DEPI DoctorAI Project Documentation

Team Members: Adel Mahmoud, Menna Ateya, Mohamed Ahmed Tolba, Mohamed Osama, Mostafa Abdo, Yousef Alaa Supervisor: Eng. Sarah Abdelmoaty Institution: Digital Egypt Pioneers & CLS 15 May 2025

Abstract

This project presents a comprehensive AI system that combines multiple deep learning models for disease diagnosis and a medical chatbot for initial patient in teraction. The system classifies brain tumors, pneumonia, anemia, lung and colon cancer, skin cancer, monkeypox, and bone fractures. The chatbot is built using an LSTM-based seq2seq model with attention, utilizing medical-specific preprocess ing and BioWordVec embeddings. The final system is deployed using a Streamlit web interface, with robust Frontend, Backend, and MLOps support for real-time medical predictions

Contents

1. Introduction

Early detection of diseases is critical to ensure timely and effective medical treatment. This project integrates Natural Language Processing (NLP) and Deep Learning (DL) to create a unified platform that assists in medical diagnosis through interaction and image-based classification.

2. Related Work

Medical image classification has been an active area of research over the past decade, with deep learning playing a pivotal role in achieving high diagnostic accuracy. Many researchers have proposed models for detecting specific diseases using various imaging modalities.

For **brain tumor detection**, convolutional neural networks (CNNs) have shown significant success in distinguishing tumor types from CT and MRI scans. Models such as ResNet and VGG have been adapted to enhance feature extraction and improve classification accuracy.

In the domain of **chest X-ray analysis**, pneumonia detection has been widely studied. Datasets like ChestX-ray14 have enabled the development of deep CNN models capable of identifying pneumonia with expert-level performance. Techniques such as transfer learning using pretrained models (e.g., DenseNet121) have been shown to be effective.

For **anemia detection**, traditional blood tests remain standard, but recent studies have explored the use of palm image analysis to estimate hemoglobin levels. These approaches apply image processing and machine learning to detect pallor and other anemia indicators.

Regarding **lung and colon cancer**, histopathological image classification is a common approach. CNNs trained on microscopic images can differentiate between adenocarcinoma and benign tissues with high precision. Studies have demonstrated that using deeper networks with dropout and regularization can significantly reduce overfitting.

Skin disease detection — including skin cancer and monkeypox — has seen a surge in deep learning applications. Dermatological image datasets, such as ISIC for skin cancer and monkeypox-specific datasets, have been used to train CNNs by using DenseNet201 to classify lesions with dermatologist-level accuracy.

Finally, **bone fracture detection** has leveraged deep learning models trained on X-ray images to assist radiologists in identifying fractures automatically, especially in emergency or resource-limited settings.

This project builds upon these foundations by integrating multiple disease-specific classification models and combining them with a conversational chatbot interface, offering a user-friendly and comprehensive diagnostic assistant.

3. System Architecture

The system consists of:

- A medical chatbot for user interaction.
- Seven classification models trained on specific medical conditions.
- A unified Streamlit-based interface for deployment.

4. Medical Chatbot Module

The chatbot is built from scratch using an LSTM seq2seq model with attention.

- **Preprocessing**: `spaCy` with en_core_sci_sm for lemmatization, POS tagging, and NER.
- **Embeddings**: BioWordVec pretrained medical word vectors.
- **Model**: Encoder with 4-layer bidirectional LSTM and Decoder with 4-layer LSTM. The chatbot interacts with users to collect information before proceeding to diagnosis.

5. Disease Classification Models

In this project, we developed and integrated seven deep learning-based classification models, each designed to detect a specific disease or medical condition using image data. Each model was trained and evaluated separately using relevant datasets and tailored preprocessing techniques to ensure optimal performance. Below is an overview of each model:

5.1 Brain Tumor Detection

- **Input:** CT Scans
- **Model:** CNN
- **Output:** Tumor / No Tumor

This model identifies the presence of a brain tumor using CT scan images. A CNN-based architecture was trained on a dataset containing labeled brain scan images, distinguishing between tumor and non-tumor cases. Preprocessing included resizing, normalization, and grayscale conversion. The model achieved high classification accuracy and can serve as a fast screening tool in clinical settings.

5.2 Pneumonia Detection

- **Input:** Chest X-ray
- **Model:** CNN or ResNet
- **Output:** Pneumonia / Normal

Chest X-ray images were used to detect pneumonia cases using a deep CNN model. The dataset included both normal and pneumonia-affected X-rays. Preprocessing steps involved histogram equalization and image resizing. The model was fine-tuned to reduce false positives and negatives, showing promising performance in binary classification (pneumonia vs. normal).

5.3 Anemia Detection

- **Input:** Palm Images
- **Model:** CNN
- **Output:** Anemic / Non-anemic

Unlike traditional blood tests, this model estimates anemia likelihood through palm images. It classifies images based on visual signs of pallor, which may indicate low hemoglobin levels. A lightweight CNN was used, and color-based segmentation was applied during preprocessing to highlight the palm region. This approach provides a non-invasive method for preliminary anemia screening.

5.4 Lung & Colon Cancer Detection

- **Input:** Histopathological Images
- **Classes:** Lung Adenocarcinoma, Lung Squamous Cell Carcinoma, Colon Adenocarcinoma, Colon Benign Tissue
- **Architecture:** Two Conv2D + MaxPooling layers, Flatten, Dense(128), Dropout(0.5), Dense(4, Softmax)
- **Accuracy:** 97.5%

We used a CNN model trained on histopathological images to classify lung and colon tissue samples into four categories:

- Lung Adenocarcinoma
- Lung Squamous Cell Carcinoma
- Colon Adenocarcinoma
- Colon Benign Tissue

The dataset was obtained from Kaggle and included high-resolution microscopic images. The model architecture comprised two convolutional layers, max-pooling, flattening, dense layers, and dropout regularization. It achieved an accuracy of **97.5%**, demonstrating strong potential for aiding pathologists.

5.5 Skin Cancer Detection

- **Input:** Skin Cancer MNIST: HAM10000
- **Model:** Custom CNNs
- **Model Architecture:**

Layer Type	Count
• InputLayer	1
• Conv2D (Convolution)	7
• MaxPooling2D	4
• BatchNormalization	7
• Dense (Fully Connected)	5
• Flatten	1
• Dropout	1

- **Output:** melanocytic nevi / melanoma / benign keratosis-like lesions / basal cell carcinoma / pyogenic granulomas and hemorrhage / Actinic keratoses and intraepithelial carcinomae

Using dermatoscopic images, this model classifies skin lesions as either benign or malignant. It is based on a pretrained **ResNet50** model fine-tuned on a labeled dataset of skin lesion images. Preprocessing involved data augmentation (rotation, flipping, zooming) to reduce overfitting. The model is capable of supporting dermatologists in early skin cancer detection.

5.6 Monkeypox Detection

- **Input:** Monkeypox Skin Lesion Dataset
- **Model:** Fine-tuned DenseNet201
- **Output:** Monkeypox / Non-monkeypox

This binary classification model distinguishes between monkeypox-affected and non-monkeypox skin images. A pretrained DenseNet201 architecture was fine-tuned and trained on a specialized monkeypox dataset. Image preprocessing included resizing and normalization, with augmentation techniques applied to improve generalization. The model provides fast and reliable predictions, especially in outbreak scenarios.

5.7 Bone Fracture Classification

- **Input:** X-ray Images
- **Model:** CNN
- **Output:** Fractured / Normal or Type of Fracture

This model analyzes X-ray images to detect bone fractures. It uses a CNN trained on labeled fracture and non-fracture images. Preprocessing included contrast enhancement and edge detection to improve feature extraction. The model assists in automating orthopedic diagnoses and is particularly useful in emergency or low-resource environments.

6. Implementation Details

- **Language:** Python
- **Libraries:** TensorFlow, PyTorch, spaCy, Streamlit, NumPy, Pandas
- **Deployment:** Streamlit web app with real-time image input and prediction output

7. Evaluation Metrics

- Accuracy
- Precision, Recall, F1-Score
- Confusion Matrix for each model

8. Challenges and Limitations

- Imbalanced datasets across some categories
- High variability in medical image quality
- Complex NLP preprocessing for chatbot
- Maintaining performance across all models

9. Future Work

- Integrate voice support in chatbot
- Improve accuracy with transfer learning
- Include more diseases like diabetic retinopathy
- Add electronic health record (EHR) integration

10. Conclusion

In this project, we developed a medical assistant system that combines multiple deep learning classification models with a chatbot interface. Each model was designed to detect a specific disease using medical images, such as CT scans, X-rays, and skin photos.

The system achieved high accuracy across all models, including brain tumor, pneumonia, anemia, lung and colon cancer, skin cancer, monkeypox, and bone fracture detection. The chatbot, built using LSTM with attention, guides users before uploading images.

Overall, the system shows strong potential as a fast and accessible tool to support early disease diagnosis and reduce the workload on medical professionals.

11 Development Workflow

This section outlines the development process, including Frontend, Backend, and MLOps components.

11.1 Frontend Development

The Frontend is built using React with Node.js 18+, providing a user-friendly interface for healthcare professionals and patients.

• Framework: React with modern JavaScript syntax and JSX.

• Styling: Tailwind CSS for responsive design.

• Features: Image upload, chatbot interaction, and real-time prediction display.

• Setup:

– Install dependencies: `npm install`.

– Configure environment variable: `VITE_API_URL=http://localhost:8000 in .env.`

– Build and deploy: `npm run build` and deploy via Azure Static Web Apps.

11.2 Backend Development

The Backend is implemented using FastAPI with Python 3.7+, ensuring scalability and security.

• Framework: FastAPI with Redis caching.

• Security: JWT-based authentication.

• Features: API endpoints for chatbot, image diagnosis, and anemia detection.

• Setup: – Install dependencies: pip install-r requirements.txt.

– Configure environment variable: `SECRET_KEY=your-secret-key in .env`.

– Run with Docker: docker-compose up–build.

– Deploy to Azure App Service using Docker images.

11.3 MLOps

MLOps pipelines ensure model training, deployment, and monitoring.

• Tools: Docker, Azure Blob Storage, Git LFS.

• Model Management:

– Store models (e.g., `bone_fracture.tflite, DEPI_SKIN_CANCER_MODEL.h5`) in Azure Blob Storage.

– Download: `az storage blob download-batch-source models`

– destination ./models.

• Deployment: Automated CI/CD with Docker for scalable inference.

• Monitoring: Track model performance and retrain with new data