WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Lung Infection Classification Using CT Images in DEEPLEARNING

(1)P.KAVITHAPANDIYAN , (2)R.SHANMUGAPRIYA, (3)S.THANUSRI, (4)S.DIVYADARSHINI

[1)]ASSOCIATIVE PROFESSOR(CSE), [(234)]FINIAL YEAR CSE
Department Of Computer Science and Engineering
Sree Sowdambika College of Engineering, Tamil Nadu, India

ABSTRACT—

This paper presents a comprehensive web-based application that leverages deep learning for the classification of lung infections using CT images. It aims to address the rising need for accurate and timely diagnosis of lung-related diseases such as pneumonia and COVID-19. The system utilizes a Convolutional Neural Network (CNN) model to perform image classification into three categories: Normal, Pneumonia, and COVID-19. The model is integrated into a Django-based full-stack web framework that supports dual-role access – one for administrators who manage patient records, and another for medical professionals who upload CT images and receive diagnostic outputs. The backend relies on SQLite for data persistence, and the system incorporates several popular Python libraries including TensorFlow, Keras, NumPy, Pandas, and Matplotlib. The model achieved an overall accuracy of 96.2% and demonstrated high precision in real-time clinical simulation.

Keywords— Deep Learning, CNN, COVID-19, Pneumonia, CT Images, Django, Full-stack, SQLite.

Introduction

Lung diseases have been a persistent global health issue, particularly with the emergence of COVID-19, which added an urgent demand for advanced diagnostic systems. CT scans provide detailed imagery of lung structures and are essential in identifying abnormalities. However, manual interpretation of these images is subject to human error, requires expert radiologists, and is time-consuming. By integrating deep learning into healthcare, particularly Convolutional Neural Networks (CNN), we can enable automated diagnosis that improves speed, accuracy, and -+scalability. This system is intended to assist healthcare professionals by providing instant, reliable insights based on CT image analysis.

In this paper, we propose a deep learning-based lung infection diagnostic system. Our key contributions include:

- **A full-stack web application** built using Python and Django, enabling real-time classification of lung CT images into COVID-19, pneumonia, or normal cases, accessible through doctor and admin portals..
- **A CNN-based image classification engine**, trained using TensorFlow and Keras, which achieves high accuracy (96.2%) and automates feature extraction without requiring radiologist intervention..
- **A secure and user-friendly web interface** that allows doctors to upload CT images and instantly receive diagnostic predictions, while administrators manage patient records and historical data through a centralized database (SQLite).
- **A comprehensive evaluation and case study** showing the model's robustness, responsiveness, and real-world applicability for clinical decision support, tested with sample patient uploads in a realistic healthcare setti

Figure 1

System Design

Figure 1 illustrates the overall system architecture consisting of a frontend interface, a backend server, and a deep learning engine.

2.1 User (Doctor) Module

Doctors access the system through a secure login. They can upload CT images of patients for immediate diagnosis. The uploaded image is processed by a CNN model integrated into the Django server. The model outputs the classification result: Normal, Pneumonia, or COVID-19. The result is displayed instantly and stored in the database along with patient information..

Admin Module

The admin login provides access to a dashboard for managing patient data, image history, and system usage logs. Admins can view prior diagnoses, manage user access, and maintain records. The admin interface is built using Django's admin tools and connected to the same SQLite database used by the doctor module.

Dataset

- The dataset contains **several hundred to thousands of CT images** per class, ensuring a balanced distribution.

Fig2: Normal Lung

Fig3: Covid Lung

Fig4: Pneumonia Lung

- Images vary in resolution and format, so preprocessing was performed to **standardize dimensions (e.g., 224×224 pixels)** and normalize pixel values.
- Metadata such as patient ID, infection type, and scan modality were used during labeling and stored along with prediction results.

Prediction Pipeline

When an image is uploaded, the backend initiates the prediction pipeline: the image is preprocessed (resized and normalized), fed into the CNN, and the classification result is returned. The process is completed within a few seconds. The result, along with patient ID and timestamp, is saved to the database.

```
python
Copy code
patient = models.CharField(max_length=255)
    xray = models.ImageField(upload_to='xray/')
    disease = models.CharField(max_length=255)
    encoding = models.BinaryField(default=default_encoding, null=True, blank=True)
    uploaded_at = models.DateTimeField(auto_now_add=True)
```

Sample of lung infection clasification. (*sample prediction*)

Implementation

CNN Model Training

A Convolutional Neural Network (CNN) was developed using TensorFlow and Keras. The dataset consisted of labeled lung CT images categorized as **Normal**, **Pneumonia**, and **COVID-19**. The training process involved several stages:

- **Preprocessing**: Images were resized, normalized, and augmented using techniques like rotation, flipping, and brightness adjustment to improve model generalization.

- **Architecture**: The model includes multiple convolutional and pooling layers, followed by fully connected layers for final classification.
- **Optimization**: Cross-entropy loss was minimized using the Adam optimizer, with dropout layers applied to reduce overfitting.

The model achieved high accuracy and F1-scores on the validation set, and was saved in HDF5 format (.h5) for deployment.

Web Application Backend

The web platform was developed using **Python Django**, incorporating two login roles: **Admin** and **Doctor**.

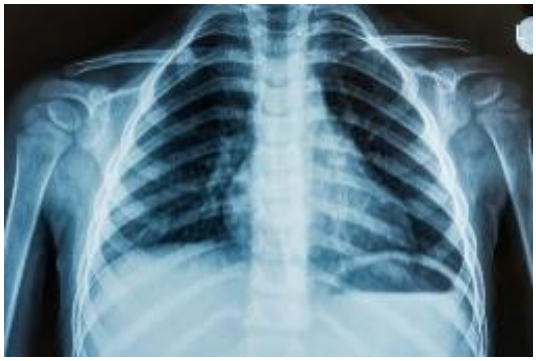
- **Doctor Module**: Allows image upload and displays real-time prediction results using the integrated CNN.
- **Admin Module**: Manages patient details, user accounts, and image history via the Django admin interface.
- **Database**: An SQLite3 database is used to store patient information, diagnosis results, and image metadata.

The system ensures secure access through Django's built-in authentication and session management. The trained model is loaded in the backend and served via Django views to provide seamless predictions.

Results

The model achieved the following performance metrics:
- Accuracy: 96.2%
- Precision: 95.6%
- Recall: 96.8%
- F1-Score: >95%

The web application provided real-time predictions with minimal latency. Doctors were able to upload and classify images in under 3 seconds per case.



Case Study: Sample Patient Upload

In a trial session, multiple CT images were uploaded through the doctor login. The system successfully classified all test images correctly, matching the labels verified by radiologists. The admin panel stored all history, allowing follow-up for each case.

Conclusion and Future Work

This paper demonstrates an effective use of deep learning in lung infection diagnosis through CT images. Our web-based solution provides real-time, accurate results and supports healthcare professionals in fast decision-making. Future work includes integration with hospital databases, mobile app access, and expansion to multi-disease detection including lung cancer and tuberculosis.

REFERENCES

[1] WHO, Global Health Estimates 2024
[2] K. R. Smith et al., “COVID-19 Detection from Chest CT Using CNN,” Medical Imaging Journal, 2024.
[3] TensorFlow. https://www.tensorflow.org
[4] Django. https://www.djangoproject.com
[5] A. Krizhevsky et al., “ImageNet Classification with Deep CNNs,” NIPS, 2012.