

**­­­­­­­­­­­­­­­**

**IMPLEMENTATION**

**6.1 SEQUENTIAL ALGORITHM:**

The Sequential model is a straightforward way to build and train neural networks for image classification using Keras, a high-level neural network API in TensorFlow. The model takes extracted features from images as input and predicts probabilities for each class as output. It involves defining the architecture by creating a Sequential model, then adding layers such as a dense layer with 512 units and ReLU activation, a dropout layer with a rate of 0.5 to prevent overfitting, and an output dense layer with softmax activation for classification. Once the model's architecture is set, it is trained using the input features and labels, specifying the number of epochs and batch size. After training, the model is evaluated on a test dataset to assess its performance and obtain metrics like accuracy. Once the model is trained and evaluated, it can be deployed for inference on new unseen data, providing predicted probabilities for each class based on the input features.

**6.1.1 STEP BY STEP PROCESS OF SEQUENTIAL:**

**Input:**

* Batch of images from the Dataset.

**Output:**

* Predicted probabilities of each class.

**Steps:**

**1. Define the architecture of the CNN model:**

- Create a Sequential model.

- Add layers sequentially:

- Add a Dense layer with 512 units and ReLU activation.

- Add a Dropout layer with a dropout rate of 0.5 to prevent overfitting.

- Add a Dense layer with the number of output classes and softmax activation for classification.

**2. Train the model:**

- Train the model using the extracted features as input and labels as output.

- Set the number of epochs and batch size for training.

**3. Evaluate the model:**

- Evaluate the trained model on the test dataset to assess its performance.

- Obtain metrics such as accuracy to measure the model's effectiveness.

**4. Use the model for inference:**

- Once trained and evaluated, deploy the model for making predictions on new unseen data.

- Provide input features to the model and obtain predicted probabilities for each class.

**6.2 RESNET50 ALGORITHM:**

ResNet50, or Residual Network with 50 layers, is a deep convolutional neural network (CNN) architecture designed to efficiently train very deep neural networks by introducing residual blocks with skip connections. These connections allow the network to bypass some layers, alleviating the vanishing gradient problem and improving training stability. ResNet50 consists of convolutional, pooling, and fully connected layers and begins with simple feature detection in the earlier layers before progressing to complex pattern recognition in deeper layers. The model can be used as a powerful feature extractor, and its pre-trained weights on large datasets like ImageNet provide a strong starting point for various tasks. ResNet50 is widely used in computer vision applications, such as image classification, object detection, and segmentation, due to its ability to balance depth with computational efficiency and state-of-the-art performance.

**6.2.1 STEP BY STEP PROCESS OF RESNET50:**

**Input:**

- Training data (train\_gen)

- Image shape: (224, 224, 3)

- Number of epochs: 6

- Batch size: 36

**Output:**

- Trained CNN model

- Model performance metrics (e.g., accuracy)

**Steps:**

**1. Define the input layer:**

- Create an input layer with the specified image shape.

**2. Define convolutional and pooling layers:**

- Add Conv2D layers with varying numbers of filters, kernel sizes, and ReLU activation functions.

- Add MaxPooling2D layers for down-sampling after each set of Conv2D layers.

**3. Define fully connected layers:**

- Flatten the output from the last pooling layer.

- Add Dense layers with ReLU activation functions.

- Make Dropout layers between Dense layers to prevent overfitting.

**4. Define the output layer:**

- Add a Dense output layer with softmax activation for multi-class classification.

**5. Create the model:**

- Combine the defined layers to create the custom CNN model.

**6. Compile the model:**

- Use an optimizer such as Adamax with a specified learning rate.

- Specify the loss function as categorical cross-entropy.

- Specify metrics such as accuracy for evaluation.

**7. Train the model:**

- Train the model using the training data, specifying the number of epochs and batch size.

- Validate the model using validation data.

**8. Evaluate the model:**

- Monitor the training and validation performance metrics, such as accuracy, over the epochs.

**6.3 VGG ALGORITHM:**

VGG, or Visual Geometry Group, is a deep convolutional neural network (CNN) architecture that was introduced by researchers from the University of Oxford in 2014. It is notable for its simplicity and effectiveness in image classification tasks. The model architecture consists of a series of convolutional layers with small kernel sizes (3x3) and rectified linear unit (ReLU) activation functions, followed by max-pooling layers for down-sampling. This approach enables the network to capture intricate details and patterns in images. After several convolutional layers, the model includes fully connected layers for classification. The VGG model has multiple configurations, such as VGG16 and VGG19, which denote the number of weight layers in each configuration. These models have performed exceptionally well in image recognition benchmarks like ImageNet, providing a solid foundation for further advancements in deep learning for computer vision tasks.

**6.3.1 STEP BY STEP PROCESS OF VGG:**

**Input:**

- Training data (train\_gen)

- Validation data (valid\_gen)

- Image shape: (224, 224, 3)

- Number of epochs: 6

**Output:**

- Trained VGG model

- Model performance metrics (e.g., accuracy)

**Steps:**

**1. Define the input layer:**

- Create an input layer with the specified image shape of `(224, 224, 3)`.

**2. Define convolutional and pooling layers:**

- Add a series of Conv2D layers with a kernel size of `3x3` and ReLU activation, increasing the number of filters in each layer.

- Use a padding of "same" in the Conv2D layers to maintain the input shape.

- After each block of Conv2D layers, add a MaxPooling2D layer with a pooling size of `(2, 2)` for down-sampling.

**3. Define fully connected layers:**

- Flatten the output from the last pooling layer using `Flatten()`.

- Add multiple Dense layers with ReLU activation functions for further processing.

**4. Define the output layer:**

- Add a Dense output layer with softmax activation and a number of units equal to the number of classes for multi-class classification.

**5. Create the model:**

- Combine the defined layers to create the VGG model.

**6. Compile the model:**

- Use an optimizer such as Adamax with a specified learning rate (e.g., `0.0001`).

- Specify the loss function as categorical cross-entropy.

- Specify metrics such as accuracy for model evaluation.

**7. Train the model:**

- Train the model using the training data, specifying the number of epochs (e.g., `5`) and optionally a batch size.

- Validate the model using validation data.

**8. Evaluate the model:**

- Monitor the training and validation performance metrics, such as accuracy, over the epochs.

**CHAPTER 7**

**OUTPUT SCREENS**

**­­­­­­­­­­**

**7.1 Gradio output screens:**

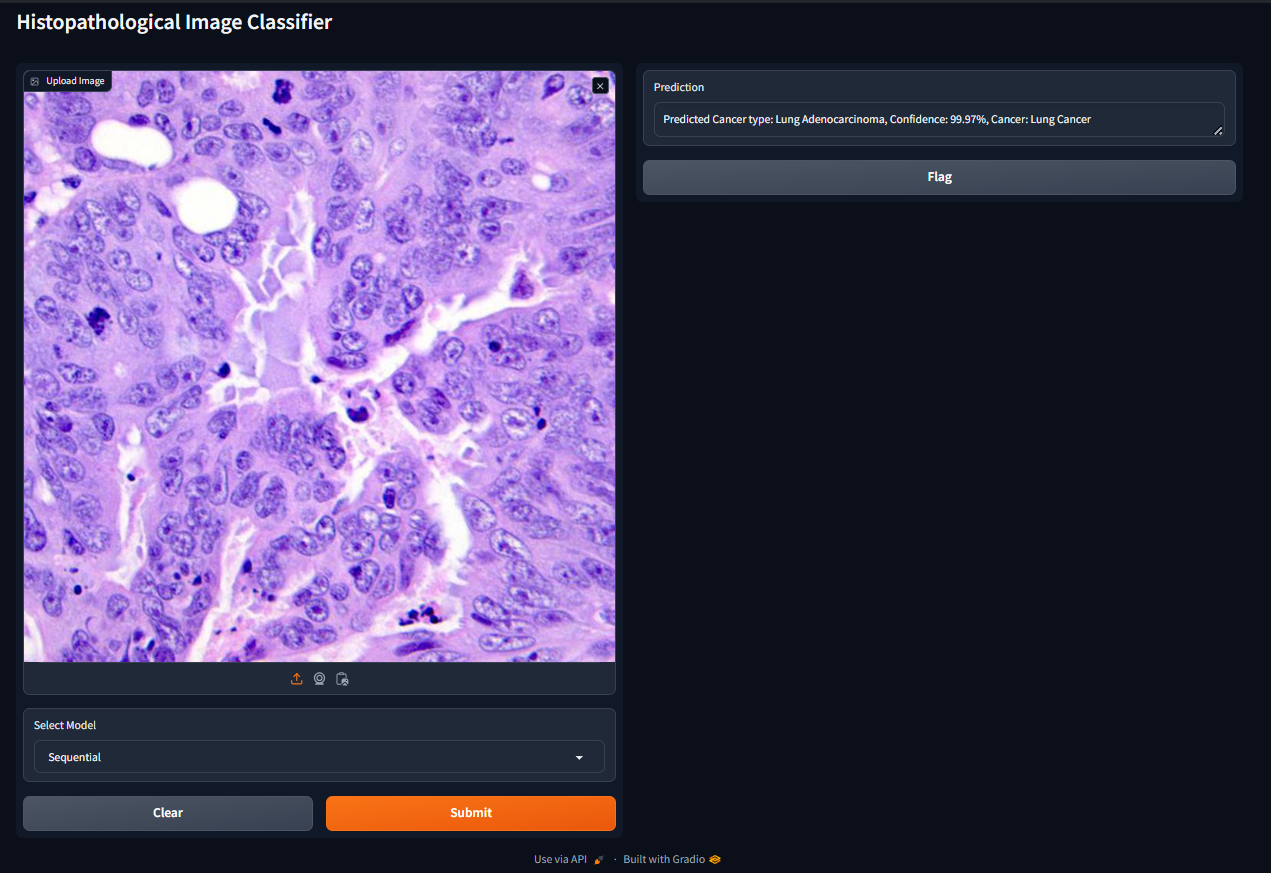
****

Fig 7.1.A: Sequential Model

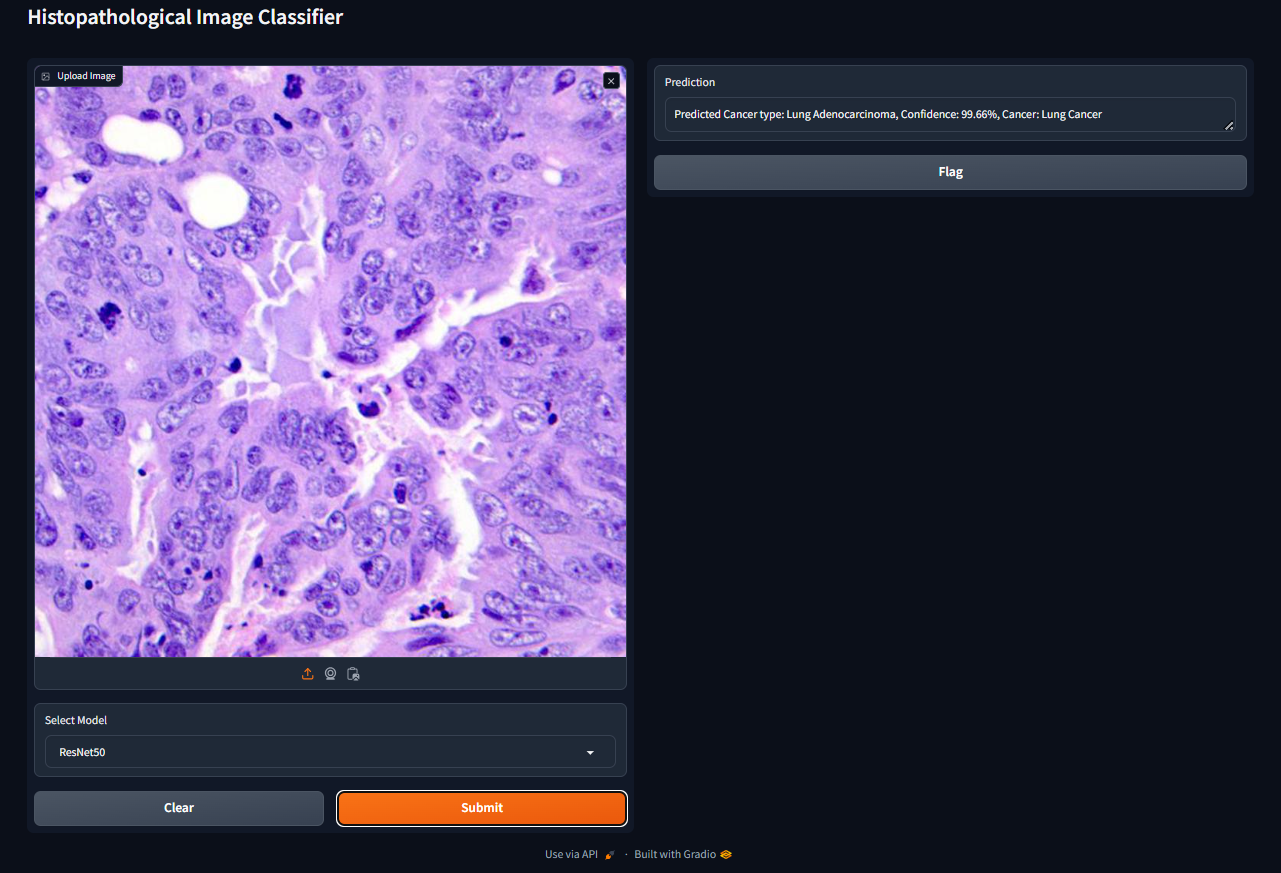
****

Fig 7.1.B: ResNet Model

**7.2 Frontend outputs:**

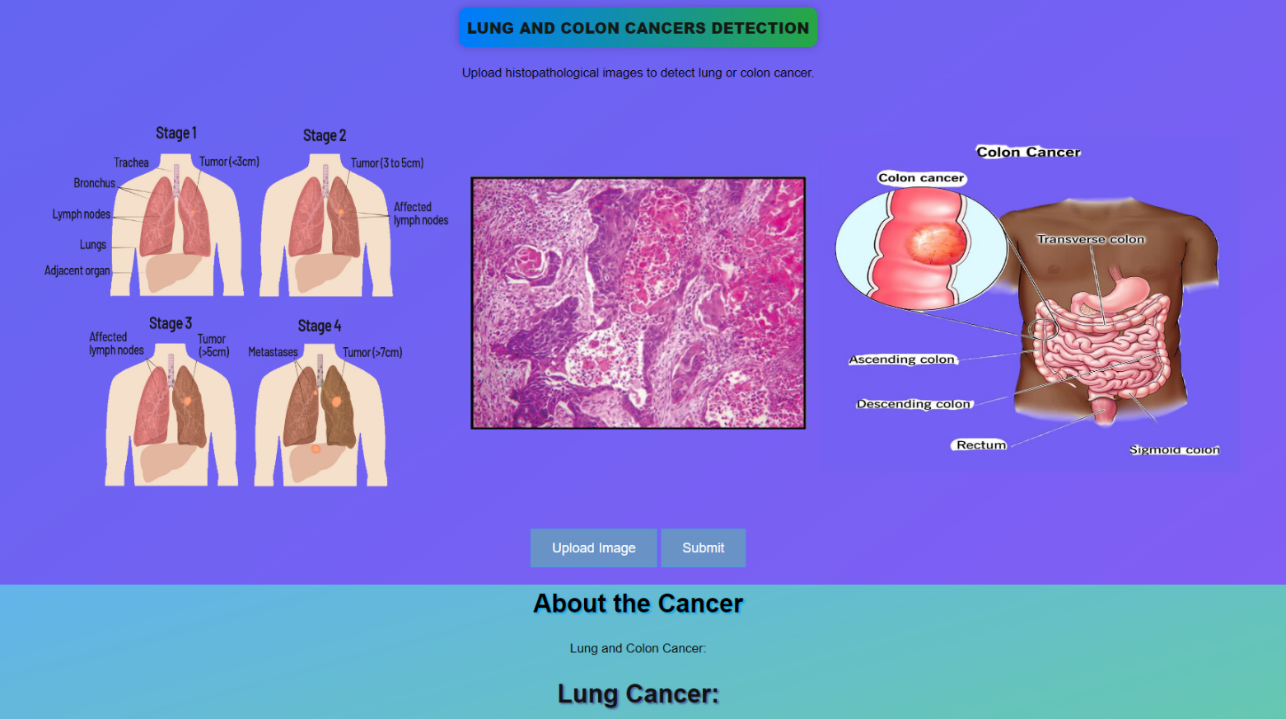
****

Fig 7.2.A: Interface

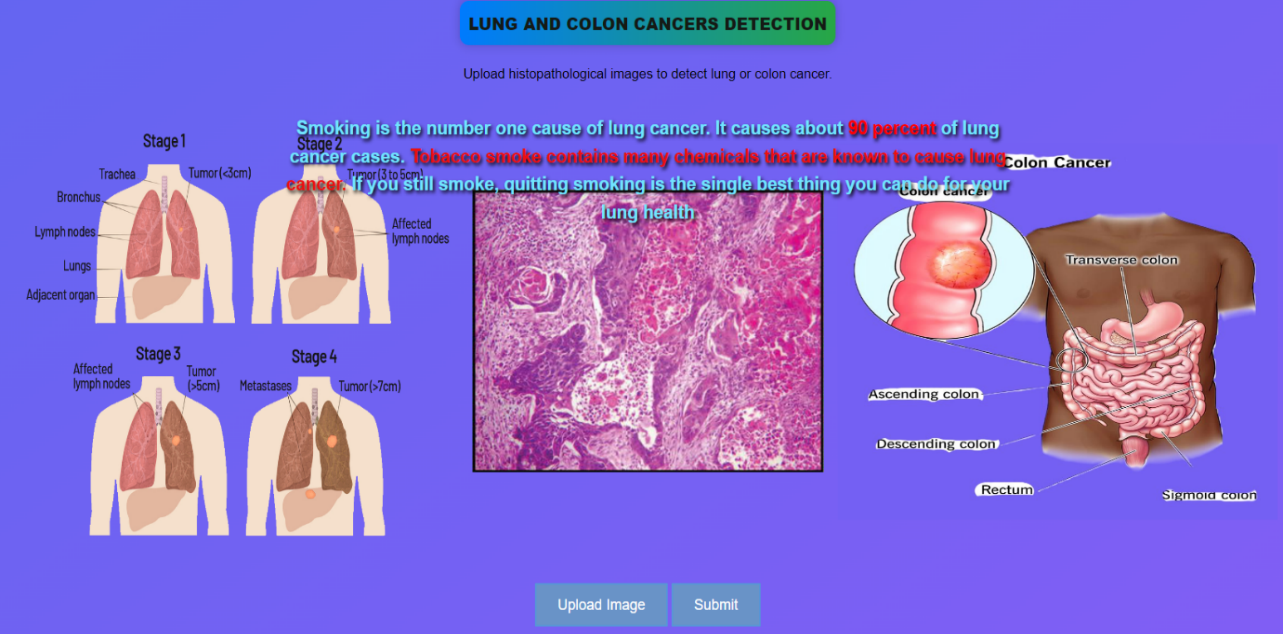
****

Fig 7.2.B: Cursor at left image

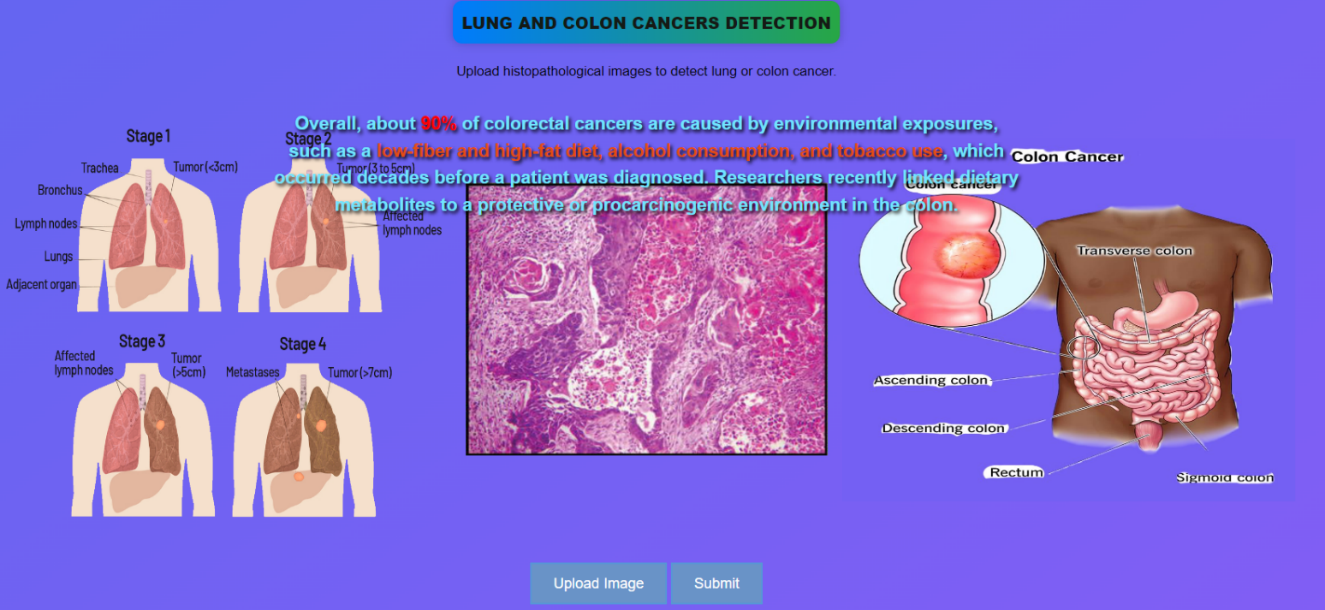
****

Fig 7.2.C: Cursor at Right image

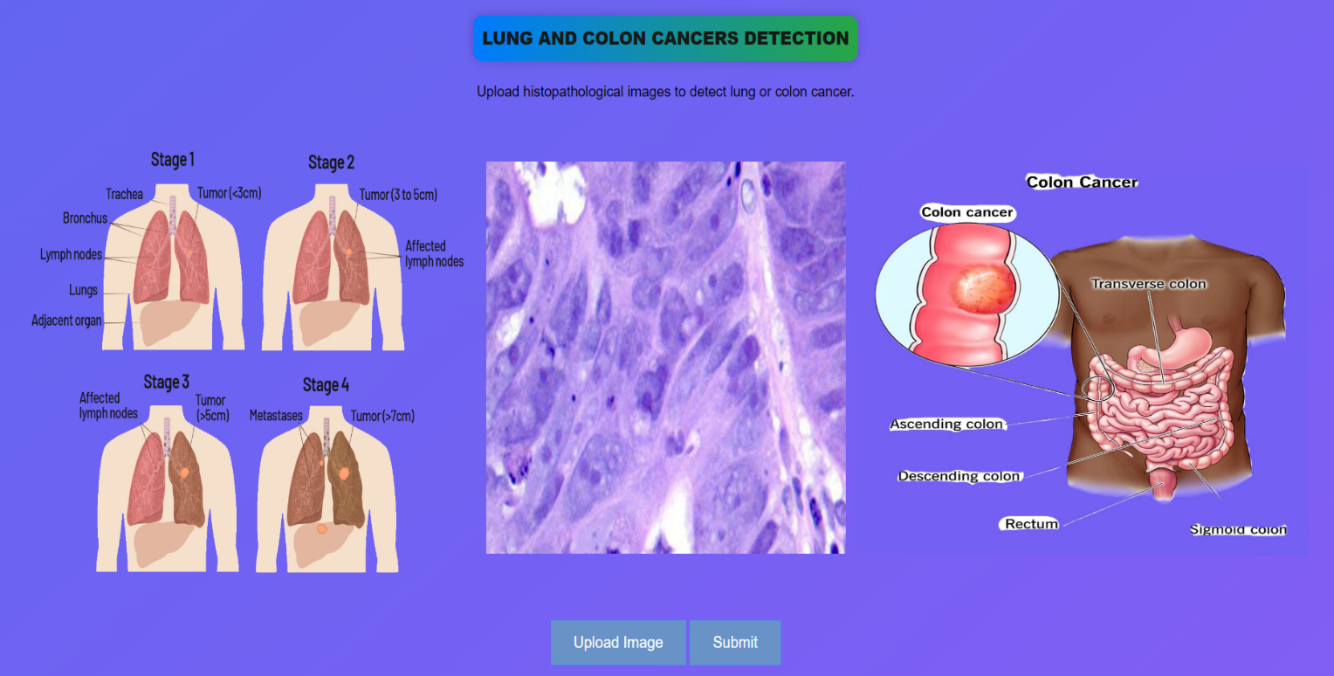
****

Fig 7.2.D: Image Uploded

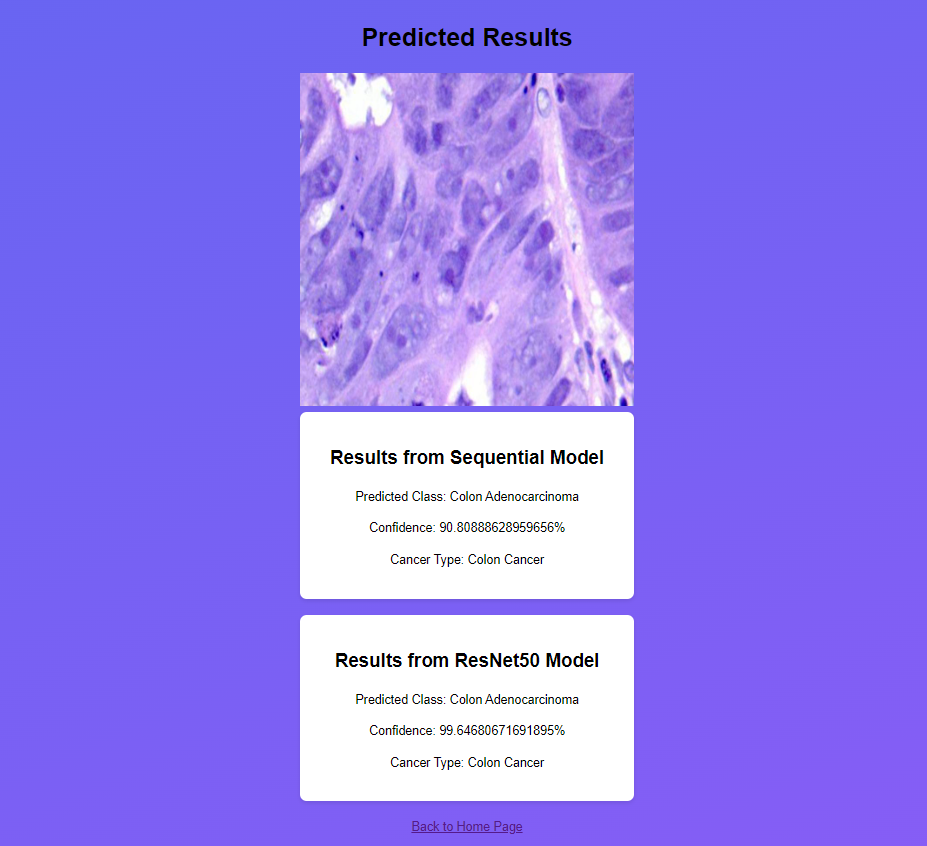
****

Fig 7.2.E: Predicted Results

**CHAPTER 8**

**CONCLUSION**

**­­­­­­­­­­­­­­­**

**CONCLUSION**

In conclusion, the use of Convolutional Neural Networks (CNNs) for Lung and Colon Cancer Detection has proven to deliver superior results in diagnosing these types of cancer. By using deep CNN models like the **Sequential model**, which **achieved 98% accuracy in training**, and the **ResNet50 model**, which **achieved 97% accuracy** ,the system enhances the accuracy and efficiency of classifying histopathological images compared to conventional machine learning models. This significant advancement in medical imaging has the potential to drastically reduce the time required for diagnosing cancer, providing more reliable performance than existing approaches and enabling doctors to treat patients more effectively and promptly. Ultimately, this innovative approach represents a major step forward in the early detection and treatment of lung and colon cancers, leading to better patient outcomes and increased survival rates.

**REFERENCES**

[1] Nishio M., Nishio M., Jimbo N., Nakane K. Homology-Based Image Processing for Automatic Classification of Histopathological Images of Lung Tissue. *Cancers.*2021;13:1192. Doi: 10.3390/cancers13061192

[2] Mangal S., Chaurasia A., Khajanchi A. Convolution neural networks for diagnosing colon and lung cancer histopathological images. *arXiv.*20202009.03878

[3] Mehmood S., Ghazal T.M., Khan M.A., Zubair M., Naseem M.T., Faiz T., Ahmad M. Malignancy detection in lung and colon histopathology images using transfer learning with class selective image processing.

[3] Tharsanee, R.M., Soundariya, R.s., Kumar, A.S., Karthiga, M., & Sountharrajan, S. (2021). Deep Convolutional neural network-based image classification for COVID-19 diagnosis. In Data Science for COVID-19 (pp. 117-145). Academic Press.

[4] Sarwinda D., Paradisa R.H., Bustamam A., Anggia P. Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. *Procedia Comput. Sci.*2021;179:423–431.

[5] Shim W.S., Yim K., Kim T.-J., Sung Y.E., Lee G., Hong J.H., Chun S.H., Kim S., An H.J., Na S.J., et al. DeepRePath: Identifying the Prognostic Features of Early-Stage Lung Adenocarcinoma Using Multi-Scale Pathology Images and Deep Convolutional Neural Network.