**DESIGNANDIMPLEMENTATION OF A DEEP LEARNING MODELFOR EARLY LUNG CANCER DETECTION USING CT IMAGING**

**ABSTRACT**

Detecting lung cancer is both challenging and crucial due to its high mortality rate. Lung cancer is one of the most prevalent forms of cancer in India, alongside prostate, mouth, and breast cancer. Contributing factors such as smoking, rising pollution levels, and exposure to carcinogenic elements significantly impact both men and women, with men being more affected. Since lung cancer poses a severe risk to both genders, early and accurate detection is essential. This project explores the use of a **Vision Transformer (ViT) algorithm** to predict lung cancer risk by analyzing medical imaging data. The Vision Transformer, known for its ability to process image patches as input and learn global patterns effectively, is well-suited for detecting nodules in CT scans and X-rays. By leveraging this state-of-the-art deep learning technique, the model can identify subtle features that traditional methods might overlook. The study also emphasizes the importance of preprocessing and augmenting medical images to enhance the algorithm's performance. With its advanced capabilities in image recognition, the Vision Transformer provides a highly accurate and scalable solution for lung cancer detection, aiming to reduce fatalities through timely and reliable diagnosis.

**LANGUAGES USED:**

**FRONT END: HTML, CSS, BOOTSTRAP**

**BACK END: PYTHON**

**FRAMEWORK: FLASK**

**MODULES DESCRIPTION**

**Dataset Collection**

This module focuses on acquiring diverse medical imaging datasets, such as CT scans and X-rays, from trusted sources, including reputable health institutions, research repositories, and public health organizations. These datasets should comprehensively include annotated images with lung cancer-related features, along with patient metadata when available, to enable precise detection and diagnosis.

**Data Preprocessing**

In this phase, medical images undergo preprocessing steps to enhance their quality and ensure suitability for Vision Transformer (ViT) algorithms. Tasks include resizing images into fixed-size patches, normalizing pixel values, and applying augmentation techniques such as rotation, flipping, and contrast adjustments. These steps improve the algorithm's ability to generalize and detect subtle patterns in medical images.

**Exploratory Data Analysis (EDA)**

EDA involves a detailed examination of the imaging dataset to uncover trends and insights. Visualizations like heatmaps and pixel intensity histograms are used to understand patterns within the images. Statistical analysis of patient metadata (e.g., age, smoking history) complements the image data, ensuring a holistic understanding of the dataset before applying Vision Transformers.

**Model Selection**

The Vision Transformer (ViT) algorithm is selected for its cutting-edge capability to process images as sequences of patches. ViT's transformer-based architecture enables it to learn both local and global features effectively, making it ideal for detecting lung cancer nodules. This approach outperforms traditional CNNs by capturing fine-grained details and context within medical images.

**Training the Model**

The dataset is split into training and testing subsets. The Vision Transformer model is trained on image patches extracted from the dataset, learning to recognize patterns indicative of lung cancer. Advanced techniques such as transfer learning, attention mechanisms, and hyperparameter tuning are applied to optimize the model's performance.

**Model Evaluation**

The trained Vision Transformer model is evaluated using metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Cross-validation is employed to ensure generalizability. Attention maps generated by the model provide insights into the areas of the image most influential in predictions, enhancing interpretability.

**Results and Predictions**

The Vision Transformer model generates predictions for new medical images, identifying lung cancer nodules with high accuracy. Results are presented in an interpretable format, including heatmaps highlighting regions of concern. This aids healthcare professionals in making informed decisions for early intervention, ultimately improving patient outcomes.

**1. INTRODUCTION**

**1.1 OVERVIEW OF THE PROJECT**

Cancer remains the second-leading cause of death worldwide, resulting from the uncontrolled growth of abnormal cells that invade and destroy healthy tissues. These cells can spread to other organs, causing diseases collectively known as cancer. Among the rising cancer types, skin, lung, prostate, and breast cancer are prominent. Despite limited knowledge on a definitive cure, advancements in research and technology are enabling earlier detection and more accurate prognoses, offering hope for better management. Machine Learning techniques, particularly deep learning methods like Vision Transformers (ViT), are playing a pivotal role in the diagnosis and prediction of cancer, especially lung cancer.

Lung cancer, with 2.21 million cases reported globally in 2020 and 1.80 million deaths, is the deadliest form of cancer. Contributing factors such as smoking, tobacco use, and exposure to carcinogens like radon gas and asbestos play a major role in its high prevalence. This project uses the Vision Transformer (ViT) algorithm for lung cancer prediction and prognosis, leveraging its ability to process medical images and extract meaningful features. ViT, a transformer-based deep learning model, is particularly suited for image-based tasks due to its capability to capture both local and global features within images.

The proposed Lung Cancer Prediction Using Vision Transformers focuses on analyzing medical images such as CT scans and X-rays, where the ViT model is trained to identify lung cancer nodules with high accuracy. By breaking images into smaller patches, the ViT algorithm learns spatial relationships and patterns in the lung tissue, making it ideal for image-based diagnostics. The model's output is then interpreted to assess the likelihood of lung cancer, aiding healthcare professionals in identifying high-risk patients for further examination and intervention.

**SYSTEM SPECIFICATION**

**HARDWARE CONFIGURATION:**

|  |  |
| --- | --- |
| Processor | : Intel icore 7 5th gen |
| Hard disk | : 500 GB |
| Ram | : 12 GB |
| Keyboard | : Logitech of 104 keys |
| Mouse | : Logitech mouse |
| Monitor | : 14 inch samtron monitor |

**SOFTWARE CONFIGURATION:**

|  |  |
| --- | --- |
| Front end  Language | : HTML, CSS, Bootstrap, JavaScript  : python |
| Operating system | : Windows 10 |
| Tools | : python IDLE |

**SOFTWARE FEATURES**

**INTRODUCTION TO VISION TRANSFORMER (ViT) ALGORITHMS**

A Vision Transformer (ViT) is a deep learning model designed to process image data using transformer-based architectures. Unlike traditional convolutional neural networks (CNNs), which rely on local receptive fields to process images, ViT operates by dividing an image into smaller patches and processing them as a sequence of tokens. The model learns the global dependencies between patches using self-attention mechanisms, which enables it to capture long-range relationships in the image. This makes ViT particularly effective in tasks that require understanding both fine-grained and broader contextual information, such as image classification, object detection, and segmentation.

**FUNCTIONALITIES OF VISION TRANSFORMER ALGORITHMS**

* **Resource Optimization:** ViT algorithms are designed to optimize computational resources by efficiently processing image data in parallel. The model leverages transformers' attention mechanisms to focus on important regions of an image, reducing computational complexity and improving performance for tasks like image recognition.
* **Interfacing with Image Data:** ViT algorithms bridge the gap between image data and machine learning models by converting images into manageable patches. These patches are treated as tokens, which are then processed by the self-attention layers to learn spatial and contextual relationships. This enables ViT models to interpret image data in a more global and context-aware manner than traditional models.
* **Task Management in Visual Data:** ViT algorithms excel in tasks involving large-scale image datasets. They are used for managing and processing tasks like classification, segmentation, and object detection, which involve organizing and analyzing visual data to extract meaningful insights. Through their attention mechanism, ViTs efficiently manage and prioritize image features based on relevance.

**USER INTERACTION THROUGH VISION TRANSFORMER ALGORITHMS**

In software systems that rely on visual data, Vision Transformer algorithms significantly enhance user interaction. ViT enables improved features such as precise image recognition, object localization, and real-time visual data processing. These capabilities lead to more responsive and intuitive user interfaces, particularly for applications involving augmented reality (AR), image search, and computer vision-based tasks. By leveraging ViT's power, software applications can provide users with a more dynamic and intelligent interaction experience, especially in environments that require complex image analysis.

**FEATURES OF VISION TRANSFORMER ALGORITHMIC IMPLEMENTATION**

* **Image Classification and Object Recognition:** ViT-based algorithms provide robust support for advanced image classification tasks, processing images in their entirety rather than just focusing on local features. This capability is particularly valuable for software systems involving visual recognition, such as facial recognition or medical image analysis.
* **Real-Time Image Processing:** ViT algorithms are designed for high-performance image processing, making them ideal for applications that require real-time analysis, such as autonomous driving, surveillance, or remote diagnostics in healthcare.
* **Advanced Web and Collaboration Tools:** In platforms relying on visual collaboration, such as video conferencing or shared digital workspaces, ViT algorithms can optimize the handling of visual data. For example, ViT can enhance video streaming quality by analyzing and compressing images efficiently, ensuring smooth real-time collaboration.

**INTRODUCTION TO BACK END**

**PYTHON:**

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**Python is Free**

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes. A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, MAC OS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

**Python is Portable**

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

**Python is Simple**

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way. A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

**FLASK:**

Flask is a lightweight and versatile web framework for building web applications in Python. Its minimalist design provides the core functionalities needed for web development without imposing unnecessary complexity, making it ideal for small to medium-sized projects. Key features of Flask include routing, templating with Jinja2, HTTP request handling, session management, and a built-in development server for easy testing during development.

One of Flask's strengths lies in its flexibility and ease of use. Developers have the freedom to customize their applications as per their specific requirements, and its simplicity makes it a great choice for both beginners and experienced developers. Flask's URL routing system allows developers to map specific URL patterns to functions, making navigation and handling different HTTP methods straightforward.

The templating engine, Jinja2, enables developers to separate logic and presentation, facilitating dynamic rendering of HTML content and displaying data from the server. Moreover, Flask supports session management, allowing developers to store user-specific data across multiple requests.

With Flask, developers can create web applications and RESTful APIs with ease, making it a popular choice for a wide range of use cases. Additionally, its ecosystem of extensions provides a wealth of additional functionalities, including database integration, authentication, and security features.

Flask's active community, extensive documentation, and continuous development ensure that it remains up-to-date and well-supported. Overall, Flask's simplicity, flexibility, and ease of learning have made it a go-to web framework for many Python developers seeking to build efficient and reliable web applications.

**HTML:**

HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation ([CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)) or functionality/behavior ([JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript)). "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

**2. SYSTEM STUDY**

**2. PROBLEM STATEMENT**

Detecting lung cancer is a significant challenge, particularly in regions like India, where it ranks as the most prevalent cancer alongside prostate, mouth, and breast cancers. Factors such as smoking, pollution, and carcinogenic exposure elevate the risk, with men being more affected than women. Early detection is vital, as lung cancer poses a high mortality risk for both genders. This project leverages the Vision Transformer (ViT) algorithm to predict lung cancer risk by analyzing medical imaging data, such as CT scans or X-rays. ViT, known for its ability to capture both local and global relationships in visual data, is well-suited for image-based diagnosis. Unlike traditional methods, ViT divides images into patches, processes them through self-attention mechanisms, and learns intricate patterns related to lung cancer indicators. Preprocessing steps ensure the data is cleaned and appropriately structured for image analysis, optimizing the model’s accuracy. The project compares ViT-based methods with conventional techniques to evaluate the effectiveness of vision transformers in detecting lung cancer nodules. By focusing on image recognition, the model presents an advanced approach to early diagnosis. This method aims to assist healthcare professionals with reliable, image-driven predictions, providing a step forward in improving lung cancer detection and patient outcomes.

**2.2 EXISTING SYSTEM**

In the traditional system of lung cancer prediction, patients face delays and accessibility issues due to the need for in-person visits and manual interpretations of CT scans and X-rays. This reliance on human expertise can introduce inconsistencies and inaccuracies in diagnosis. In contrast, the Vision Transformer (ViT) algorithm offers a more reliable and consistent method for analyzing medical images. By using self-attention mechanisms, ViT detects patterns in lung scans with greater precision, reducing human error. This approach enhances early detection, providing a scalable and efficient solution for lung cancer diagnosis.

**2.2.1 DISADVANTAGES OF EXISTING SYSTEM**

* **Time-Consuming Process:** Scheduling appointments and undergoing diagnostic tests such as CT scans and X-rays require significant time investment, leading to delays in diagnosis and treatment.
* **High Costs:** The traditional method of diagnostic testing, along with doctor consultations, can be financially burdensome for patients, especially in regions with limited healthcare resources.
* **Patient Anxiety:** The waiting period for test results and final diagnoses can cause considerable stress and anxiety for patients, delaying potential treatment and increasing emotional distress.
* **Limited Continuous Monitoring:** Existing systems lack real-time, continuous monitoring of patient health, relying on episodic testing rather than ongoing assessments.

**2.3 PROPOSED SYSTEM**

The proposed system seeks to transform lung cancer prediction by utilizing a Vision Transformer (ViT) algorithm, which eliminates the need for patients to visit hospitals or consult doctors in person. By leveraging the power of ViT's self-attention mechanism, this system allows for highly accurate and automated analysis of lung scans from anywhere, anytime. The use of remote monitoring technology combined with ViT enhances accessibility to lung cancer prediction, ensuring patients can manage their health effectively without compromising on diagnostic precision or reliability. This approach not only streamlines the process but also empowers patients to take control of their lung health more efficiently.

**2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM**

* **High Accuracy:** ViT offers improved accuracy by analyzing lung scans with a high degree of consistency, minimizing human error.
* **Cost Saving:** By reducing the need for in-person visits and manual interpretations, the system lowers healthcare costs.
* **Continuous Monitoring:** Remote monitoring via ViT enables early detection of lung health changes, facilitating timely interventions.
* **Remote Accessibility:** Patients can access the system for lung cancer prediction from anywhere, enhancing convenience and flexibility.

**SYSTEM DESIGN**

**INPUT DESIGN**

Input design is the process of transforming user-oriented data into a computer-compatible format. The objective of input design is to simplify data entry, ensuring it is logical, efficient, and error-free. The quality of the input directly impacts the quality of the system output. In the context of using the Vision Transformer (ViT) algorithm, the input design process involves creating a seamless interface for users to upload medical images (e.g., lung scans) for analysis. ViT automatically processes these images, detecting lung cancer with high accuracy. The system could include options for users to select from pre-defined input categories, reducing manual entry errors. This design optimizes the flow of data and enhances the reliability of the output, as the ViT algorithm can better analyze consistent, structured inputs.

**OUTPUT DESIGN**

Output design is critical in any system, as it directly impacts the user's decision-making process. For the Vision Transformer (ViT) algorithm, the output focuses on providing clear and accurate analysis of medical images, such as lung scans. The system generates visual reports that highlight areas of concern, along with a risk assessment. These outputs are designed to be intuitive and actionable, aiding healthcare professionals in diagnosing lung cancer. Effective output design ensures that the results are reliable and easy to interpret, enhancing the decision-making process.

**PROJECT ARCHITECTURE**

Frontend

Python

Flask

Server

Machine Learning

Model

Architecture of the Application

**FLOWCHART**

Raw Medical Data

Data Pre-Processing

Machine learning Model Training

Features Extraction for disease prediction

Classification

Having disease or not

Model Testing

GUI

Development

General Flowchart of Methodology

**SYSTEM TESTING AND IMPLEMENTATION**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been integrated and perform allocated functions. During testing I tried to make sure that the product does exactly what is supposed to do. Testing is the final verification and validation activity within the organization itself. In the testing stage, I try to achieve the following goals; to affirm the quality of the product, to find and eliminate any residual errors from previous stages, to validate the software as a solution to the original problem, to demonstrate the presence of all specified functionality in the product, to estimate the operational reliability of the system. During testing the major activities are concentrated on the examination and modification of the source code.

System testing for a Vision Transformer (ViT) based lung cancer prediction system involves several stages to ensure the model's effectiveness and accuracy. Each stage of testing focuses on verifying that the ViT model can correctly process and interpret medical images and provide accurate predictions.

**UNIT TESTING**

The system is divided into individual components, such as image preprocessing, model inference, and result generation. After testing each part, the ViT model is assessed to ensure that each component works independently, with particular focus on ensuring the correct processing of lung scans and image classification tasks.

**INTEGRATION TESTING**

After individual components are unit tested, they are integrated into the overall system. In this phase, we perform a top-down integration approach, where each module (e.g., image input, ViT model inference, and output display) is integrated progressively, ensuring the combined functionality operates seamlessly.

**VALIDATION TESTING**

Validation testing ensures that the system can accurately handle different image inputs, particularly focusing on the ViT model's ability to detect patterns in lung scans. It checks if the model can classify images correctly, ensuring that it doesn't misinterpret or fail to identify relevant features for lung cancer prediction.

**SAMPLE TEST CASE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TESTING** | **SCENARIO** | **TEST STEP** | **EXCEPTED**  **RESULT** | **ACTUAL OUTCOME** | **RESULT** |
| Unit testing | Verify that the Vision Transformer (ViT) model correctly processes a lung scan image. | Input a lung scan image into the ViT model, which should classify the image as either "Cancer Detected" or "No Cancer Detected." | The ViT model should correctly classify the image and move to the next step of processing. | The ViT model accurately classifies the lung scan image as either "Cancer Detected" or "No Cancer Detected." | success |
| Integration testing | Verify that the ViT model integrates with the image preprocessing and result display modules. | Input a lung scan image, preprocess it (resize, normalize), pass it through the ViT model, and display the result on the interface. | The image should be preprocessed correctly, the ViT model should generate a prediction, and the result should be displayed on the user interface. | The system preprocesses the image, the ViT model generates the prediction, and the result is correctly displayed on the UI. | success |
| Acceptance testing | Verify that the ViT-based lung cancer detection application satisfies the user requirements. | Execute the application with sample lung scan images provided by the end user (e.g., healthcare professionals) and ensure that the system works as expected. | The application should process the lung scan images, predict the results accurately, and satisfy the user's expectations for functionality. | The application processes the images correctly and meets the expected functionality. The user is satisfied with the result. | success |

**SYSTEM IMPLEMENTATION**

System Implementation ensures that the ViT-based system is deployed and made available to a prepared set of users while transitioning to an ongoing support and maintenance phase. The transition involves confirming that all data required for the ViT model's operations is available, accurate, and functioning properly. During this phase, the system's ownership moves from the project team to the organization’s operational support team. The stages involved in ViT algorithm-based system implementation include:

**1. Planning**

Planning involves deciding the method and timeline for deploying the ViT algorithm-based system. Key tasks include ensuring the ViT model's seamless integration with existing infrastructure and aligning resources. The team coordinates with different departments and stakeholders, such as technical support and data scientists, to ensure a smooth deployment. This phase also includes:

* Identifying system environment implications for the ViT model.
* Allocating tasks related to the ViT integration and training.
* Consulting with teams to ensure resources and backup facilities are available.

**2. Training**

Training focuses on educating consumers and system users about how to utilize the ViT model effectively. This is crucial because the ViT model, being an advanced deep learning algorithm, requires thorough understanding and expertise for optimal usage. The training ensures:

* The end-users are well-versed in interacting with the ViT system.
* Training sessions are professionally conducted to build trust and confidence in the system.

**3. System Testing**

System testing involves verifying the ViT model’s integration with other system components and validating its performance in real-world conditions. This includes:

* Testing the ViT model with real-world data to ensure it can handle various scenarios.
* Confirming that the model’s predictions are accurate and aligned with expectations.

**4. Changeover Planning**

Changeover planning ensures that the transition from the old system to the ViT-based system happens smoothly. This includes:

* Ensuring there are no disruptions in business operations during the changeover.
* Managing the temporary unavailability of systems and setting up manual log systems if necessary.
* Communicating deployment activities clearly to avoid confusion during the changeover.

**Key Impacts During Implementation:**

* Users may face temporary unavailability of the ViT-powered system and must keep manual records during the transition.
* Technical support teams may be overwhelmed with support requests due to system adjustments and the learning curve associated with the new ViT algorithm.
* The communication of deployment steps is critical to ensure smooth operations.

The successful deployment of the ViT system requires rigorous planning, leadership, and coordinated communication across all involved parties. During the final phase, the ownership of the ViT system is transferred to the operational team, and the project manager ensures a smooth transition and adequate ongoing support.

In essence, transitioning to a ViT-based system requires effective system deployment planning, extensive user training, thorough system testing, and careful management of the changeover process to ensure that the organization can effectively adopt and maintain the ViT model for optimal performance.

**SYSTEM MAINTENANACE**

The maintenance phase of the software life cycle ensures that a software product continues to function properly after deployment. It involves activities like system enhancements, adapting the software to new environments, and correcting errors. Regular backups of the system, including executable files and reports, are essential for data safety and recovery. Enhancements can involve adding new features, improving user interfaces, or upgrading performance.

Adapting the software to new environments might include migrating it to different hardware or software platforms. Error correction is an ongoing task where bugs are fixed as they are discovered, either immediately or during scheduled updates. Maintenance also requires revisiting earlier stages of development, such as design or analysis, to implement changes.

Effective software maintenance ensures that the product remains stable, secure, and up-to-date, addressing evolving user needs and technology changes. It requires careful planning and consistent attention to ensure that the system continues to meet its goals.

**CONCLUSION**

Through the project, it was identified that **machine learning algorithms** play a crucial role in achieving high accuracy for lung cancer prediction, depending on various factors such as the **dataset** and **feature selection**. The study explores different **discrete machine learning methods** applied by various researchers, highlighting both their advantages and limitations based on the chosen data and features. The project also reveals that the **accuracy and performance** of predictions can be significantly improved by employing **hybrid methods** and combinations of machine learning algorithms. These hybrid approaches provide more robust results, demonstrating that careful selection of algorithms, datasets, and features is key to enhancing the prediction capabilities for lung cancer detection.

**FUTURE ENHANCEMENT**

The future outlook for this system involves expanding its scope by incorporating various data types, such as images and more advanced deep learning algorithms, to further enhance prediction accuracy. This system could offer a more refined understanding of cancer risk factors, benefiting individuals by raising awareness about potential causes of cancer. Moreover, the system could be expanded to diagnose other types of cancer, thereby broadening its impact on early detection and prevention.

Another future direction includes the integration of interactive tools and educational resources powered by machine learning to inform patients about health risks, lifestyle changes, and the importance of treatment adherence. To ensure the privacy and security of sensitive patient information, robust measures would be implemented, in line with regulatory standards such as HIPAA, GDPR, and CCPA, to protect confidentiality and prevent unauthorized access.

By embracing these enhancements, the project would evolve into a more personalized and accurate healthcare solution, improving lung cancer diagnosis, reducing the global burden of lung diseases, and ultimately enhancing patient outcomes.

**DATA FLOW DIAGRAM**

**Level 0**

Lung cancer classification

Person

**LEVEL 1**

Feature Extraction

Test Image

Predict

Lung Cancer

Training Dataset

**Level 2**

Testing dataset

Training dataset

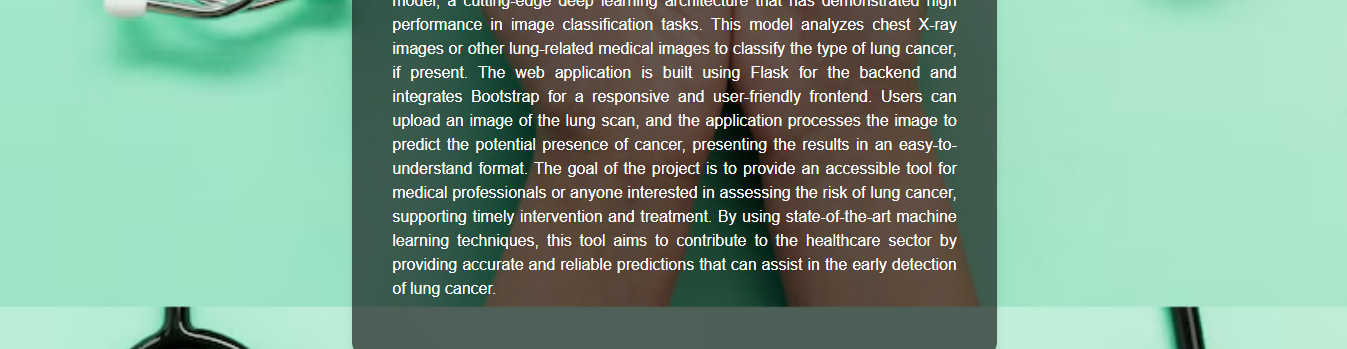
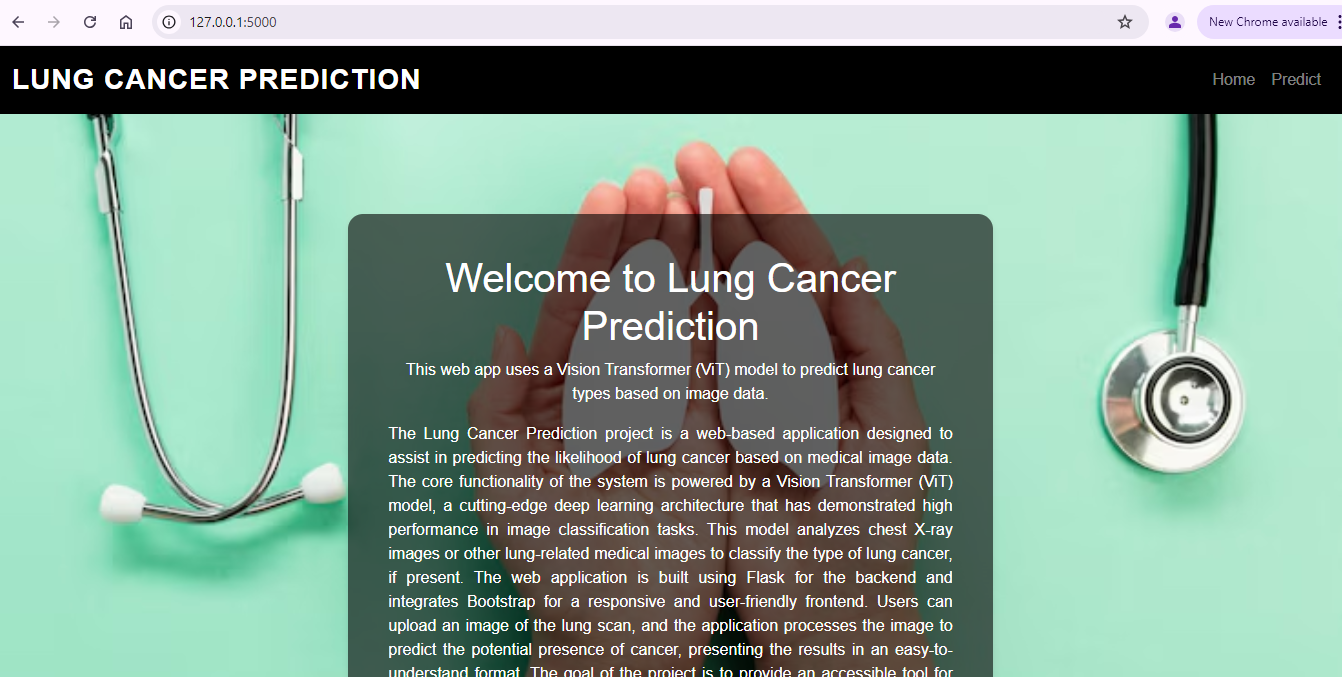
**Level 3**

Lung cancer

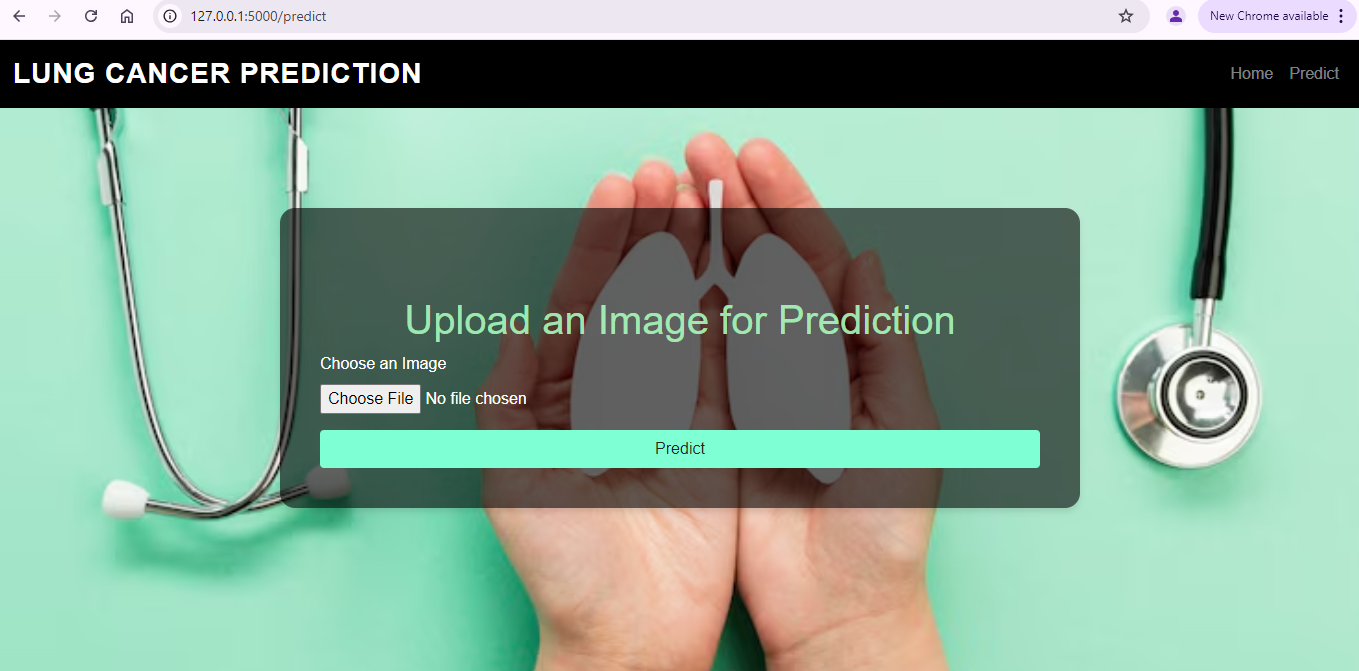
Features

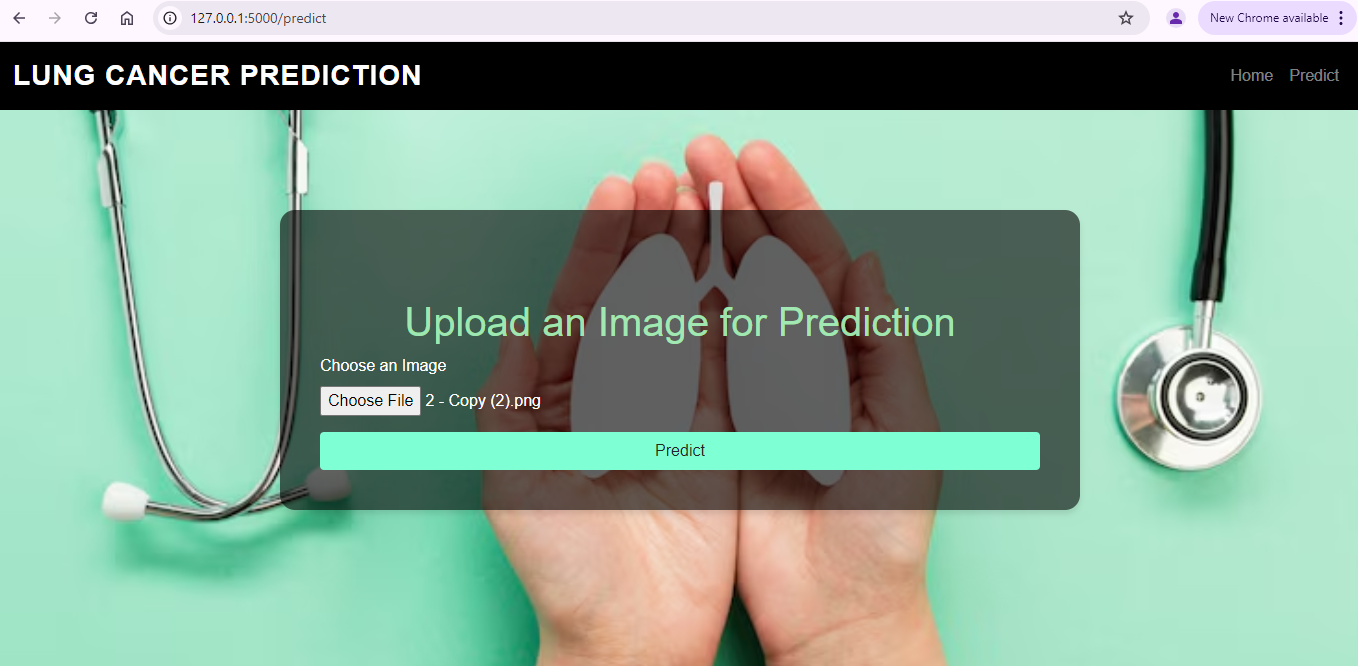
Classified Lung Cancer

**SCREENSHOT**

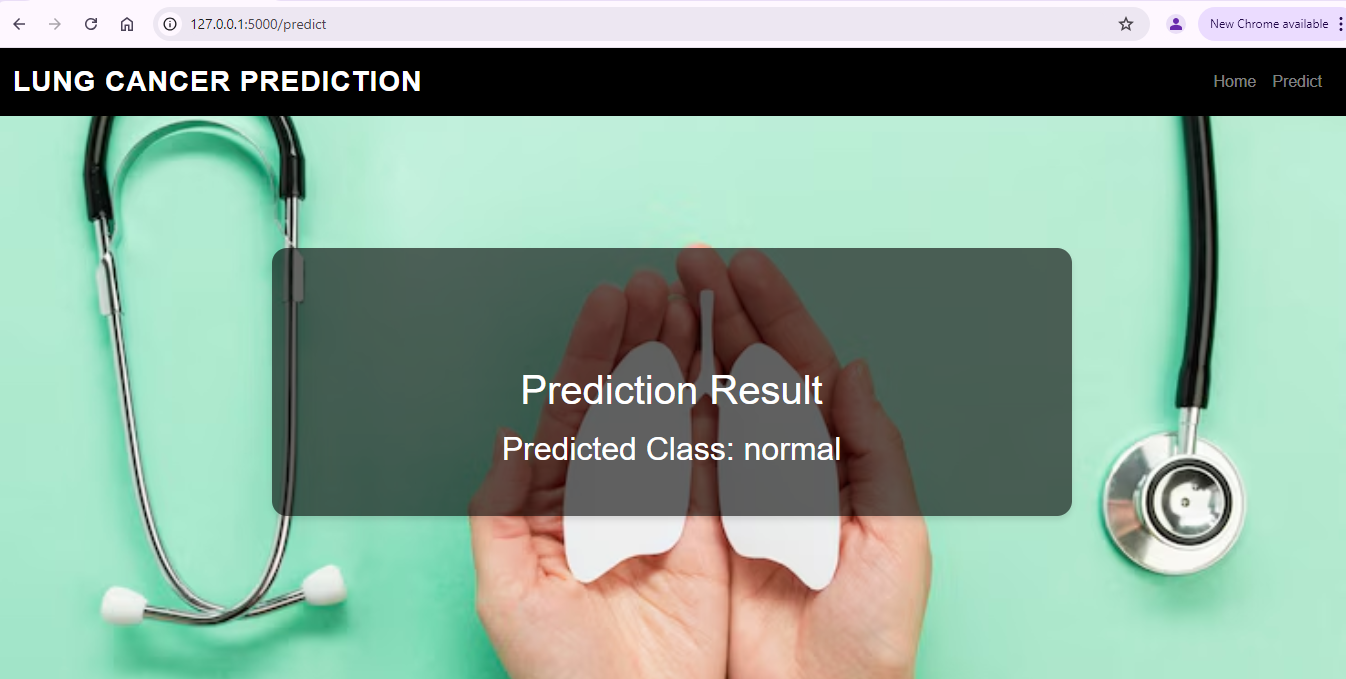


**PREDICT PAGE**

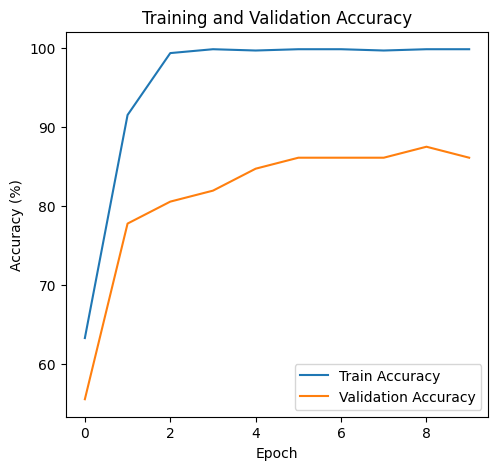


****

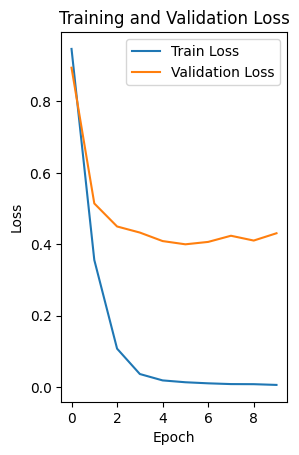
**RESULT PAGE**



**Training and Validation Accuracy**



**Training and validation loss**



**SAMPLE CODING**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Home - Lung Cancer Prediction</title>

<link

rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-image: url('https://img.freepik.com/free-photo/hands-with-lungs-shape\_23-2148533078.jpg?t=st=1731912641~exp=1731916241~hmac=b9e8e87c741c95f45c07c0e83dd6b48738f461889a2414cf82cfcadf2f7b447c&w=740');

background-size: cover;

background-position: center;

height: 100vh; /\* Ensure the background covers the entire screen height \*/

color: white; /\* Text color to stand out against the background \*/

}

.navbar {

width: 100%;

background-color: black;

}

.navbar-brand {

font-family: 'Helvetica', sans-serif; /\* Elegant font family \*/

font-weight: 700; /\* Bold text \*/

font-size: 1.75rem; /\* Larger text size \*/

letter-spacing: 1px; /\* Slight spacing between letters \*/

color: #f0f0f0; /\* Light color for better contrast \*/

text-transform: uppercase; /\* Uppercase for a more formal look \*/

}

.container {

max-width: 50%;

padding: 0 15px;

margin-top: 100px;

}

.content-container {

background-color: rgba(0, 0, 0, 0.6); /\* Semi-transparent black background \*/

padding: 40px;

border-radius: 15px;

margin-top: 100px; /\* Adjust space from the top \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow for a lifted effect \*/

}

.text-center {

text-align: center;

}

.mt-5 {

margin-top: 5rem;

}

</style>

</head>

<body>

<!-- Full-width Navbar -->

<nav class="navbar navbar-expand-lg navbar-dark w-100">

<a class="navbar-brand" href="#">Lung Cancer Prediction</a>

<div class="collapse navbar-collapse">

<ul class="navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" href="/">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/predict">Predict</a>

</li>

</ul>

</div>

</nav>

<!-- Content Container -->

<div class="container">

<div class="content-container">

<h1 class="text-center">Welcome to Lung Cancer Prediction</h1>

<p class="text-center">This web app uses a Vision Transformer (ViT) model to predict lung cancer types based on image data.</p>

<p class="text-justify">

The Lung Cancer Prediction project is a web-based application designed to assist in predicting the likelihood of lung cancer based on medical image data. The core functionality of the system is powered by a Vision Transformer (ViT) model, a cutting-edge deep learning architecture that has demonstrated high performance in image classification tasks. This model analyzes chest X-ray images or other lung-related medical images to classify the type of lung cancer, if present. The web application is built using Flask for the backend and integrates Bootstrap for a responsive and user-friendly frontend. Users can upload an image of the lung scan, and the application processes the image to predict the potential presence of cancer, presenting the results in an easy-to-understand format. The goal of the project is to provide an accessible tool for medical professionals or anyone interested in assessing the risk of lung cancer, supporting timely intervention and treatment. By using state-of-the-art machine learning techniques, this tool aims to contribute to the healthcare sector by providing accurate and reliable predictions that can assist in the early detection of lung cancer.

</p></div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Predict - Lung Cancer Prediction</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-image: url('https://img.freepik.com/free-photo/hands-with-lungs-shape\_23-2148533078.jpg?t=st=1731912641~exp=1731916241~hmac=b9e8e87c741c95f45c07c0e83dd6b48738f461889a2414cf82cfcadf2f7b447c&w=740');

background-size: cover;

background-position: center;

height: 100vh; /\* Ensure the background covers the entire screen height \*/

color: white; /\* Text color to stand out against the background \*/

}

.navbar {

width: 100%;

background-color: black;

}

.container {

max-width: 800px; /\* Fixed width for the form \*/

padding: 0 15px;

margin-top: 5rem; /\* To give some space from top \*/

background-color: rgba(0, 0, 0, 0.6); /\* Semi-transparent black background \*/

padding: 40px;

border-radius: 15px;

margin-top: 100px; /\* Adjust space from the top \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow for a lifted effect \*/

}

.form-group {

width: 100%;

}

.btn-block {

width: 100%;

}

.text-center {

text-align: center;

}

.mt-5 {

margin-top: 5rem;

}

.navbar-brand {

font-family: 'Helvetica', sans-serif; /\* Elegant font family \*/

font-weight: 700; /\* Bold text \*/

font-size: 1.75rem; /\* Larger text size \*/

letter-spacing: 1px; /\* Slight spacing between letters \*/

color: #f0f0f0; /\* Light color for better contrast \*/

text-transform: uppercase; /\* Uppercase for a more formal look \*/

}

</style>

</head>

<body>

<!-- Navbar (remains outside the container as is) -->

<nav class="navbar navbar-expand-lg navbar-dark w-100">

<a class="navbar-brand" href="#">Lung Cancer Prediction</a>

<div class="collapse navbar-collapse">

<ul class="navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" href="/">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/predict">Predict</a>

</li>

</ul>

</div>

</nav>

<!-- Form section inside the container -->

<div class="container">

<h1 class="text-center mt-5" style="color:rgba(167, 245, 187, 0.911)">Upload an Image for Prediction</h1>

<form action="/predict" method="POST" enctype="multipart/form-data">

<div class="form-group">

<label for="image">Choose an Image</label>

<input type="file" class="form-control-file" id="image" name="image" required>

</div>

<button type="submit" class="btn btn-block" style="background-color: aquamarine;">Predict</button>

</form>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Prediction Result - Lung Cancer Prediction</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-image: url('https://img.freepik.com/free-photo/hands-with-lungs-shape\_23-2148533078.jpg?t=st=1731912641~exp=1731916241~hmac=b9e8e87c741c95f45c07c0e83dd6b48738f461889a2414cf82cfcadf2f7b447c&w=740');

background-size: cover;

background-position: center;

height: 100vh; /\* Ensure the background covers the entire screen height \*/

color: white; /\* Text color to stand out against the background \*/

}

.navbar {

width: 100%;

background-color: black;

}

.container {

max-width: 800px; /\* Fixed width for the form \*/

padding: 0 15px;

margin-top: 5rem; /\* To give some space from top \*/

background-color: rgba(0, 0, 0, 0.6); /\* Semi-transparent black background \*/

padding: 40px;

border-radius: 15px;

margin-top: 100px; /\* Adjust space from the top \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow for a lifted effect \*/

}

.text-center {

text-align: center;

}

.mt-5 {

margin-top: 5rem;

}

.img-fluid {

max-width: 100%;

}

/\* Ensure full-width navbar \*/

.navbar {

width: 100%;

position: fixed; /\* Keep the navbar at the top \*/

top: 0;

left: 0;

z-index: 10; /\* Make sure it is above other content \*/

}

.navbar-nav {

margin-left: auto;

}

/\* Padding to avoid content being hidden under the fixed navbar \*/

.container {

padding-top: 70px; /\* Adjust depending on navbar height \*/

margin-top: 200px;

}

.navbar-brand {

font-family: 'Helvetica', sans-serif; /\* Elegant font family \*/

font-weight: 700; /\* Bold text \*/

font-size: 1.75rem; /\* Larger text size \*/

letter-spacing: 1px; /\* Slight spacing between letters \*/

color: #f0f0f0; /\* Light color for better contrast \*/

text-transform: uppercase; /\* Uppercase for a more formal look \*/

}

</style>

</head>

<body>

<div class="container">

<!-- Full-width Fixed Navbar -->

<nav class="navbar navbar-expand-lg navbar-dark w-100">

<a class="navbar-brand" href="#">Lung Cancer Prediction</a>

<div class="collapse navbar-collapse">

<ul class="navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" href="/">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/predict">Predict</a>

</li>

</ul>

</div>

</nav>

<!-- Content Section -->

<h1 class="text-center mt-5">Prediction Result</h1>

<div class="text-center">

<h2 class="mt-3">Predicted Class: {{ predicted\_class }}</h2>

</div>

</div>

</body>

</html>

import os

from flask import Flask, render\_template, request, redirect, url\_for

import torch

from transformers import ViTForImageClassification, ViTFeatureExtractor

from torchvision import transforms

from PIL import Image

app = Flask(\_\_name\_\_)

# Device configuration

device = torch.device("cuda" if torch.cuda.is\_available() else "cpu")

# Load the pretrained ViT model

model = ViTForImageClassification.from\_pretrained(

'google/vit-base-patch16-224',

num\_labels=4, # Set this to 4 because the checkpoint is for a model with 4 classes

ignore\_mismatched\_sizes=True # Ignore size mismatch for classifier layer

)

model = model.to(device)

# Load the trained model weights (ensure they match the classifier size)

checkpoint = torch.load('lung\_cancer\_vit\_model.pth', map\_location=device)

model.load\_state\_dict(checkpoint, strict=False) # strict=False will ignore mismatch in classifier

# Set the model to evaluation mode

model.eval()

# Load feature extractor for preprocessing

feature\_extractor = ViTFeatureExtractor.from\_pretrained('google/vit-base-patch16-224')

# Define the transform for input images

transform = transforms.Compose([

transforms.Resize((224, 224)),

transforms.ToTensor(),

transforms.Normalize(mean=feature\_extractor.image\_mean, std=feature\_extractor.image\_std),

])

# Prediction function

def predict\_image(image\_path):

image = Image.open(image\_path).convert('RGB')

input\_tensor = transform(image).unsqueeze(0).to(device)

with torch.no\_grad():

outputs = model(input\_tensor).logits

\_, predicted\_class = torch.max(outputs, dim=1)

return predicted\_class.item()

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])

def predict():

if request.method == 'POST':

# Get the uploaded image

file = request.files['image']

if file:

# Save the image to a temporary location

image\_path = os.path.join('static', 'uploads', file.filename)

file.save(image\_path)

# Predict the class of the image

predicted\_class = predict\_image(image\_path)

class\_names = ['adenocarcinoma', 'large.cell.carcinoma', 'normal', 'squamous.cell.carcinoma']

predicted\_label = class\_names[predicted\_class]

# Render the result page with the prediction

return render\_template('result.html', predicted\_class=predicted\_label, image\_path=image\_path)

return render\_template('predict.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True) import os

from flask import Flask, render\_template, request, redirect, url\_for

import torch

from transformers import ViTForImageClassification, ViTFeatureExtractor

from torchvision import transforms

from PIL import Image

app = Flask(\_\_name\_\_)

# Device configuration

device = torch.device("cuda" if torch.cuda.is\_available() else "cpu")

# Load the pretrained ViT model

model = ViTForImageClassification.from\_pretrained(

'google/vit-base-patch16-224',

num\_labels=4, # Set this to 4 because the checkpoint is for a model with 4 classes

ignore\_mismatched\_sizes=True # Ignore size mismatch for classifier layer

)

model = model.to(device)

# Load the trained model weights (ensure they match the classifier size)

checkpoint = torch.load('lung\_cancer\_vit\_model.pth', map\_location=device)

model.load\_state\_dict(checkpoint, strict=False) # strict=False will ignore mismatch in classifier

# Set the model to evaluation mode

model.eval()

# Load feature extractor for preprocessing

feature\_extractor = ViTFeatureExtractor.from\_pretrained('google/vit-base-patch16-224')

# Define the transform for input images

transform = transforms.Compose([

transforms.Resize((224, 224)),

transforms.ToTensor(),

transforms.Normalize(mean=feature\_extractor.image\_mean, std=feature\_extractor.image\_std),

])

# Prediction function

def predict\_image(image\_path):

image = Image.open(image\_path).convert('RGB')

input\_tensor = transform(image).unsqueeze(0).to(device)

with torch.no\_grad():

outputs = model(input\_tensor).logits

\_, predicted\_class = torch.max(outputs, dim=1)

return predicted\_class.item()

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Predict - Lung Cancer Prediction</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-image: url('https://img.freepik.com/free-photo/hands-with-lungs-shape\_23-2148533078.jpg?t=st=1731912641~exp=1731916241~hmac=b9e8e87c741c95f45c07c0e83dd6b48738f461889a2414cf82cfcadf2f7b447c&w=740');

background-size: cover;

background-position: center;

height: 100vh; /\* Ensure the background covers the entire screen height \*/

color: white; /\* Text color to stand out against the background \*/

}

.navbar {

width: 100%;

background-color: black;

}

.container {

max-width: 800px; /\* Fixed width for the form \*/

padding: 0 15px;

margin-top: 5rem; /\* To give some space from top \*/

background-color: rgba(0, 0, 0, 0.6); /\* Semi-transparent black background \*/

padding: 40px;

border-radius: 15px;

margin-top: 100px; /\* Adjust space from the top \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow for a lifted effect \*/

}

.form-group {

width: 100%;

}

.btn-block {

width: 100%;

}

.text-center {

text-align: center;

}

.mt-5 {

margin-top: 5rem;

}

.navbar-brand {

font-family: 'Helvetica', sans-serif; /\* Elegant font family \*/

font-weight: 700; /\* Bold text \*/

font-size: 1.75rem; /\* Larger text size \*/

letter-spacing: 1px; /\* Slight spacing between letters \*/

color: #f0f0f0; /\* Light color for better contrast \*/

text-transform: uppercase; /\* Uppercase for a more formal look \*/

}

</style>

</head>

<body>

<!-- Navbar (remains outside the container as is) -->

<nav class="navbar navbar-expand-lg navbar-dark w-100">

<a class="navbar-brand" href="#">Lung Cancer Prediction</a>

<div class="collapse navbar-collapse">

<ul class="navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" href="/">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/predict">Predict</a>

</li>

</ul>

</div>

</nav>

<!-- Form section inside the container -->

<div class="container">

<h1 class="text-center mt-5" style="color:rgba(167, 245, 187, 0.911)">Upload an Image for Prediction</h1>

<form action="/predict" method="POST" enctype="multipart/form-data">

<div class="form-group">

<label for="image">Choose an Image</label>

<input type="file" class="form-control-file" id="image" name="image" required>

</div>

<button type="submit" class="btn btn-block" style="background-color: aquamarine;">Predict</button>

</form>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Prediction Result - Lung Cancer Prediction</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-image: url('https://img.freepik.com/free-photo/hands-with-lungs-shape\_23-2148533078.jpg?t=st=1731912641~exp=1731916241~hmac=b9e8e87c741c95f45c07c0e83dd6b48738f461889a2414cf82cfcadf2f7b447c&w=740');

background-size: cover;

background-position: center;

height: 100vh; /\* Ensure the background covers the entire screen height \*/

color: white; /\* Text color to stand out against the background \*/

}

.navbar {

width: 100%;

background-color: black;

}

.container {

max-width: 800px; /\* Fixed width for the form \*/

padding: 0 15px;

margin-top: 5rem; /\* To give some space from top \*/

background-color: rgba(0, 0, 0, 0.6); /\* Semi-transparent black background \*/

padding: 40px;

border-radius: 15px;

margin-top: 100px; /\* Adjust space from the top \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow for a lifted effect \*/

}

.text-center {

text-align: center;

}

.mt-5 {

margin-top: 5rem;

}

.img-fluid {

max-width: 100%;

}

/\* Ensure full-width navbar \*/

.navbar {

width: 100%;

position: fixed; /\* Keep the navbar at the top \*/

top: 0;

left: 0;

z-index: 10; /\* Make sure it is above other content \*/

}

.navbar-nav {

margin-left: auto;

}

/\* Padding to avoid content being hidden under the fixed navbar \*/

.container {

padding-top: 70px; /\* Adjust depending on navbar height \*/

margin-top: 200px;

}

.navbar-brand {

font-family: 'Helvetica', sans-serif; /\* Elegant font family \*/

font-weight: 700; /\* Bold text \*/

font-size: 1.75rem; /\* Larger text size \*/

letter-spacing: 1px; /\* Slight spacing between letters \*/

color: #f0f0f0; /\* Light color for @app.route('/')

def home():

return render\_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])

def predict():

if request.method == 'POST':

# Get the uploaded image

file = request.files['image']

if file:

# Save the image to a temporary location

image\_path = os.path.join('static', 'uploads', file.filename)

file.save(image\_path)

# Predict the class of the image

predicted\_class = predict\_image(image\_path)

class\_names = ['adenocarcinoma', 'large.cell.carcinoma', 'normal', 'squamous.cell.carcinoma']

predicted\_label = class\_names[predicted\_class]

# Render the result page with the prediction

return render\_template('result.html', predicted\_class=predicted\_label, image\_path=image\_path)

return render\_template('predict.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**REFERENCES**

* **Lung Cancer Prediction using Machine Learning Algorithms**. (2021). International Journal of Computer Applications (IJCA). Retrieved from <https://www.ijcaonline.org/>
* **Machine Learning for Lung Cancer Diagnosis and Prognosis**. (2020). Journal of Cancer Research and Clinical Oncology. DOI: 10.1007/s00432-020-03277-7
* **Early Detection of Lung Cancer Using Machine Learning Algorithms**. (2019). IEEE Transactions on Biomedical Engineering. DOI: 10.1109/TBME.2019.2905783
* **Machine Learning in Cancer Diagnosis: A Review**. (2018). Artificial Intelligence in Medicine. DOI: 10.1016/j.artmed.2018.01.003
* **The Role of Machine Learning in Cancer Treatment and Prediction**. (2021). Journal of Clinical Oncology. DOI: 10.1200/JCO.20.03049
* **Deep Learning for Cancer Diagnosis and Prognosis**. (2019). Nature Biomedical Engineering. DOI: 10.1038/s41551-019-0375-7
* **Regulations and Challenges in Healthcare Data Privacy**. (2020). Journal of Data Protection & Privacy. Retrieved from https://www.henrystewartpublications.com/jdpp
* **Machine Learning and Big Data in Healthcare: Opportunities and Challenges**. (2020). Journal of Healthcare Engineering. DOI: 10.1155/2020/8386721
* **The Impact of Deep Learning on Healthcare and Medical Imaging**. (2021). IEEE Access. DOI: 10.1109/ACCESS.2021.3055406
* **Data Privacy and Security in Healthcare: A Review of GDPR, HIPAA, and CCPA**. (2020). Health Information Management Journal. DOI: 10.1177/1833358320919194