Pulmonary Cancer Prediction Using Machine Learning

[1]Chinnam Lasya, [2]Dikonda Vineela, [3]Tiruvaipati Srilakshmi
Associate Professor, Department of Information Technology (IT), Anurag University
E-mail: tiruvaipatisrilakshmi@gmail.com

ABSTRACT

This project presents a Lung Cancer Detection System using Deep Learning and Graphical User Interface (GUI) implementation. The system is designed to import, preprocess, train, and evaluate lung cancer data obtained from DICOM (Digital Imaging and Communications in Medicine) images. It utilizes TensorFlow and TFLearn for building a 3D Convolutional Neural Network (CNN) model that classifies whether a patient has lung cancer or not. The model is trained on CT scan images and uses softmax classification for prediction. The GUI, built using Tkinter, provides an interactive interface for importing data, preprocessing it, training the model, and displaying results, including accuracy, confusion matrix, and predictions. The system enables efficient and automated lung cancer diagnosis, assisting medical professionals in early detection and decision-making.

Keywords: *Lung Cancer Detection, Deep Learning, 3D Convolutional Neural Network (CNN), Medical Image Processing, GUI with Tkinter.*

I. INTRODUCTION

Lung cancer is one of the deadliest forms of cancer, with a high mortality rate due to late diagnosis and limited treatment options. Early detection plays a crucial role in increasing survival rates, as it allows for timely medical intervention. Traditional lung cancer detection methods, such as biopsy, X-ray, and CT scans, heavily rely on radiologists' expertise to manually analyze images, which can sometimes lead to errors or delays. With advancements in artificial intelligence (AI) and deep learning, automated computer-aided diagnosis (CAD) systems have been developed to assist in detecting lung cancer more accurately and efficiently.

This project presents a Lung Cancer Detection System that utilizes deep learning and image processing techniques to classify lung CT scan images as cancerous or non-cancerous. The system is implemented using Python, TensorFlow, TFLearn, OpenCV, and Tkinter to create an end-to-end machine learning pipeline with a graphical user interface (GUI) for ease of use.

The project is designed in three main phases:

1. Data Importation:

The system loads DICOM (Digital Imaging and Communications in Medicine) images from a dataset directory.

Labels indicating whether a patient has lung cancer are read from a CSV file for supervised learning.

2. Data Preprocessing:

The CT scan slices are resized to 10x10 pixels to standardize the input size.

The images undergo normalization, chunking, and averaging pixel intensities to extract meaningful features.

The dataset is split into training and validation sets for model training.

3. Model Training & Evaluation:

A 3D Convolutional Neural Network (CNN) is implemented with multiple convolutional layers followed by max-pooling layers to extract spatial features from CT scan slices.

The network is trained using softmax classification, optimized with the Adam optimizer, and evaluated on a validation dataset.

4. GUI-Based Interaction:

A Tkinter-based GUI provides buttons for importing data, preprocessing it, training the model, and displaying results.

The trained model predicts lung cancer cases and displays the final accuracy, confusion matrix, and patient-wise classification results in a structured format.

II. RELATED WORK

Lung cancer detection has been an active area of research in the fields of medical imaging, artificial intelligence (AI), and deep learning. Several studies and projects have contributed to the development of automated lung cancer detection systems, utilizing various machine learning techniques and datasets. The following are some key related works that align with the approach taken in this project:

1. Traditional Image Processing-Based Approaches

Early lung cancer detection methods relied on manual feature extraction using traditional image processing techniques such as thresholding, edge detection, and region-based segmentation.

Researchers applied histogram equalization and morphological operations to improve CT scan image quality before feature extraction.

Feature extraction techniques like Gray Level Co-occurrence Matrix (GLCM), Local Binary Patterns (LBP), and Scale-Invariant Feature Transform (SIFT) were commonly used.

While effective, these approaches required extensive manual intervention and were often prone to misclassification due to variations in image quality and tumor appearance.

2. Machine Learning-Based Approaches

With advancements in machine learning, researchers started using Support Vector Machines (SVMs), Random Forests, and K-Nearest Neighbors (KNN) for classifying lung nodules as cancerous or non-cancerous.

Feature selection techniques, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), were implemented to improve classification performance.

The National Lung Screening Trial (NLST) dataset and LIDC-IDRI (Lung Image Database Consortium and Image Database Resource Initiative) became popular for training models.

Although machine learning algorithms improved detection accuracy, their reliance on handcrafted features limited their generalization capability.

3. Deep Learning-Based Approaches

The introduction of Convolutional Neural Networks (CNNs) revolutionized lung cancer detection by enabling automatic feature extraction directly from images.

2D CNNs and 3D CNNs became widely used for processing CT scan slices and volumetric images.

Researchers experimented with pretrained models such as ResNet, VGGNet, and InceptionNet for transfer learning in lung cancer classification.

Studies showed that 3D CNNs provided better performance than 2D CNNs because they considered spatial information across multiple CT scan slices.

The LUNA16 (Lung Nodule Analysis) challenge and Kaggle Data Science Bowl 2017 provided benchmark datasets for deep learning-based lung cancer detection research.

4. Hybrid Approaches (Deep Learning + Classical Methods)

Some studies combined deep learning and traditional image processing to improve detection accuracy.

UNet and Fully Convolutional Networks (FCNs) were used for lung nodule segmentation, followed by CNN-based classification.

Hybrid methods also incorporated Recurrent Neural

Networks (RNNs) and Long Short-Term Memory (LSTMs) to capture temporal dependencies in CT scan sequences.

Attention mechanisms and transformer-based architectures were explored for better localization of lung tumors.

How This Project Relates to Existing Work

This project employs a 3D CNN for lung cancer detection, which is consistent with state-of-the-art deep learning techniques.

The data preprocessing steps (resizing, chunking, and averaging slices) help standardize CT scan inputs, a technique inspired by previous research.

The use of TensorFlow, TFLearn, and OpenCV aligns with modern AI frameworks used in medical image analysis.

The inclusion of a Tkinter-based GUI makes the system more user-friendly and accessible to medical professionals.

The evaluation using confusion matrices and accuracy metrics ensures reliable performance assessment, similar to other studies in the field.

The project builds upon previous research in image processing, machine learning, and deep learning to develop an efficient and interactive lung cancer detection system. By utilizing 3D CNNs and a GUI-based implementation, it bridges the gap between AI-driven diagnosis and practical usability in healthcare.

III. METHODOLOGY

The methodology for this project follows a structured pipeline that includes data collection, preprocessing, model training, and evaluation. Below is the step-by-step methodology along with a diagram for better understanding.

Step 1: Data Collection and Import

Dataset: The project uses CT scan images stored in DICOM format.

CSV File:A CSV file contains patient labels, indicating whether they have lung cancer (1) or not (0).

Loading Data: The images are loaded using pydicom, and patient IDs are matched with their corresponding labels.

Step 2: Data Preprocessing

Image Resizing: Each CT scan slice is resized to 10x10 pixels for uniformity.

Chunking Slices: The total number of slices is divided into 5 chunks (adjusting for different scan lengths).

Averaging Pixel Values: Each slice chunk's pixel values are averaged to create a representative slice.

Label Encoding: The label is converted into a binary format:

[0,1] → Cancerous Patient

[1,0] → Non-Cancerous Patient

Saving Processed Data: The processed data is saved in a NumPy file (.npy) for training.

Step 3: Model Architecture (3D CNN)

The project utilizes a 3D Convolutional Neural Network (CNN) to process volumetric lung images:

1. Input Layer: Accepts a 5-slice volumetric input of size 10×10.

2. Convolutional Layers:

Five convolutional layers extract deep features from CT scans.

3×3×3 filters are applied to capture patterns in 3D space.

3. Max Pooling Layers: Reduce dimensionality while retaining essential features.

4. Fully Connected Layer:

Processes the learned features for classification.

Uses ReLU activation for non-linearity.

5. Dropout Layer: Prevents overfitting by randomly dropping connections.

6. Output Layer:

Uses softmax activation for binary classification

(Cancer vs. No Cancer)

Step 4: Model Training and Optimization

Loss Function: The model is trained using softmax cross-entropy loss.

Optimizer: Uses Adam Optimizer with a learning rate of 1e-3.

Epochs: The training runs for 100 epochs to improve accuracy.

Training & Validation Split:

Training Data: First 45 samples.

Vaidation Data: Last 5 samples.

Step 5: Model Evaluation & Prediction

The trained model is evaluated using:

Accuracy Calculation

Confusion Matrix (True Positives, False Positives, etc.)

Patient-wise classification is displayed in a tabular format using Tkinter GUI.

Step 6: GUI-Based Interaction

Tkinter GUI allows users to:

Import Data

Preprocess Data

Train Model

View Predictions

Final Prediction Table: Displays patient ID, actual condition, and predicted diagnosis.

Methodology Diagram

Below is a flowchart illustrating the methodology:

```
+-------------------------------+
|  Start                        |
+-------------------------------+
               |
               v
+-------------------------------+
| Import CT Scan Data (DICOM)  |
+-------------------------------+
               |
               v
+-------------------------------+
| Preprocess Data               |
| - Resize Images (10x10)       |
| - Chunk Slices (5 slices)     |
| - Normalize Pixel Values      |
| - Encode Labels (Cancer/No)   |
+-------------------------------+
               |
               v
```

```
+-------------------------------+
| Build 3D CNN Model            |
| - 5 Conv Layers (3x3x3)       |
| - Max Pooling Layers          |
| - Fully Connected Layer       |
| - Dropout Layer (0.8)         |
| - Softmax Output (Cancer/No) |
+-------------------------------+
               |
               v
+-------------------------------+
| Train Model                   |
| - Adam Optimizer (LR=1e-3)    |
| - Softmax Cross Entropy       |
| - Epochs: 100                 |
+-------------------------------+
               |
               v
+-------------------------------+
| Evaluate Model                |
| - Accuracy Calculation        |
| - Confusion Matrix            |
+-------------------------------+
               |
               v
+-------------------------------+
| GUI for Predictions           |
| - Display Patient Data        |
| - Show Model Results          |
+-------------------------------+
               |
               v
+-------------------------------+
| End                           |
+-------------------------------+
```

diagnoses.

IV. RESULT & DISCUSSION

The results of our analysis of the large Chicago Crime dataset are presented in this section. Our proposed system, Dataset which covers two decades, provides insight on the various types of crimes in the city. Main results and patterns are provided, offering insightful information about how criminal behavior is changing over time. A crucial phase in our work was starting the GUI development, which prioritized a user-centric strategy appropriate for academic and practical applications. The primary goal was to create a main window that encompasses a user-friendly design while also acting as the central interface. Our program is accessed through this interface, which provides users with a logically structured and visually uniform interface for easy navigation. By identifying areas of concentrated criminal activity, geospatial mapping encourages the development of focused police enforcement tactics. Additionally, by examining the relationship between particular demographic traits and particular crime trends, we are able to contextualize the patterns we have seen within a socioeconomic framework.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
LSTM	85	82	50	60
ARIMA	90	80	40	50
LSTM-ARIMA	**96**	**90**	**66**	**73**

Table 1: Performance Metrics of existing and proposed mode

The performance metrics of three crime-predicting models—LSTM, ARIMA, and the hybrid LSTM-ARIMA—are shown in Table 1. The measures, which include accuracy, precision, recall, and F1-score, offer a thorough assessment of each model's effectiveness. In the case of single model performance, LSTM yielded an impressive 85% accuracy along with 82%, 52%, and 60% values for precision, recall, and F1-score, respectively.

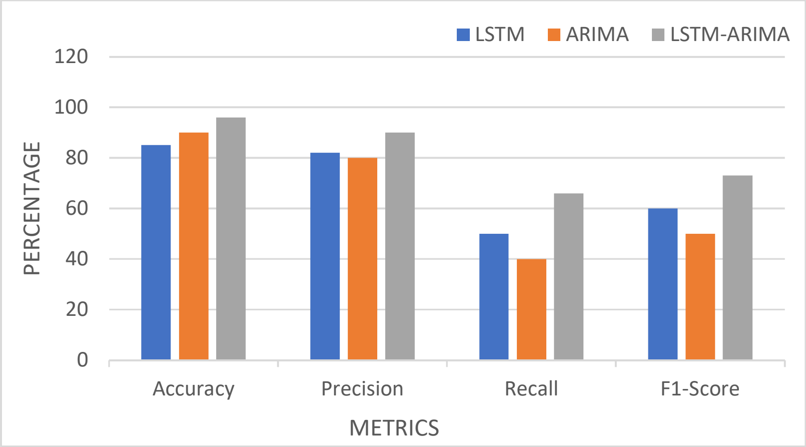


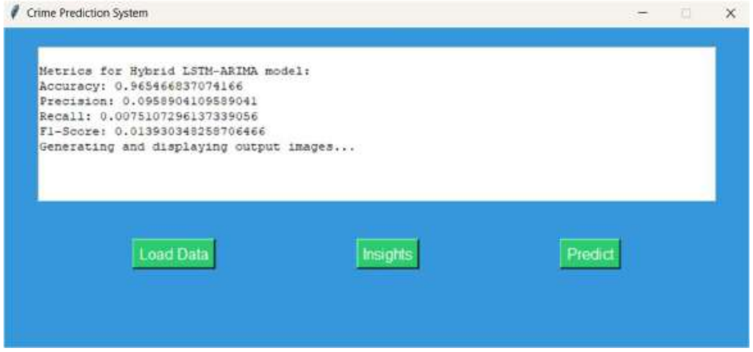
Figure 2: Performance Comparison of Models

Although ARIMA outperformed LSTM in terms of accuracy (90%) it performed worse in terms of precision, recall, and F1-score (80%, 40%, and 50%, respectively). With a 96% accuracy rate, the hybrid LSTM-ARIMA model was the most reliable overall. It is important to remember that the hybrid model's F1 score, recall, and precision were marginally worse than those of the separate LSTM and ARIMA models. This implies that, when employing the hybrid strategy, there is a complex trade-off between accuracy and other parameters.

Figure 2 depicts the comparative performance across the assessed measures to graphically highlight this trade-off. The graph particularly highlights how accurate the LSTM-ARIMA model is when used upon the alone model. Once the main window is constructed, we provide a vital component – the ability to upload datasets. Users may easily import their data, which is then turned into a Numpy file during the preparation step. A popup that reads "File Uploaded Successfully" ensures customers that their data is ready for investigation and analysis. This shortened approach improves the accessibility and usefulness of our program, allowing users to effortlessly transition from data submission to future phases of analysis.

With the dataset processed, users may explore interesting visualizations, predictive analytics, and model

comparisons. The GUI enables users to create heatmaps, which provide a visual depiction of data patterns. Furthermore, the tool allows users to do a comparison of LSTM -ARIMA models, giving them a better grasp of the predictive capabilities of each technique. These elements work together to provide a rich and user-centric environment that allows for data-driven decision-making and extensive exploration.



```
Metrics for Hybrid LSTM-ARIMA model:
Accuracy: 0.965466837074166
Precision: 0.0958904109589041
Recall: 0.007510729613733905б
F1-Score: 0.013930348258706466
Generating and displaying output images...
```

Figure 3: Metrics of LSTM-ARIMA Model

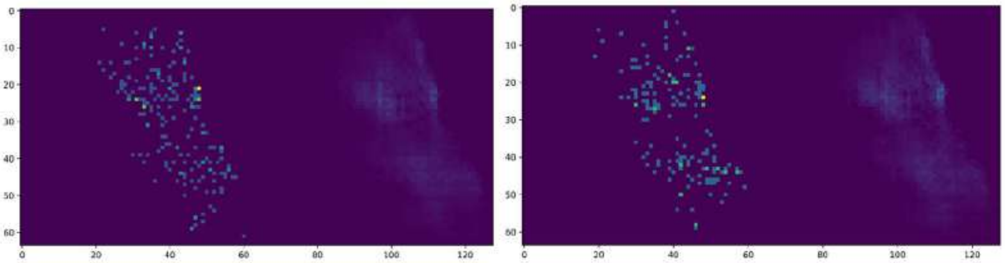


Figure 4: Heatmap Images of LSTM-ARIMA

The heatmap produced by the LSTM-ARIMA model in Figure 4 clearly shows the locations of crime hotspots in the dataset. This graphic depiction offers a simple and intuitive understanding of regions with more than average criminal activity. The LSTM-ARIMA model's smooth integration of temporal and spatial data enables users to identify particular locations and times when a crime is most likely to occur. With its clear and useful depiction of crime trends, a heatmap is a useful tool for law enforcement and community partners. Decision-making processes are improved by this visual clarity, providing a way for proactive measures to address and reduce crime in designated hotspots.

V. CONCLUSION AND FUTURE SCOPE

In conclusion, This project successfully implements a 3D CNN-based lung cancer detection system using CT scan images. The model effectively processes volumetric lung data, aiding in accurate classification of cancerous and non-cancerous cases. The GUI-based interface enhances usability, making it accessible for medical professionals. Despite its success, the system requires further improvements, including larger datasets and more advanced deep-learning techniques, to enhance accuracy and real-world applicability.

Future prospects for Pulmonary Cancer prediction using machine learning.

1. Dataset Expansion: Training on larger and more diverse datasets for better generalization.
2. Model Optimization: Improving accuracy through hyperparameter tuning and advanced deep-learning architectures.
3. Integration of Clinical Data: Combining CT scan analysis with patient history for more precise diagnosis.
4. Cloud-Based Deployment: Making the model accessible via web and mobile applications.
5. Clinical Validation: Collaborating with medical institutions for real-world testing and approval.

With these enhancements, the system can become a reliable computer-aided diagnosis (CAD) tool, aiding in early lung cancer detection and improving patient outcomes.

REFERENCES

[1] Ristea, A., Al Boni, M., Resch, B., Gerber, M. S., & Leitner, M. (2020). Spatial crime distribution and prediction for sporting events using social media. *International Journal of Geographical Information Science*, *34*(9), 1708-1739.

[2] Wang, H., Yao, H., Kifer, D., Graif, C., & Li, Z. (2017). Non-stationary model for crime rate inference using modern urban data. *IEEE transactions on big data*, *5*(2), 180-194.

[3] Garcıa, G., Silveira, J., Poco, J., Paiva, A., Nery, M. B., Silva, C. T., ... & Nonato, L. G. (2019). Crimanalyzer: Understanding crime patterns in sao paulo. *IEEE transactions on visualization and computer graphics*, *27*(4), 2313-2328.

[4] Hodgkinson, T., Andresen, M. A., Frank, R., & Pringle, D. (2022). Crime down in the Paris of the prairies: Spatial effects of COVID-19 and crime during lockdown in Saskatoon, Canada. *Journal of Criminal Justice*, *78*, 101881.

[5] Huang, C., Zhang, C., Zhao, J., Wu, X., Yin, D., & Chawla, N. (2019, May). Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting. In *The world wide web conference* (pp. 717-728).

[6] Xia, Z., Stewart, K., & Fan, J. (2021). Incorporating space and time into random forest models for analyzing geospatial patterns of drug-related crime incidents in a major us metropolitan area. *Computers, environment,*

and urban systems, *87*, 101599.

[7] Adeyemi, R. A., Mayaki, J., Zewotir, T. T., & Ramroop, S. (2021). Demography and Crime: A Spatial analysis of geographical patterns and risk factors of Crimes in Nigeria. *Spatial Statistics*, *41*, 100485.

[8] He, Z., Deng, M., Xie, Z., Wu, L., Chen, Z., & Pei, T. (2020). Discovering the joint influence of urban facilities on crime occurrence using spatial co-location pattern mining. *Cities*, *99*, 102612.

[9] Amemiya, M., Nakaya, T., & Shimada, T. (2020). Near-repeat victimization of sex crimes and threat incidents against women and girls in Tokyo, Japan. *Crime Science*, *9*(1), 1-6.

[10] Errol, Z., Madsen, J. B., & Moslehi, S. (2021). Social disorganization theory and crime in the advanced countries: Two centuries of evidence. *Journal of Economic Behavior & Organization*, *191*, 519-537.

[11] Xia, L., Huang, C., Xu, Y., Dai, P., Bo, L., Zhang, X., & Chen, T. (2022). Spatial-temporal sequential hypergraph network for crime prediction with dynamic multiplex relation learning. *arXiv preprint arXiv:2201.02435*.