Intelligent Radiologist Assistant (Lungs Tumor segmentation)

Architecture Document Design

Author: Tanjina Proma

Date: 12/09/2024

Table of Contents

1 Introduction 3

1.1 Scope 3

1.2 Objective 3

1.3 Significance of the project.. 4

2. System Architecture Overview 5

3. User I/O workflow.. 6

4. System Components 6

5. Components in Detail 7

6. Integration and Deployment.. 8

1 Introduction

1.1 Scope

The scope for Lung Tumor Classification from Chest CT Scans using VGG16 Model in TensorFlow, Python, Docker, MLflow, DagsHub, and DVC encompasses a comprehensive and scalable solution for automated diagnosis and treatment planning in lung cancer detection. Leveraging TensorFlow and Python, the project aims to develop a deep learning model based on the VGG16 architecture, capable of accurately classifying lung tumors as malignant or benign. Docker facilitates seamless deployment of the model, ensuring consistent execution across different environments. MLflow enables efficient experiment tracking, hyperparameter optimization, and model packaging, while DVC ensures versioning and reproducibility of datasets and model artifacts. Integration with DagsHub provides collaborative version control and continuous integration capabilities, streamlining project management and collaboration. The system promises to enhance diagnostic accuracy, streamline radiology workflows, and ultimately contribute to improved patient outcomes by enabling early detection and personalized treatment strategies for lung cancer patients.

1.2 Objective

The objective of this project is to develop a highly accurate and scalable system for lung tumor classification from chest CT scans using the VGG16 model implemented in TensorFlow, Python, Docker, MLflow, DagsHub, and DVC. By leveraging state-of-the-art deep learning techniques, the system aims to assist radiologists in accurately identifying and categorizing lung tumors as malignant or benign, thereby facilitating early diagnosis and personalized treatment planning for patients. Specifically, the objectives include, implementing the VGG16 model for feature extraction and classification, leveraging transfer learning to adapt the model to the specific task of lung tumor classification. Utilizing MLflow for experiment tracking, hyperparameter tuning, and model versioning to ensure reproducibility and scalability. Containerizing the model using Docker for seamless deployment across different environments, ensuring consistency and portability. Integrating with DagsHub and DVC for efficient collaboration, version control, and management of data and model artifacts, facilitating streamlined development workflows and enabling reproducible research practices. Overall, the objective is to deliver a robust and reliable system that enhances the accuracy and efficiency of lung tumor classification from chest CT scans, ultimately improving patient outcomes in the diagnosis and treatment of lung cancer.

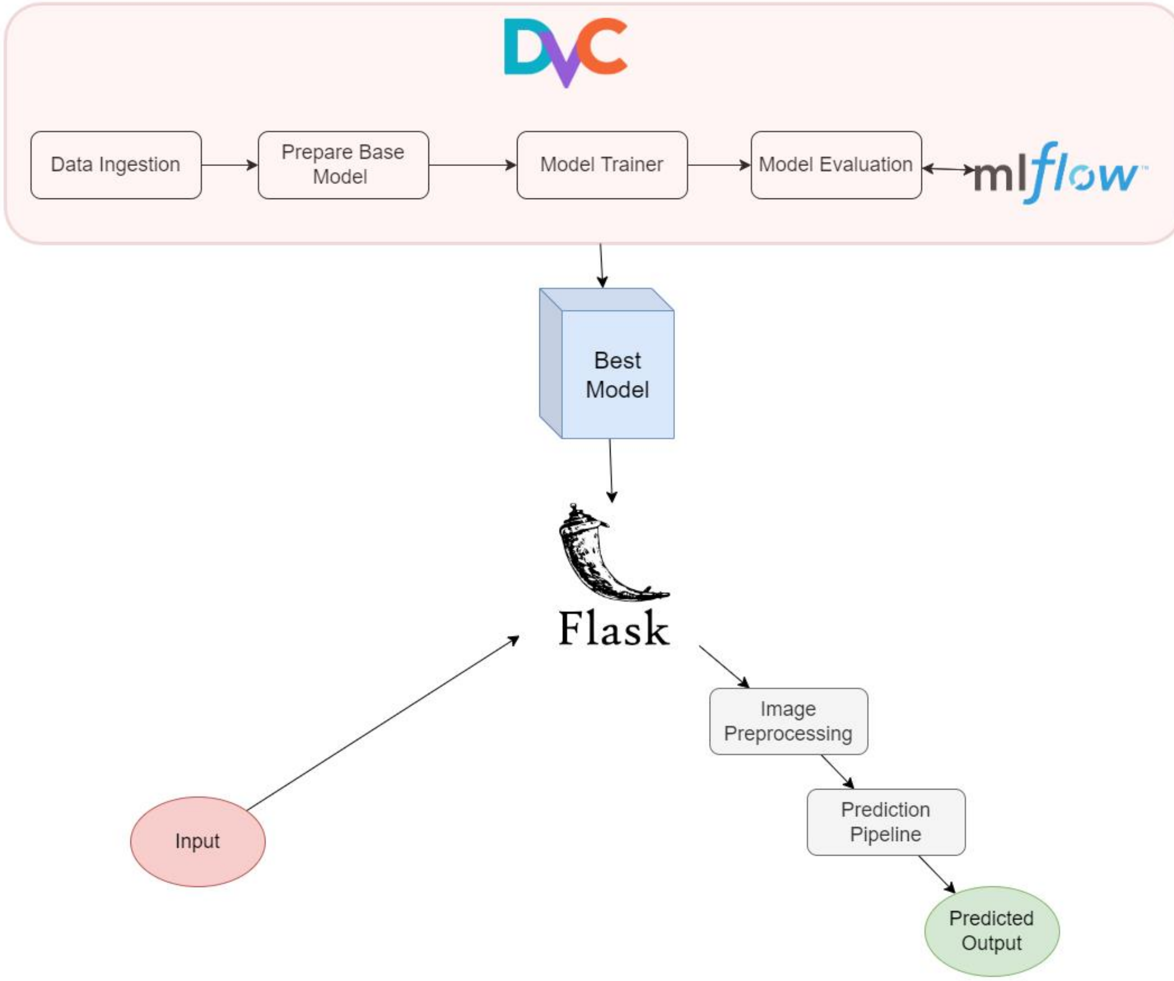
1.3 Significance of the project

Lung tumor classification from chest CT scans holds significant importance in the field of medical imaging and healthcare. By accurately identifying and categorizing lung tumors as malignant or benign, this approach aids radiologists and oncologists in making timely and informed decisions for patient diagnosis, treatment planning, and prognosis.

The use of the VGG16 model, a powerful deep learning architecture, combined with TensorFlow, enables robust feature extraction and classification from complex CT scan images. Docker ensures consistent and reproducible deployment of the model across various computing environments, facilitating seamless integration into clinical workflows. MLflow, DagsHub, and DVC enhance collaboration, version control, and reproducibility, ensuring transparency and accountability in the development and deployment process.

Ultimately, this approach enhances the efficiency and accuracy of lung cancer diagnosis, leading to earlier detection, personalized treatment strategies, and improved patient outcomes in the fight against one of the most prevalent and deadly forms of cancer worldwide.

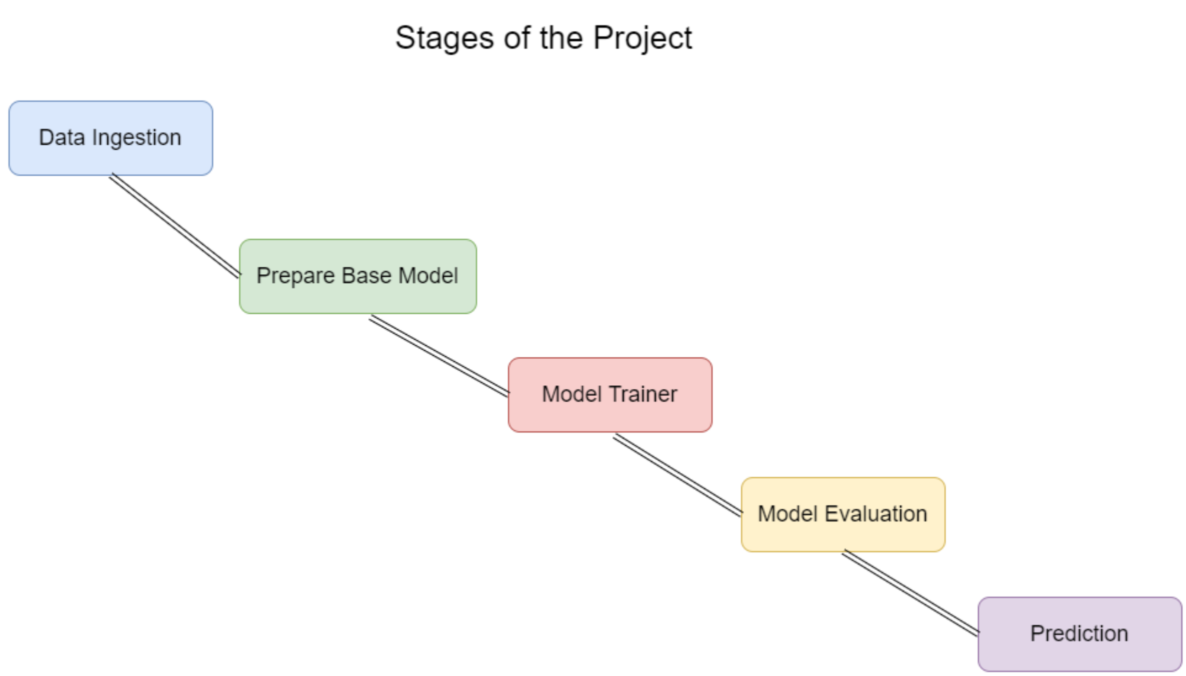
2. System Architecture Overview



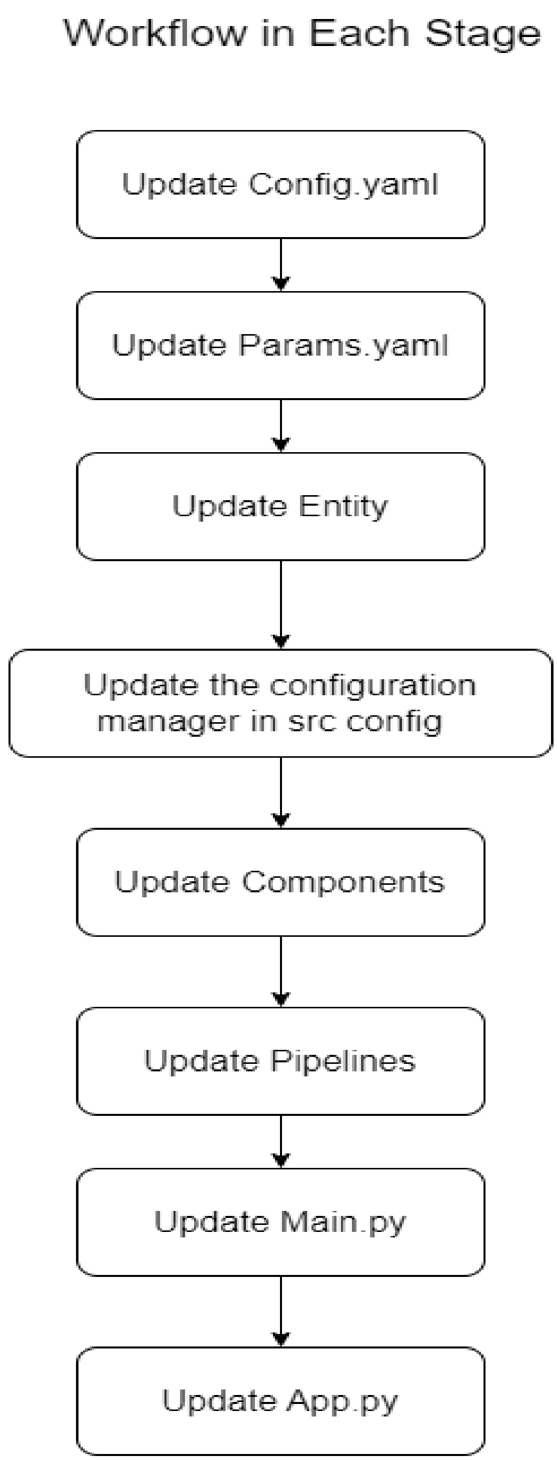
The system architecture for Lung Tumor Classification from Chest CT Scans using VGG16 Model in TensorFlow, Python, Docker, MLflow, DagsHub, and DVC is designed to provide a comprehensive and robust solution for automated tumor classification. At its core, the architecture consists of several interconnected components. Chest CT scan datasets are collected, preprocessed, and versioned using DVC, ensuring data integrity and reproducibility. The VGG16 model, implemented in TensorFlow and trained on the preprocessed data, is fine-tuned for lung tumor classification. MLflow tracks experiments, hyperparameters, and model versions. Model performance is evaluated using metrics such as accuracy and AUC-ROC. Docker containers encapsulate the trained model for deployment, while MLflow facilitates model serving. CI/CD pipelines, integrated with DagsHub, automate testing, building, and deployment processes, ensuring seamless integration and delivery of updates. This architecture provides a scalable, reproducible, and efficient solution for lung tumor classification, empowering healthcare professionals with reliable tools for early detection and treatment planning.

3. User I/O workflow

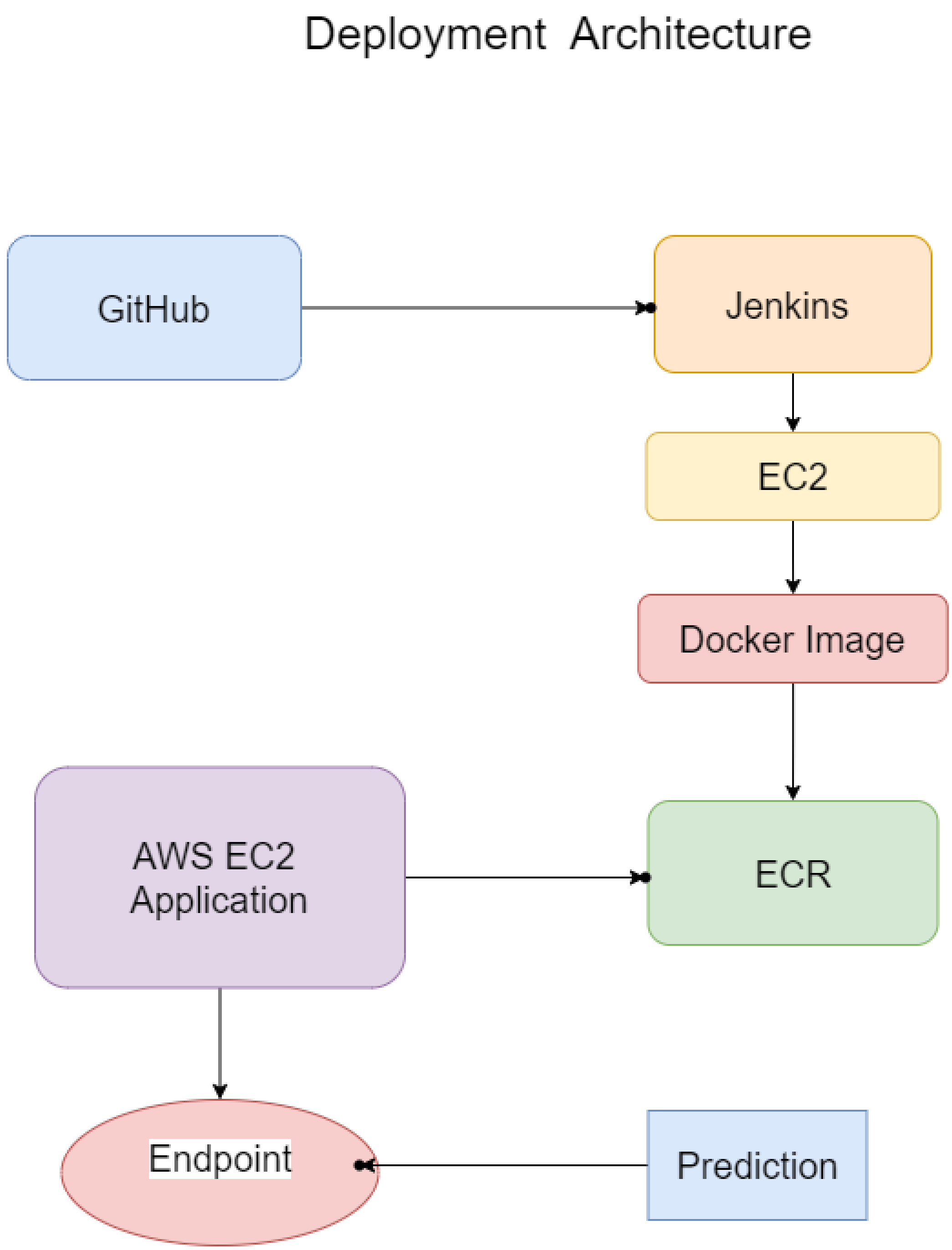
4. System Components



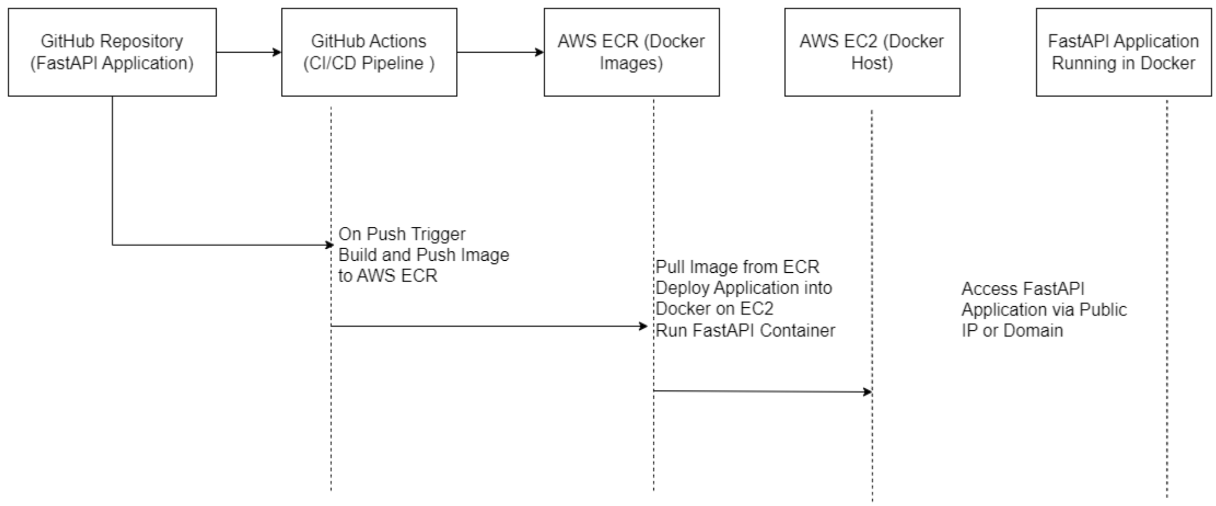
5. Components in Detail



6. Integration and Deployment



Wrokflow diagram for the Deployment Process



This workflow showcases the process starting from the GitHub repository, utilizing Jenkins for CI/CD to build and push Docker images to AWS ECR. Then, an EC2 instance pulls the Docker image from ECR and runs the FastAPI application within a Docker container. Finally, the application is accessible through the EC2 instance's public IP or domain.