**Hybrid CNN–Transformer Model for Automated Chest X-ray Disease Classification with Grad-CAM Explainability**

**Project Description**

Early and accurate detection of lung diseases is critical in modern healthcare. However, manual interpretation of chest X-rays is time-consuming and prone to human error. To address this challenge, the Hybrid CNN–Transformer Chest X-ray Classifier leverages deep learning to automate the identification of thoracic diseases such as Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, and Pneumonia.

The model fuses the strengths of ResNet18 (CNN) — known for extracting fine-grained spatial and textural details — with the Swin Transformer, which captures long-range dependencies and contextual information within medical images. The fusion results in a powerful hybrid network capable of detecting both localized and structural abnormalities in chest radiographs.

The system is enhanced with Grad-CAM (Gradient-weighted Class Activation Mapping) for explainability, providing visual insight into the specific lung regions influencing the model’s decision. The complete solution is deployed through an interactive Streamlit web interface featuring voice-based predictions, dropdown-based result exploration, and PDF report generation with personalized recovery advice and medication suggestions

### Scenario 1: AI-Assisted Hospital Diagnostics

In ahospital setup, radiologists and technicians upload patient chest X-rays into the system. The AI model instantly analyzes the image using the hybrid CNN–Transformer architecture and classifies potential diseases such as *Pneumonia* or *Cardiomegaly*. Grad-CAM visualization highlights critical lung regions, while voice feedback announces the diagnosis. The system helps reduce the diagnostic load, prioritize critical patients, and support radiologists with rapid, explainable predictions.

**Scenario 2: Remote Clinic Diagnostic Support**

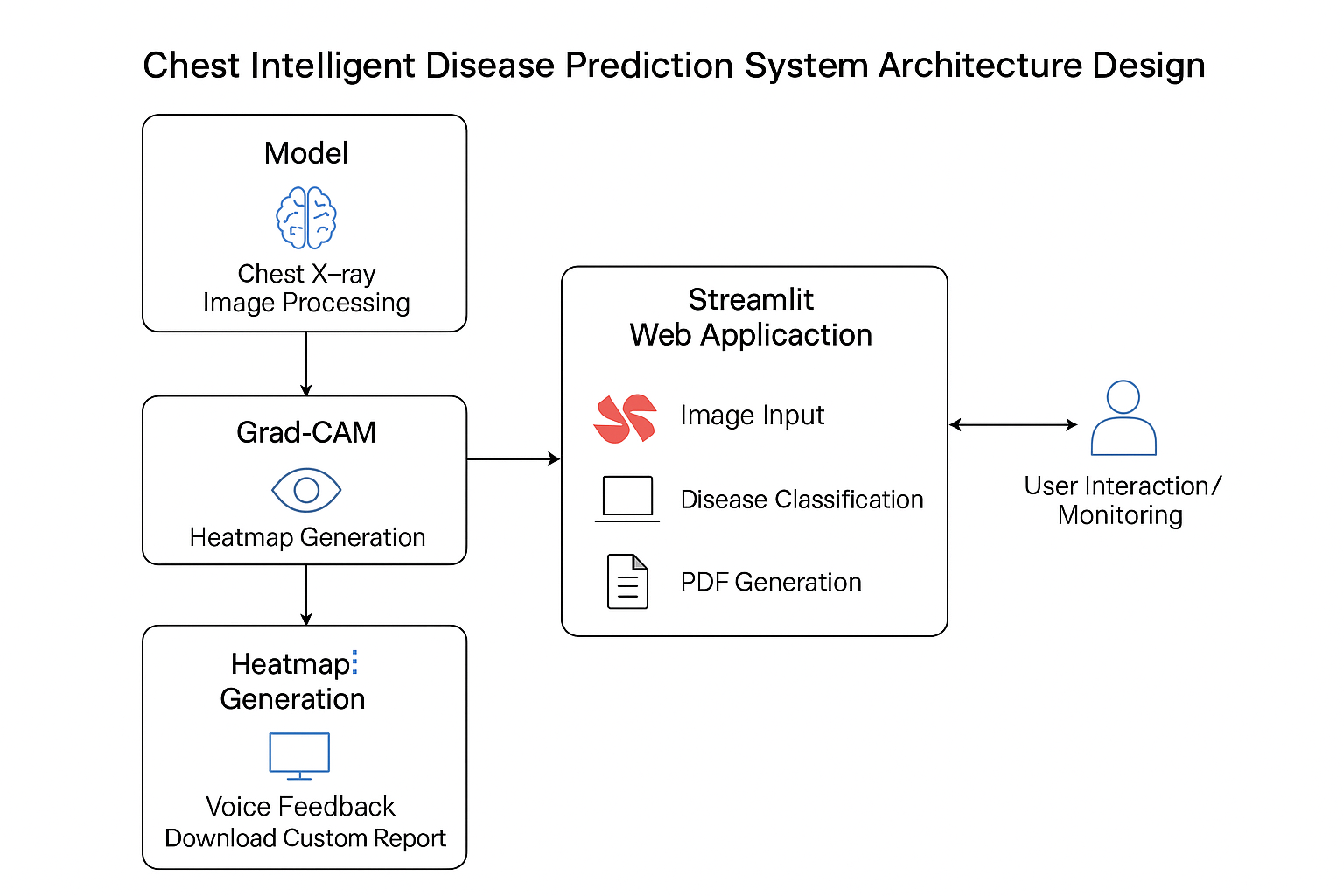
In rural or semi-urban healthcare centers with limited access to radiologists, the AI tool acts as an on-site assistant. Medical staff upload patient X-rays into the Streamlit interface, and the model provides immediate analysis with visual Grad-CAM heatmaps and spoken feedback. It also suggests initial medical recommendations and medications. The system enables faster detection, better triage, and improved patient management without specialist intervention..

**Scenario 3: Medical Training and Research Assistant**

In medical institutions and research labs, students and researchers use the AI model to study disease classification and interpretability. They upload chest X-rays, observe Grad-CAM visualizations, and compare how CNN and Transformer layers capture features. This setup helps learners understand deep learning decision-making in medical imaging and encourages research in explainable AI for healthcare.

**Architecture Overview**

The system architecture is composed of three primary layers — **Model Fusion**, **Explainability**, and **Interactive Deployment** — working in harmony to provide an interpretable, user-friendly diagnostic experience.



**Component Overview**

* **HybridFusionModel:** Combines CNN (ResNet18) and Transformer (Swin-Tiny) features for enhanced visual understanding.
* **Grad-CAM Explainability:** Generates visual heatmaps highlighting critical lung regions influencing classification.
* **Streamlit Frontend:** Interactive user interface for image uploads, result visualization, and report generation.
* **Voice Engine (pyttsx3):** Provides audio feedback announcing the predicted disease.
* **FPDF2 Report Generator:** Creates downloadable PDF diagnostic reports with advice and recommendations.

**Core Technologies**

* **PyTorch & timm:** Model creation and fusion using ResNet18 and Swin Transformer architectures.
* **torchcam:** Grad-CAM explainability for deep visual interpretation.
* **Streamlit:** Web interface for user interaction and visualization.
* **pyttsx3:** Voice synthesis engine for auditory feedback.
* **FPDF2:** Generates professional diagnostic reports.
* **Matplotlib & Pillow:** Visualization and overlay rendering.

**Component-Wise Architecture**

* **Preprocessing Layer:** Converts uploaded X-ray to normalized tensor format for model inference.
* **Feature Extraction Layer:** Extracts multi-level visual features from ResNet and Swin Transformer.
* **Fusion Layer:** Merges extracted features via 1×1 convolution for joint representation.
* **Classification Layer:** Predicts probability scores for six disease classes.
* **Explainability Layer :** Uses Grad-CAM to visualize attention maps.
* **UI Layer:** Displays predictions, probabilities, and heatmaps interactively.

**Project Workflow**

### 1. Model Design & Initialization

**Activity 1.1:** Configure the development environment with Python, PyTorch, and necessary deep learning libraries (timm, torchcam, torchvision).  
 **Activity 1.2:** Import pretrained architectures — **ResNet18** for convolutional feature extraction and **Swin-Tiny Transformer** for attention-based global feature mapping.  
 **Activity 1.3:** Construct the **HybridFusionModel** class combining both models, dynamically initializing the fusion layer during the first forward pass.  
 **Activity 1.4:** Validate feature compatibility and ensure smooth concatenation of CNN and Transformer outputs.

### 2. Dataset Preparation & Preprocessing

**Activity 2.1:** Collect and organize the **Chest X-ray Dataset (Dataset2.0)** containing six target diseases.  
**Activity 2.2:** Apply image transformations — resizing to (224×224), normalization using ImageNet mean and standard deviation, and tensor conversion.  
**Activity 2.3:** Split the dataset into **training (80%)** and **validation (20%)** subsets.  
**Activity 2.4:** Implement DataLoader pipelines with batching, shuffling, and augmentations to optimize GPU memory usage.

### 3. Model Training & Optimization

**Activity 3.1:** Define loss and optimization functions — CrossEntropyLoss and Adam optimizer.  
**Activity 3.2:** Integrate a **CosineAnnealingLR** scheduler for adaptive learning rate decay.  
**Activity 3.3:** Train the model over multiple epochs, tracking accuracy and loss metrics.  
**Activity 3.4:** Save the best-performing model weights as hybrid\_fusion\_gradcam\_best.pth after achieving optimal validation accuracy.

### 4. Explainability & Grad-CAM Integration

**Activity 4.1:** Implement **Grad-CAM** (Gradient-weighted Class Activation Mapping) using the torchcam library.  
**Activity 4.2:** Target the final fusion convolutional layer (fusion\_conv.2) to visualize discriminative regions in lung images.  
**Activity 4.3:** Normalize and overlay Grad-CAM heatmaps onto the original chest X-rays for intuitive interpretability.  
**Activity 4.4:** Validate that heatmaps correspond to true pathological areas such as lung opacity, consolidation, or fluid zones.

### 5. Streamlit Interface Development

**Activity 5.1:** Develop a responsive **Streamlit-based dashboard** for interactive inference.  
**Activity 5.2:** Implement user input components for **X-ray image upload** and dynamic prediction display.  
**Activity 5.3:** Integrate Grad-CAM visualization, dropdown probability menus, and real-time voice feedback using pyttsx3.  
**Activity 5.4:** Add a **medical-themed background (lungs.jpeg)** and custom styling for a professional, accessible interface.

### 6. Health Recommendation & Reporting Module

**Activity 6.1:** Build a health advisory system that provides **recovery suggestions and medication recommendations** per disease class.  
**Activity 6.2:** Integrate **FPDF2** for automatic PDF generation containing predicted disease, Grad-CAM visualization, and recommendations.  
**Activity 6.3:** Include a **download button** for users to export personalized reports.  
**Activity 6.4:** Implement voice confirmation post-report generation for better accessibility.

### 7. Testing & Deployment

**Activity 7.1:** Perform end-to-end testing across different environments (macOS and Windows).  
**Activity 7.2:** Validate accuracy of predictions, Grad-CAM overlays, and system performance under various input sizes.  
**Activity 7.3:** Optimize the model loading process with Streamlit’s @st.cache\_resource to minimize inference latency.  
**Activity 7.4:** Package and prepare the project for deployment as a **local or cloud-based healthcare assistant tool**.

**MILESTONE 1: Hybrid Model Initialization and Environment Setup**

In this foundational stage, the goal is to **establish and configure the deep learning environment** required for developing and deploying the Hybrid CNN–Transformer Chest X-ray Classifier.  
This milestone focuses on setting up all dependencies, initializing pretrained networks, and validating their integration to ensure both architectures (ResNet and Swin Transformer) operate seamlessly for multimodal feature extraction.

Proper execution of this milestone ensures that the system is ready for downstream activities — such as dataset loading, fusion layer configuration, and Grad-CAM explainability — forming the backbone of the entire AI diagnostic pipeline.

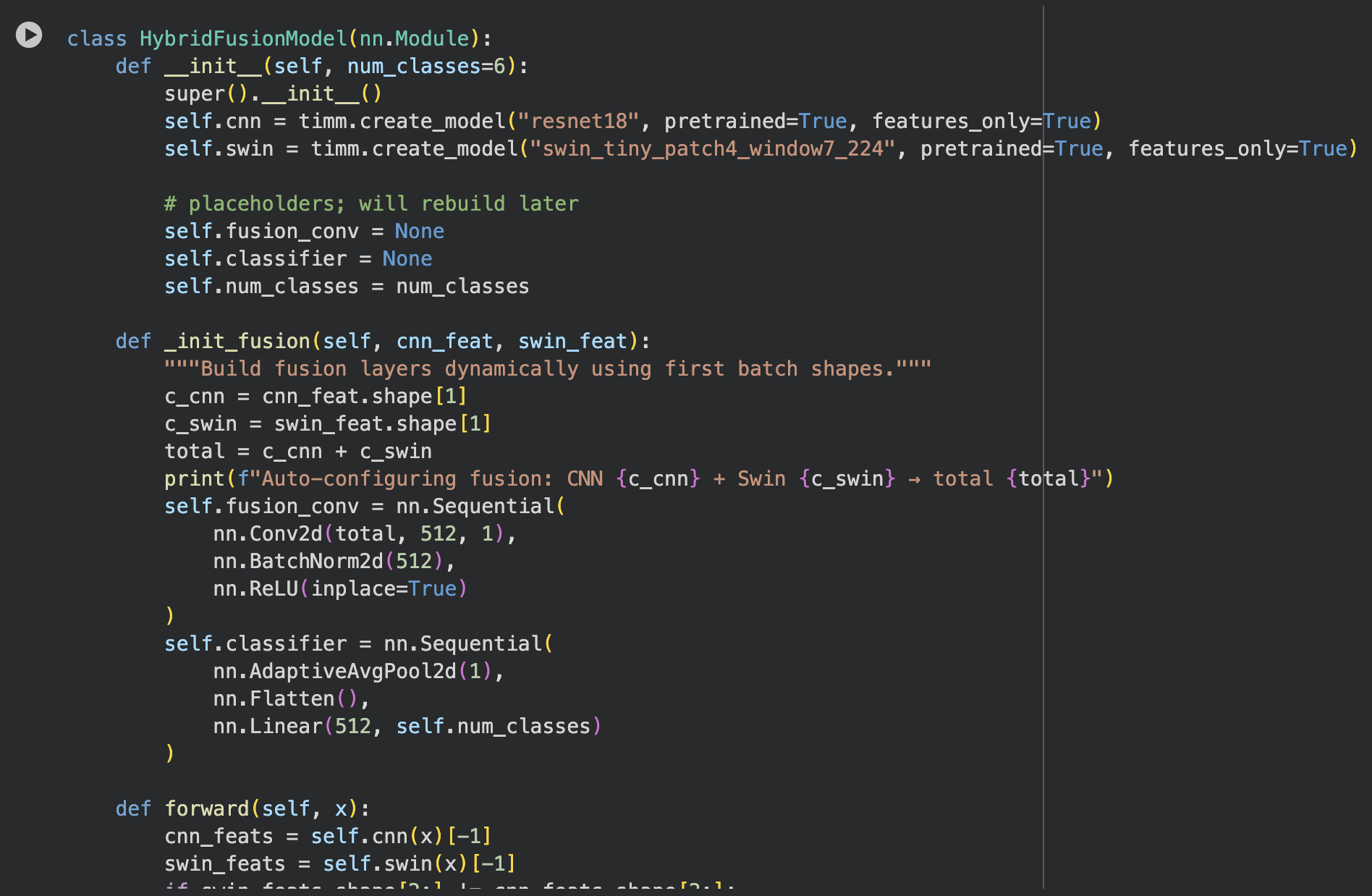
### Activities

**Activity 1.1:** Install essential dependencies — PyTorch, torchvision, timm, torchcam, matplotlib, and Streamlit.  
 **Activity 1.2:** Configure the GPU environment or fallback CPU device for training and inference.  
 **Activity 1.3:** Initialize pretrained backbones:

**ResNet18** for local texture and edge detection.

**Swin-Tiny Transformer** for long-range contextual analysis.

**Activity 1.4:** Create the **HybridFusionModel** class and verify first forward pass for dynamic layer initialization.

  
**Activity 1.5:** Conduct a quick test inference to confirm that both feature extractors and fusion layers are compatible.

  
**Activity 1.6:** Log architecture details and layer outputs for reference during Grad-CAM implementation.

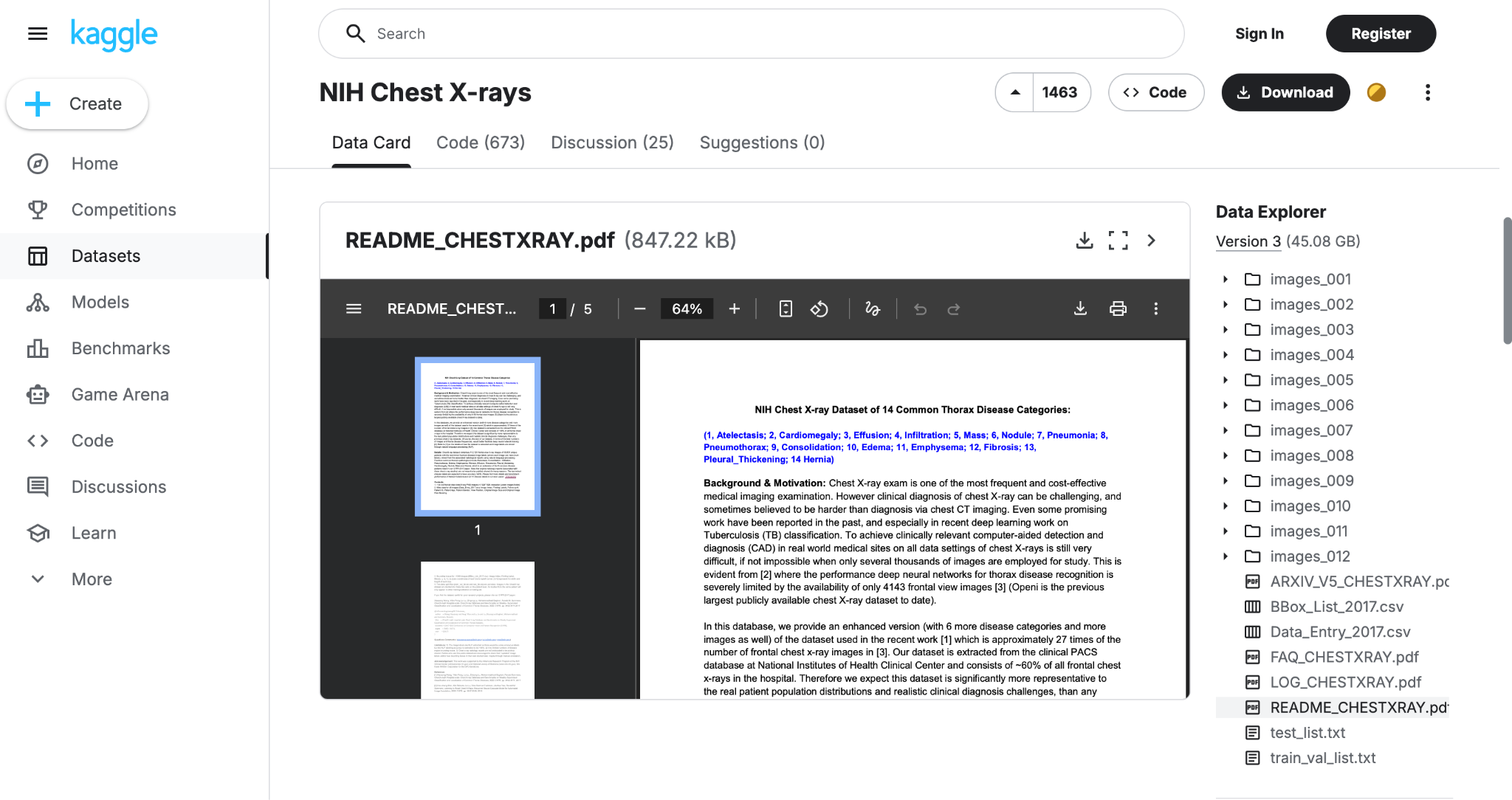
**MILESTONE 2: Dataset Preparation and Preprocessing**

**Objective:** Prepare and preprocess the chest X-ray dataset for training and evaluation to ensure balanced, high-quality data input for the hybrid model.

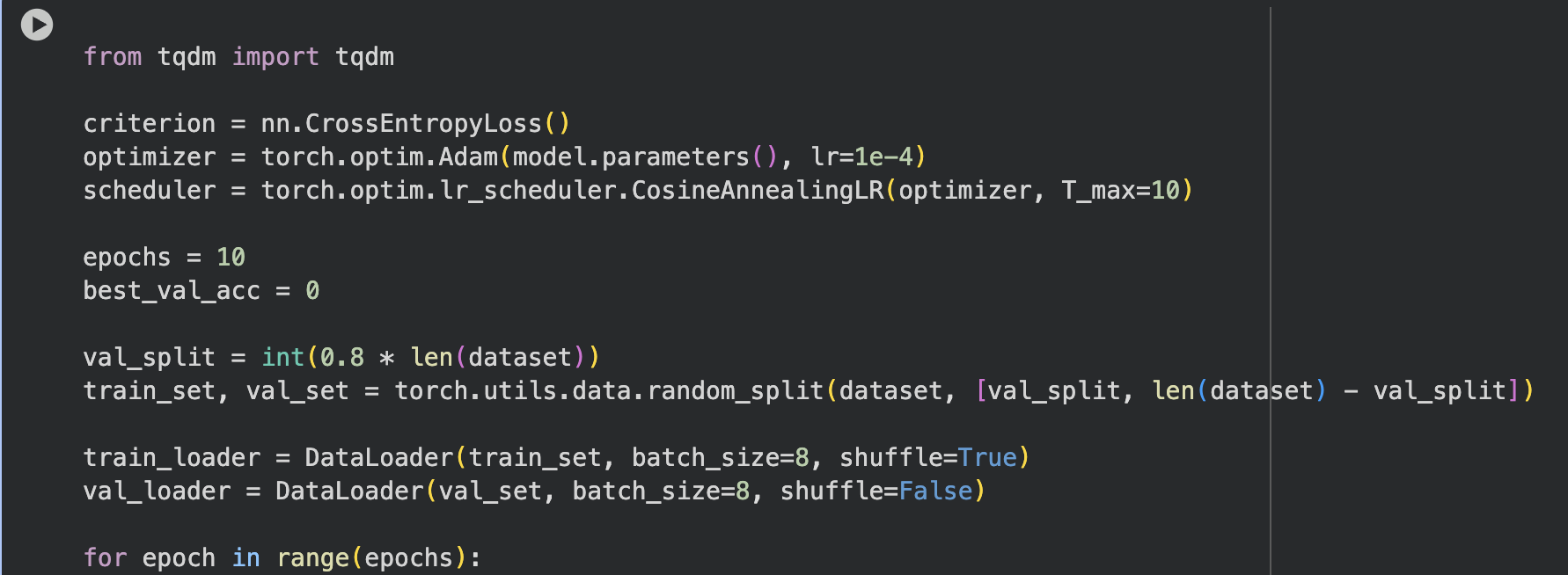
**Description:** This milestone focuses on curating the dataset, implementing preprocessing transformations, and ensuring proper batching for the training process. The goal is to maintain image quality and ensure uniform input resolution across all data samples.

**Activities:**

* **Activity 2.1:** Collect and organize the dataset (Dataset2.0) containing six target disease classes.(KAGGLE ChestX-Ray NIH)



* **Activity 2.2:** Apply preprocessing transformations – resizing to (224×224), normalization, and tensor conversion.
* **Activity 2.3:** Implement **DataLoader** instances with appropriate batch size and shuffling mechanisms.



* **Activity 2.4:** Split data into training and validation subsets (80:20).
* **Activity 2.5:** Augment data (flipping, rotation, brightness) to improve model generalization.
* **Activity 2.6:** Test sample visualization to verify correct data loading and label assignment.

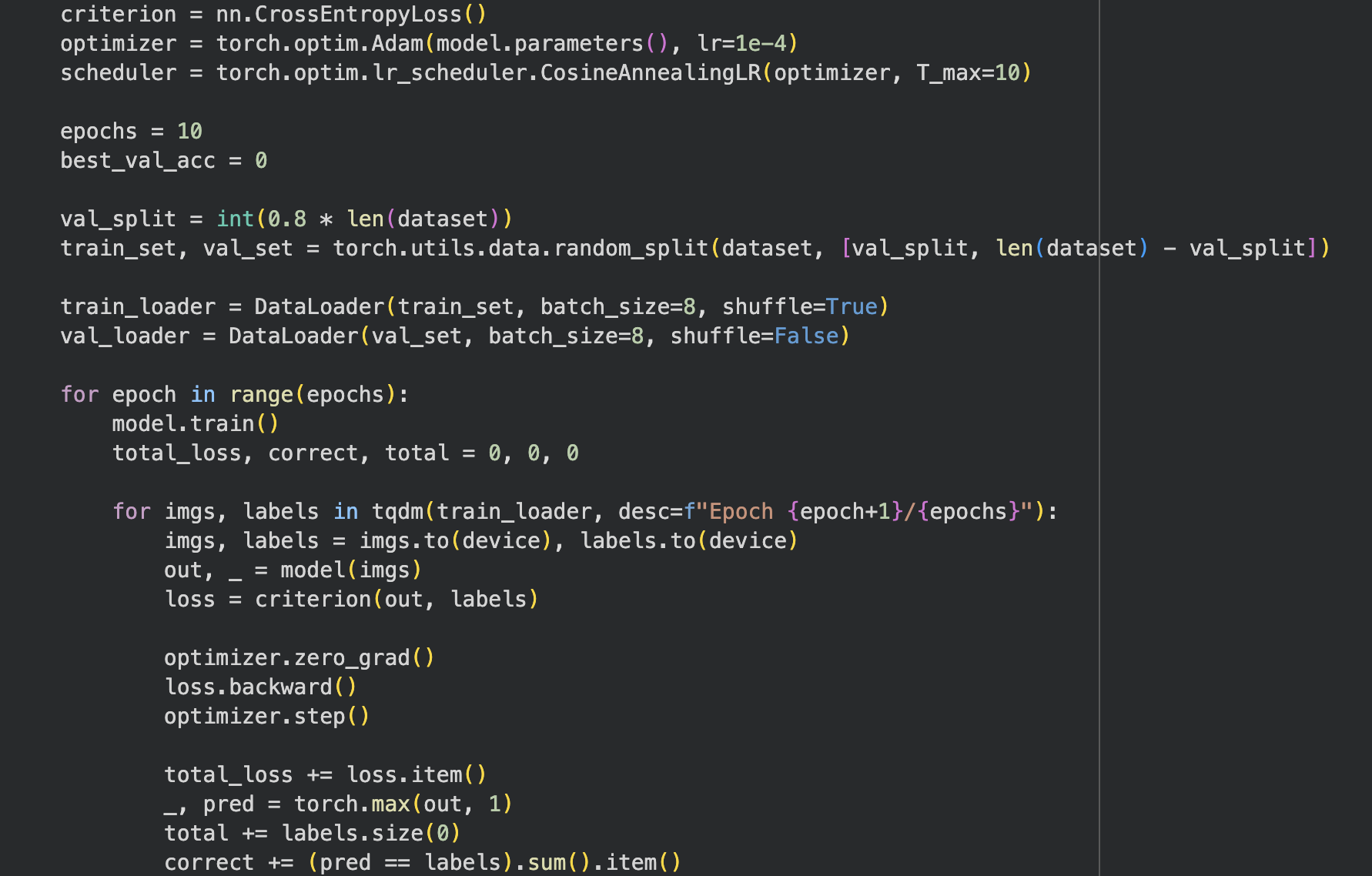
**MILESTONE 3: Model Training and Evaluation**

**Objective:** Train the HybridFusionModel using the preprocessed dataset, monitor its performance, and save the optimized version for deployment.

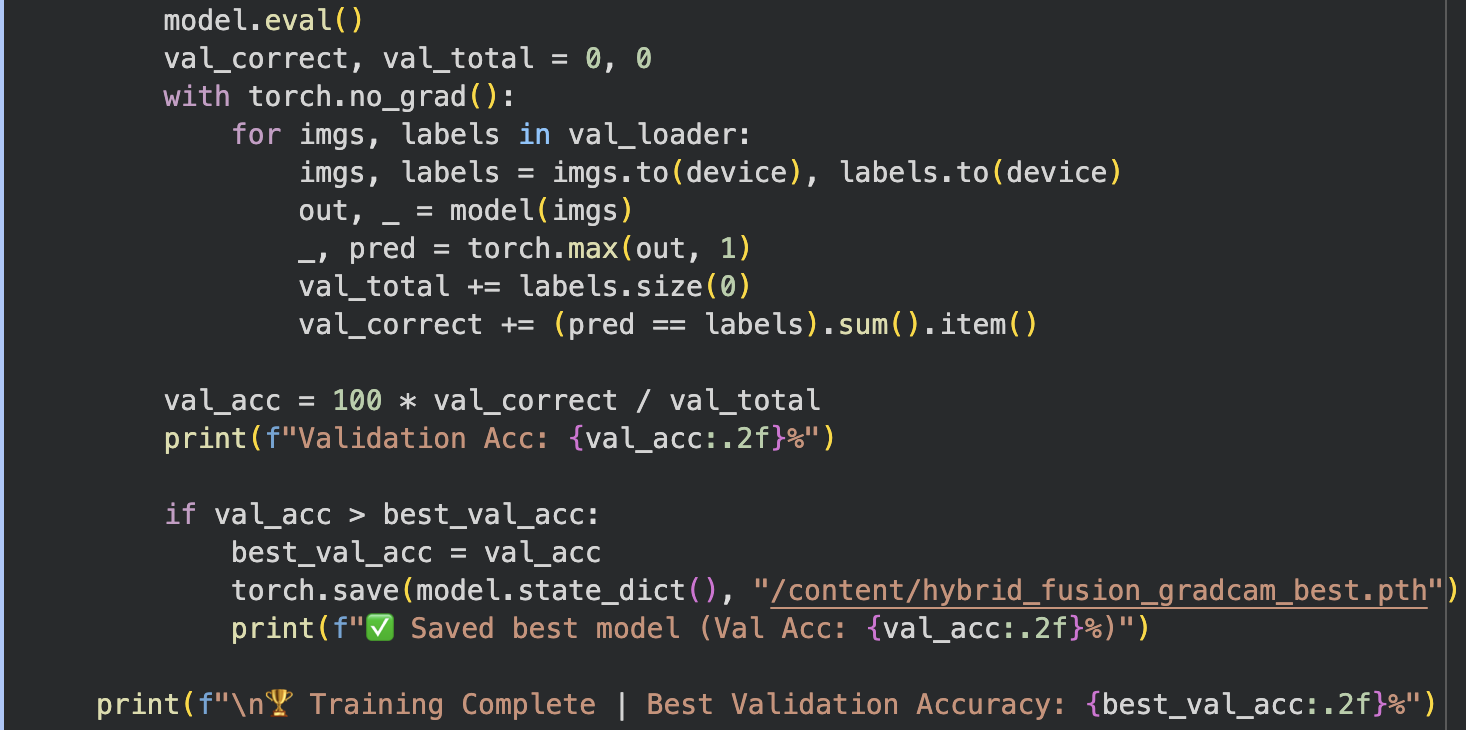
**Description:** This milestone involves defining the loss function, optimizer, and scheduler to train the model efficiently. Evaluation metrics like accuracy and validation loss are tracked to identify the best checkpoint. The trained model forms the backbone of the entire diagnostic system.

**Activities:**

* **Activity 3.1:** Define **CrossEntropyLoss** and **Adam** optimizer with learning rate = 1e-4.



* **Activity 3.2:** Implement a **CosineAnnealingLR scheduler** for adaptive learning rate control.
* **Activity 3.3:** Train the model for multiple epochs while tracking training and validation accuracy.
* **Activity 3.4:** Save the best model weights (hybrid\_fusion\_gradcam\_best.pth) when validation accuracy peaks.



* **Activity 3.5:** Perform loss analysis and visualize performance metrics over epochs.
* **Activity 3.6:** Document model accuracy, MAE, and RMSE for report inclusion.

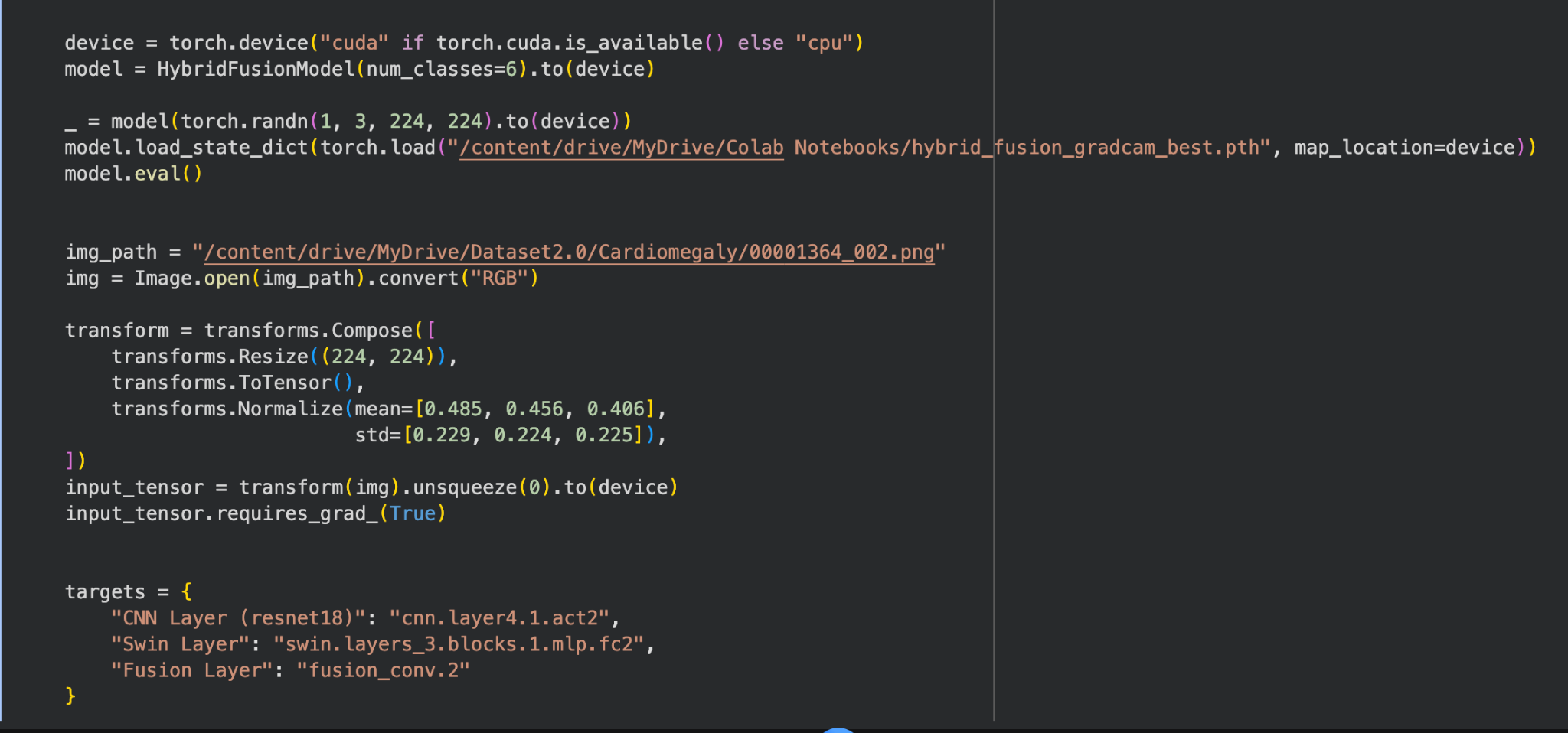
**MILESTONE 4: Explainability and Grad-CAM Integration**

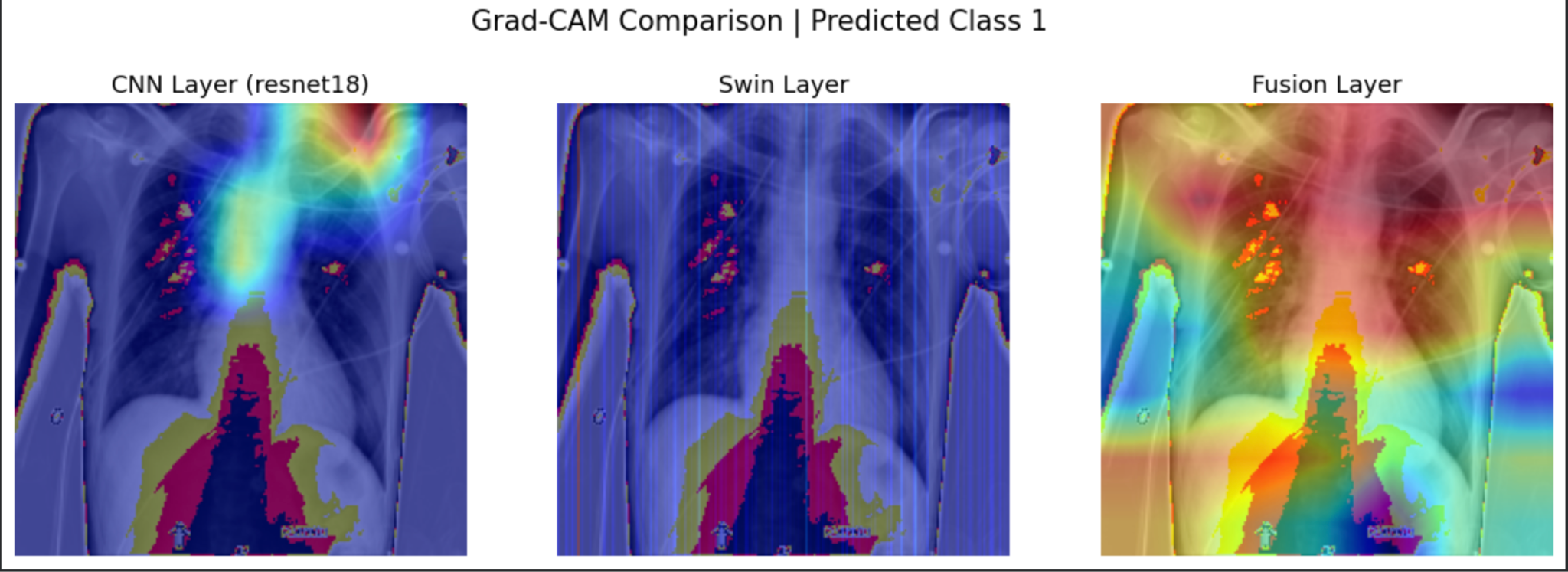
**Objective:** Incorporate interpretability into the diagnostic system using Grad-CAM visualizations to identify the regions influencing the AI’s decisions.

**Description:**This milestone introduces explainable AI into the workflow, enabling the model to visually highlight abnormal regions in lung X-rays. It enhances model transparency, allowing users to validate predictions through evidence-based heatmaps.

**Activities:**

* **Activity 4.1:** Implement **Grad-CAM** from the torchcam library, targeting the fusion convolution layer (fusion\_conv.2).

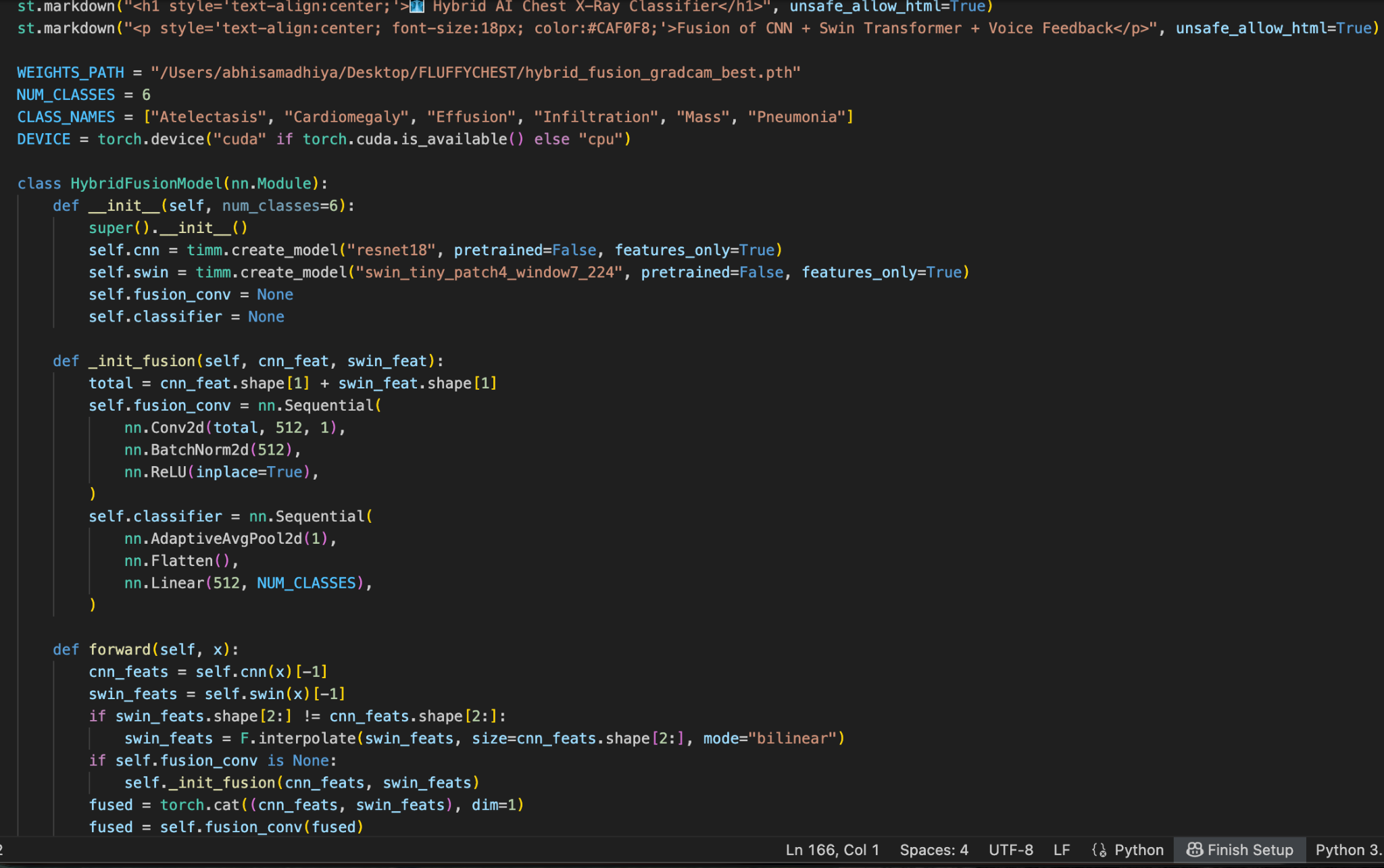


* **Activity 4.2:** Generate heatmaps for predicted classes and normalize outputs for better clarity.
* **Activity 4.3:** Overlay activation maps onto input X-rays using matplotlib for visual alignment.  
  
* **Activity 4.4:** Verify that highlighted areas correspond to actual pathological zones.
* **Activity 4.5:** Integrate Grad-CAM visualization into the Streamlit interface for real-time display.
* **Activity 4.6:** Test Grad-CAM outputs with multiple classes to ensure stability and interpretability.

## 

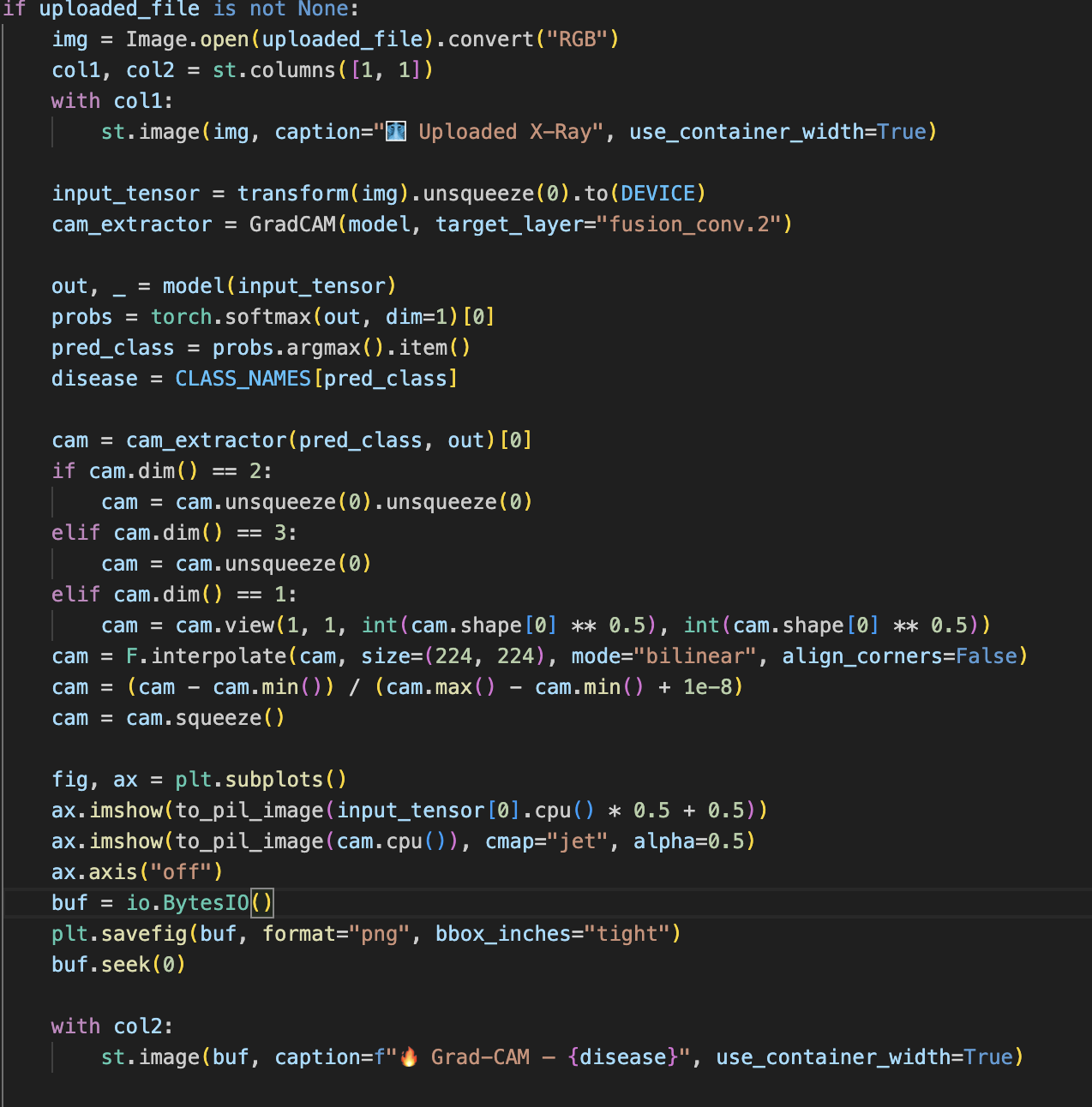
## MILESTONE 5: Streamlit Interface Development and Deployment

**Objective:**Develop an interactive web interface to deploy the model for real-world medical use with explainability, voice feedback, and reporting features.

**Description:**This milestone transforms the model into an accessible application through **Streamlit**, allowing users to upload chest X-rays, view predictions, and download diagnostic reports. The UI includes Grad-CAM overlays, dropdown probability displays, and a voice-enabled feedback system for accessibility.

**Activities:**

* **Activity 5.1:** Design a **Streamlit interface** with image uploader, prediction area, and visualization columns.
* **Activity 5.2:** Integrate **voice feedback** using pyttsx3 to announce detected conditions.
* **Activity 5.3:** Implement dropdown-based probability display for detailed class analysis

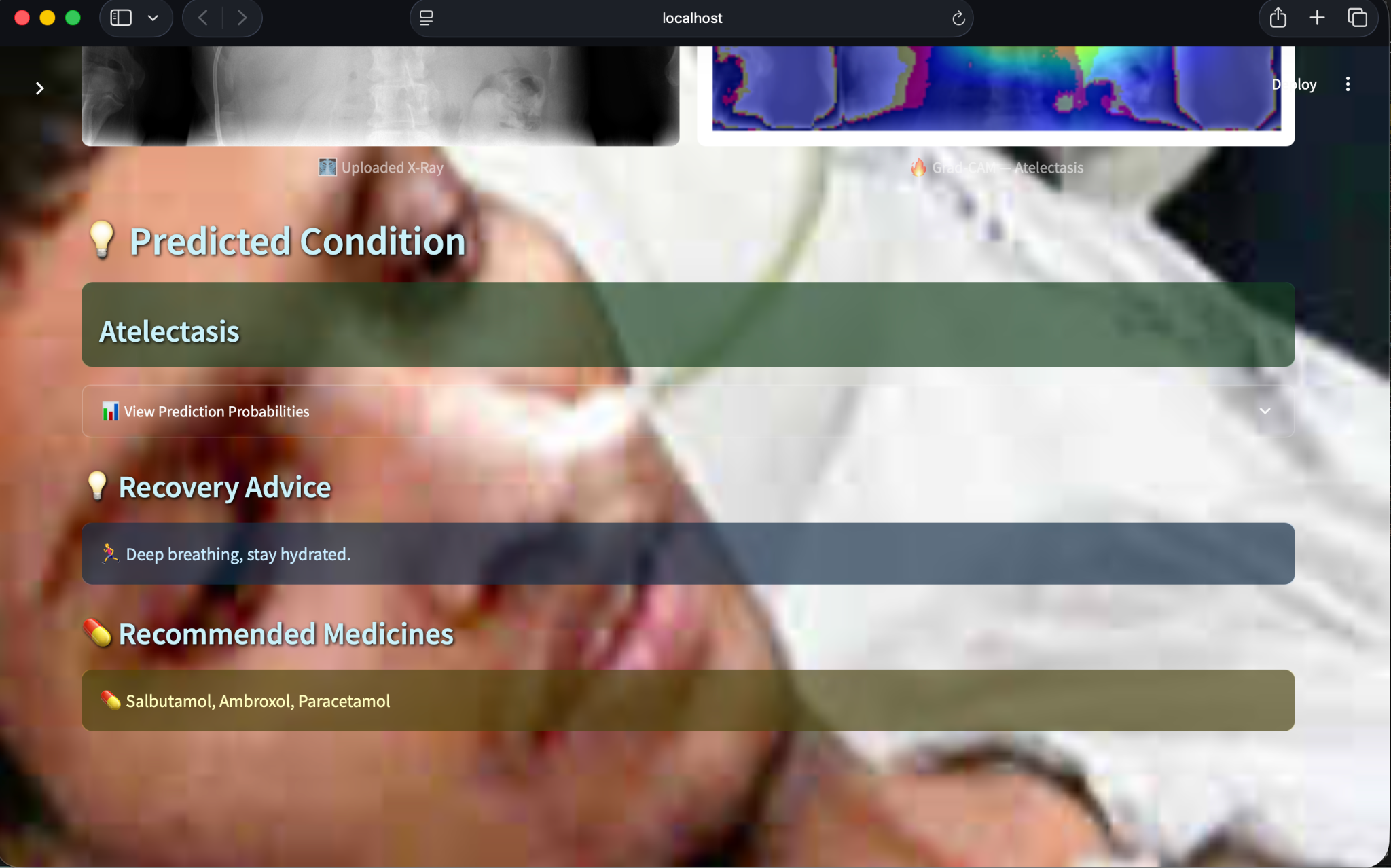


* **Activity 5.4:** Add **Grad-CAM heatmap overlays** for transparent diagnosis.
* **Activity 5.5:** Include a **recommendation system** displaying health advice and medicines.
* **Activity 5.6:** Implement **FPDF2** for generating professional diagnostic PDF reports.
* **Activity 5.7:** Test the app across systems (macOS, Windows) and optimize runtime via Streamlit caching.
* **Activity 5.8:** Deploy final version locally or to Streamlit Cloud for healthcare and educational access.

## 

## 

## OUTPUT:



## Future Enhancements

The current version of the **Hybrid CNN–Transformer Chest X-ray Classifier** establishes a strong foundation for automated, explainable radiographic analysis. However, several improvements can elevate its clinical reliability, scalability, and real-world adaptability.

### Planned Enhancements:

**1. Multi-Label Disease Classification**

* Enable the model to identify multiple co-existing conditions (e.g., *Pneumonia* + *Effusion*) within a single X-ray.
* Implement sigmoid-based multi-label heads to handle overlapping diagnoses more accurately.

**2. Integration with DICOM Standards**

* Add support for **DICOM (Digital Imaging and Communications in Medicine)** format to make the system compatible with hospital-grade imaging devices.
* Incorporate metadata (e.g., patient ID, scan date) for traceability and record management.

**3. Grad-CAM++ and Advanced Explainability**

* Extend beyond Grad-CAM to use **Grad-CAM++**, **Score-CAM**, or **Integrated Gradients** for improved attention visualization.
* Compare saliency maps for different model layers to increase interpretability for medical experts.

**4. Cloud and Edge Deployment**

* Deploy the model on **Streamlit Cloud, AWS, or Google Cloud Run** to make it accessible via web or mobile devices.
* Implement lightweight quantized versions of the model for **edge deployment** in low-resource clinical settings.

**5. Clinical Integration and Validation**

* Collaborate with medical institutions for real-world validation using clinical datasets.
* Implement feedback loops where radiologists can review predictions to improve the model’s confidence calibration.

**6. Multilingual Voice & Text Support**

* Extend pyttsx3 capabilities to include regional language voice feedback for accessibility.
* Provide multilingual on-screen recommendations for broader usability across healthcare networks.

**7. Continuous Learning Framework**

* Implement a feedback-based retraining mechanism that learns from new labeled data without full retraining.
* Use **federated learning** for privacy-preserving model updates across hospitals.

## Conclusion

The **Hybrid CNN–Transformer Chest X-ray Disease Classifier** represents an innovative fusion of **deep learning and medical imaging**, combining the **local feature precision** of CNNs with the **global contextual awareness** of Vision Transformers.  
 Through **feature fusion, Grad-CAM explainability**, and a **Streamlit-based interactive interface**, the system delivers accurate, interpretable, and user-friendly diagnostic assistance.

The project demonstrates how **AI can enhance clinical workflows** by reducing the diagnostic load, improving consistency, and enabling faster decision-making. The inclusion of **Grad-CAM heatmaps** and **voice-guided feedback** ensures that the tool remains transparent, accessible, and intuitive — even for non-technical users.

This work successfully bridges the gap between **data-driven automation** and **human interpretability**, creating a prototype suitable for research, education, and early-stage clinical testing.  
 With future integration of DICOM compatibility, cloud deployment, and multi-label prediction capabilities, this system has the potential to evolve into a **scalable AI healthcare assistant** for hospitals and diagnostic centers worldwide.