Lung Nodule Detection From CT Scans Using CNN: A Comparative Analysis of CNN Architectures With Hyperparameter Tuning

Arvin Bahreini, Sarah Metry, Jewelianna Langston
Department of Electrical Engineering
University of Rochester
Rochester, NY, United States
abahrein@ur.rochester.edu
smetry@ur.rochester.edu
jlangsto@u.rochester.edu

***Abstract*—Early detection of lung nodules in computed tomography (CT) scans is crucial for improving lung cancer survival rates. This project presents a comparative analysis of a custom 2D convolutional neural network (CNN) model along with four state-of-the-art pretrained CNN architectures—ResNet18, EfficientNet-B0, ResNet50, and DenseNet121—for lung nodule detection using the LUNA16 dataset. We implement a comprehensive preprocessing pipeline for lung segmentation and image normalization, followed by model training. For pretrained models transfer learning is utilized from ImageNet weights. Furthermore, we employ Particle Swarm Optimization (PSO) for hyperparameter tuning to enhance model performance. Our experimental results demonstrate that ResNet50, when optimized with PSO, achieves the highest accuracy of 96.32%, outperforming other architectures. This study provides insights into effective CNN architecture selection and the value of PSO-based optimization for medical image analysis tasks.**

***Index Terms*—deep learning, convolutional neural networks, lung nodule detection, particle swarm optimization, hyperparameter tuning, computer-aided diagnosis, computed tomography**

I. Introduction

Lung cancer remains one of the leading causes of cancer-related deaths worldwide, with approximately 1.8 million deaths annually. Early detection of malignant lung nodules significantly improves the five-year survival rate from 15% to over 50% [1]. Computed tomography (CT) scans are the primary imaging modality for detecting lung nodules, but manual interpretation is time-consuming and prone to inter-observer variability. Also, these automated deep learning systems can serve as a valuable second pair of eyes for radiologists, helping experts detect subtle lung nodules that might otherwise be missed during routine screening, potentially improving early diagnosis while reducing reader fatigue in high-volume clinical settings.

Deep learning approaches, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable success in medical image analysis tasks, including lung nodule detection. While various CNN architectures have shown promising results individually, a systematic comparison across different architectures is essential for identifying optimal solutions.

In this study, we address three key challenges in CNN-based lung nodule detection:

1) Analyzing the performance of a custom deep CNN model architecture based on Abraham et. al design [2].
2) Identifying the most effective CNN architecture among popular pretrained models (ResNet18, EfficientNet-B0, ResNet50, and DenseNet121)
3) Optimizing hyperparameters to enhance model performance using grid search and Particle Swarm Optimization (PSO)

We utilize the LUNA16 (Lung Nodule Analysis 2016) dataset, which contains annotated CT scans from the Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI). Our approach includes a preprocessing pipeline for image normalization, lung segmentation, saving, 3D scan sorting, and slice extraction, followed by model training and evaluation using standardized metrics.

The key contributions of this project include:

- Implementation of the custom CNN model, training and testing it along with hyperparameter tuning.
- A systematic comparison of four state-of-the-art pretrained CNN architectures for lung nodule detection
- Implementation of PSO for hyperparameter optimization of CNN models
- Quantitative analysis of performance improvements achieved through PSO
- Recommendations for optimal CNN architecture selection based on empirical results

II. CNN Architectures

We implemented and evaluated the custom CNN architecture as well as four state-of-the-art CNN architectures pretrained on the ImageNet dataset. The pretrained architectures

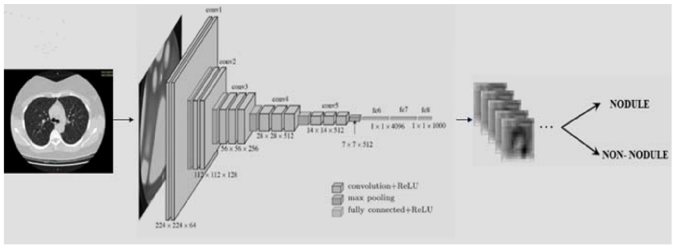


Fig. 1. Overview of the CNN architecture for lung nodule detection. The model processes CT scan slices through multiple convolutional layers, extracting hierarchical features before classification into normal or nodule-containing categories [2].

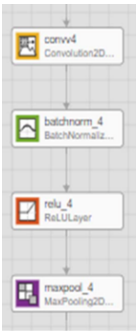


Fig. 2. A single CNN block design includes a convolutional layer, batch normalization layer, ReLU and maxpool. [2].

were selected based on their proven effectiveness in medical image analysis tasks and varying architectural properties [1]:

A. Proposed CNN Architecture

The custom CNN model proposed consists of five convolutional blocks followed by fully connected layers, drawing from the design proposed by Abraham et. al. Fig. 1 illustrates the model architecture. Each block consists of convolution, ReLU activation, and max pooling layers as shown in Fig. 2. Dropout layer is added after each fully-connected layer for regularization. The final fully-connected layer classifies the input slices into nodule and non-nodule classes [2].

B. ResNet18

ResNet (Residual Network) introduced skip connections to address the vanishing gradient problem in deep networks. ResNet18 consists of 18 layers with identity mapping connections that enable the training of deeper networks. Its relatively lightweight structure (11.7 million parameters) makes it efficient while maintaining good performance [4].

C. EfficientNet-B0

EfficientNet uses compound scaling to balance network depth, width, and resolution. The B0 baseline model (5.3 million parameters) employs mobile inverted bottleneck convolution (MBConv) blocks with squeeze-and-excitation optimization. Its design prioritizes efficiency while maintaining competitive accuracy [5].

D. ResNet50

ResNet50 extends the residual learning framework with a deeper architecture of 50 layers (25.6 million parameters). It utilizes bottleneck blocks to reduce computational complexity while preserving representational capacity. The deeper network enables learning more complex features that can be beneficial for detecting subtle nodule patterns [4].

E. DenseNet121

DenseNet employs dense connections where each layer receives inputs from all preceding layers. DenseNet121 has 121 layers with a growth rate of 32 (8.0 million parameters). This architecture facilitates feature reuse, strengthens feature propagation, and reduces the number of parameters, making it well-suited for medical imaging tasks.

As shown in Fig. 1, for all architectures, we replaced the final classification layer to output two classes (non-nodule and nodule). The models were initialized with pre-trained ImageNet weights, with the final layer trained from scratch.

III. Training and Evaluation

A. Dataset Preparation

We utilized the LUNA16 dataset, which contains 888 3D CT scans with 1186 annotated nodules. Among the CT scans, there were 601 scans with at least one nodule and 287 without nodules. The supporting documents used consisted of 10 ZIP files containing 3D CT scans, a CSV file with annotated nodule coordinates, and a directory containing lungs segmentations. The preprocessing pipeline consisted of the following steps:

1) **Normalization**: Images were normalized to the range [0, 255] using windowing (center: 1200 HU, width: 1800 HU).
2) **Lung Segmentation**: The lung segmentation masks provided by LUNA16 were applied to the corresponding CT slices to isolate lung regions through element-wise multiplication.
3) **Saving Format**: All processed slices were saved as PNG images and moved to a folder labeled with their respective scan's name (seriesuid).
4) **Sorting**: The 3D scans were sorted into training, validation, and testing subsets in a 60:20:20 ratio.
5) **Slice Extraction**: For each detected nodule, six slices were extracted. The location of extraction was determined by the nodule's Z-coordinate. To maintain class balance, an equal number of slices were randomly sampled from scans without nodules.

The final dataset contained 7950 training images, 2912 validation images, and 2510 testing images (6686 with nodule and 6686 non-nodule).

B. Training Strategy

We implemented the following training configuration:

1) **Custom CNN Model**
   - **Hyperparameter Tuning**: Ray Tune Library using grid search for [16, 32, 64] batch sizes, [0.01, 0.001,

TABLE I
RAY TUNE HYPERPARAMETER TUNING SAMPLE OUTPUT

	Batch	lr	wd	Epoch	Loss	f1	Accuracy
0	16	0.0001	0.0001	20	0.455	0.778	0.779
1	32	0.0001	0.0001	4	0.562	0.675	0.676
2	64	0.0001	0.0001	1	0.572	0.638	0.668
7	32	0.01	0.0001	1	0.694	0.333	0.5
8	64	0.01	0.0001	4	0.667	0.648	0.668
9	16	0.0001	0.001	20	0.484	0.755	0.758

TABLE II
RESULTS OF LAST FIVE EPOCHS OF TRAINING THE CUSTOM CNN ARCHITECTURE

Epoch	Loss	Accuracy	Precision	F1-Score
21	0.4663	77.3%	0.78	0.7732
22	0.4879	75.5%	0.7738	0.7550
23	0.5132	74.1%	0.7644	0.7409
24	0.4369	79.1%	0.7924	0.7911
25	0.4427	78.28%	0.7830	0.7828

0.0001] learning rates and [0.01, 0.001, 0.0001] weight decays for 20 epochs. The tuning leveraged the ASHA (Asynchronous Successive Halving Algorithm) scheduler to efficiently allocate resources and prune suboptimal trials. The best configuration was identified using the weighted F1-score as the primary metric. Table I shows results for some combinations including loss, f1 and accuracy. The first combination showed the best performance for our model.

- **Loss Function**: Cross-entropy loss
- **Optimizer**: Adam with the best hyperparameters after tuning: batch size of 16, learning rate of 0.0001 and weight decay of 0.0001
- **Learning Rate Scheduling**: ReduceLROnPlateau with monitoring of validation loss
- **Augmentation**:
  - Random horizontal flips and rotations (±15°)
  - Color Jitter
  - Gaussian Blur
- **Image Resize**: The images are resized to 124*124 that can be compatible with the structure of the pretrained models
- **Batch Size**: 16 (based on hyperparameter tuning)
- **Training Duration**: 25 epochs

2) **Pretrained Models**

- **Loss Function**: Cross-entropy loss
- **Optimizer**: AdamW with initial learning rate of 0.0001 and weight decay of 0.0001
- **Learning Rate Scheduling**: ReduceLROnPlateau with monitoring of validation F1-score
- **Augmentation**: Random horizontal flips and rotations (±10°)
- **Image Resize**: The images are resized to 124*124 that can be compatible with the structure of the pretrained models
- **Batch Size**: 64
- **Training Duration**: 25 epochs

All models were trained on a GPU with CUDA acceleration. We also used PyTorch as the deep learning framework.

C. Evaluation Metrics

Model performance was evaluated using the following metrics:

- Loss
- Accuracy
- Precision
- F1-score
- Area Under the ROC Curve (AUC)
- Model Size
- Train-Time for one Epoch

IV. HYPERPARAMETER TUNING USING PSO

Particle Swarm Optimization (PSO) is a population-based optimization technique inspired by social behavior in bird flocking. We employed PSO to optimize the following hyperparameters: [7]

- Learning rate (range: 1e-6 to 3e-3)
- Weight decay (range: 0 to 5e-4)
- Batch size (range: 16 to 128)
- Optimizer parameters (β_1: 0.5 to 0.95, β_2: 0.8 to 0.9999, ϵ: 1e-9 to 1e-6)

Our PSO implementation used the following configuration, which are standard parameters that control particle behavior in the optimization process [7]:

- Number of particles: 3
- Number of dimensions: 6 (one for each hyperparameter)
- Cognitive coefficient (c_1): 0.8
- Social coefficient (c_2): 0.8
- Inertia weight (w): 0.6
- Number of iterations: 3
- Fitness function: 1 - validation F1-score

To efficiently evaluate the particles, We reduced the training epochs to 4 during the optimization phase. After identifying the optimal hyperparameters, the best-performing model was retained (ResNet50) with the full 25 epochs.

V. RESULTS

A. Performance of the Custom CNN Model with Hyperparameter Tuning

Performance of the custom CNN model was evaluated after training for 25 epochs using the best hyperparameters achieved. Table II shows the validation results of the last five epochs of training on our validation dataset. The training loss is decreasing through the 25 epochs while the validation loss saturated towards the end with some fluctuations. This suggests that the model is still able to learn, but it was starting to overfit on our dataset.

TABLE III
PERFORMANCE COMPARISON OF CNN ARCHITECTURES

Model	Loss	Accuracy	Precision	F1-Score
EfficientNet-B0	0.2538	92.45%	0.9257	0.9249
ResNet18	0.2495	92.56%	0.9255	0.9249
DenseNet121	0.2342	93.07%	0.9311	0.9304
ResNet50	0.2018	94.21%	0.9428	0.9423
ResNet50 with PSO	0.1393	96.32%	0.9634	0.9633

B. Performance Comparison of CNN Architectures

Table III presents the performance of the four CNN architectures on the validation set before PSO. Also, it shows the performance of ResNet50 (which was the best model) after applying PSO.

ResNet50 achieved the highest performance across all metrics, with an accuracy 94.21%, followed by DenseNet121 (93.07%), EfficientNet-B0 (92.56%), and ResNet18 (92.45%).

C. Confusion Matrices

Fig. 3 presents the confusion matrices for all model architectures. These visualizations provide detailed insights into each model's classification performance:

The ResNet50 model (Fig. 3a) achieved a balanced performance with 1185 true negatives and 639 true positives, but still had 67 false positives and 45 false negatives. After PSO optimization (Fig. 3b), the ResNet50 model substantially improved, with increased true negatives (1237) and true positives (670), while reducing false positives to just 15 and false negatives to 14. This represents a significant enhancement in both precision and recall.

ResNet18 (Fig. 3c) showed good true negative (1197) and true positive (605) counts, but had higher misclassification rates with 55 false positives and 79 false negatives, indicating lower recall for nodule detection. DenseNet121 (Fig. 3d) achieved 1178 true negatives and 628 true positives, with 74 false positives and 56 false negatives. EfficientNet (Fig. 3e) had strong true negative performance (1203) but showed the weakest nodule detection capability among the models with 589 true positives and 95 false negatives, suggesting difficulties in identifying nodules. (Fig. 3f) shows the confusion matrix of the custom CNN.

D. PSO Hyperparameter Optimization

The PSO optimization converged to the following hyperparameter values for ResNet50:

- Learning rate: 4.44e-4
- Weight decay: 1.66e-4
- Batch size: 44
- β_1: 0.712
- β_2: 0.953
- ϵ: 3.9e-7

After retraining ResNet50 with these optimized hyperparameters for 25 epochs, we observed significant improvement in performance, achieving an accuracy of 96.3%. Also, the loss function was reduced by 31%.

E. Training Dynamics

Fig. 4 shows the accuracy curves of all models during training, illustrating how training and validation accuracy improved over epochs:

Several key observations can be made from these accuracy curves:

- All models showed rapid improvement in the first 10 epochs, with most of the learning accomplished in this period
- The PSO-optimized ResNet50 (Fig. 4a) exhibited both the highest final validation accuracy (approximately 0.96) and more stable convergence compared to its non-optimized version
- ResNet50 (Fig. 4b) showed good performance but with more fluctuations in validation accuracy
- ResNet18 (Fig. 4c) displayed the most unstable validation accuracy curve with significant fluctuations, suggesting potential sensitivity to batch composition
- DenseNet121 (Fig. 4d) had a stable learning curve with a gradual improvement in validation accuracy
- EfficientNet (Fig. 4e) demonstrated a smooth learning curve but with a persistent gap between training and validation accuracy, suggesting some degree of overfitting
- Custom CNN Model (Fig. 4f) shows a good learning curve while validation accuracy fluctuated slightly.

Notably, the PSO-optimized ResNet50 model showed the smallest gap between training and validation accuracy curves in later epochs, indicating better generalization capability.

VI. CONCLUSION AND FUTURE WORK

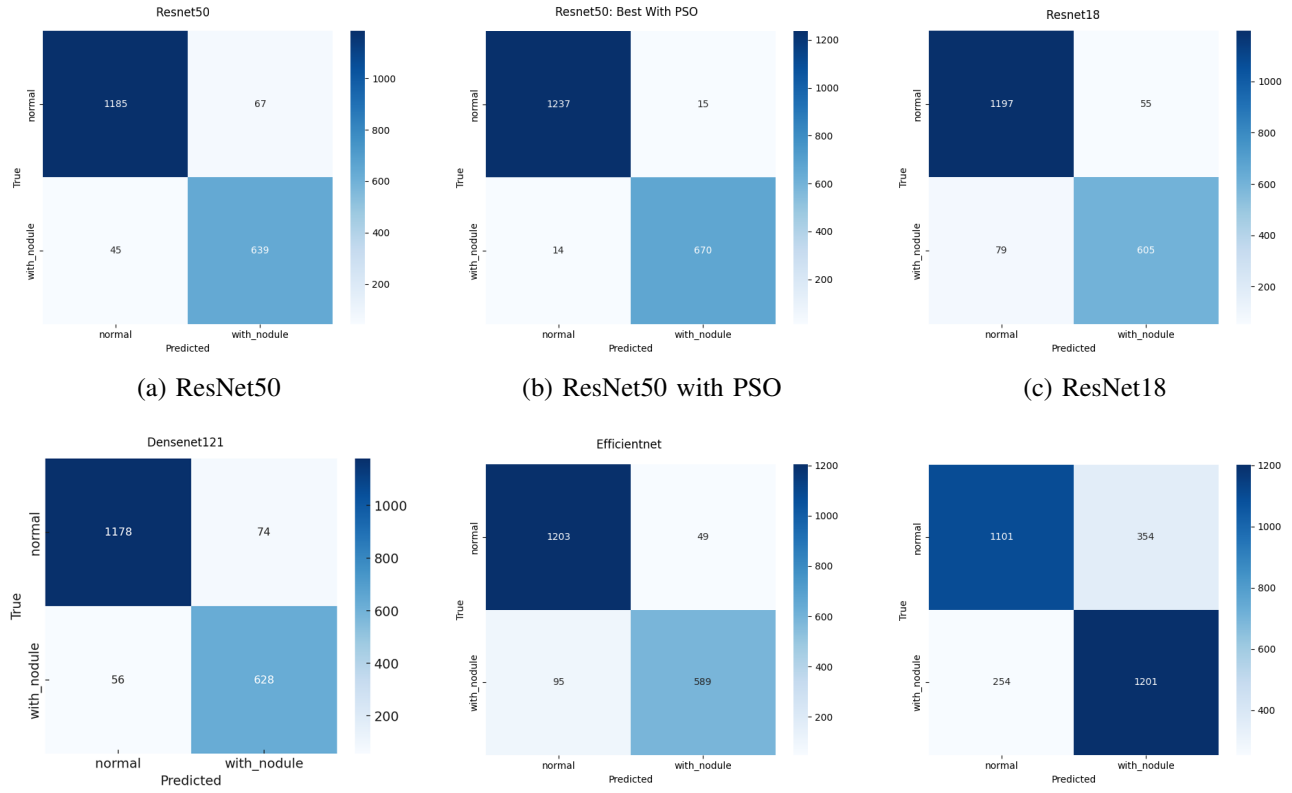
In this project, we presented a comparative analysis of different CNN architectures for lung nodule detection in CT scans, enhanced by grid search and PSO-based hyperparameter tuning. Our findings indicate that ResNet50, when optimized with PSO, achieves the highest performance with an accuracy of 96.32% and an F1-score of 0.9633. The results demonstrate that deeper architectures like ResNet50 can better capture the complex patterns associated with lung nodules, while the PSO optimization significantly improves model convergence and overall performance. In addition, our results demonstrate that pre-trained models outperform our custom CNN model which proves transfer learning effectiveness.

The key observations from our study include:

- Deeper architectures (ResNet50, DenseNet121) outperform shallower ones for lung nodule detection
- PSO effectively optimizes learning dynamics, leading to improved model performance
- Batch size and learning rate had the most significant impact on model performance during optimization
- Transfer learning from ImageNet provided a solid foundation for all architectures

Future work directions include:

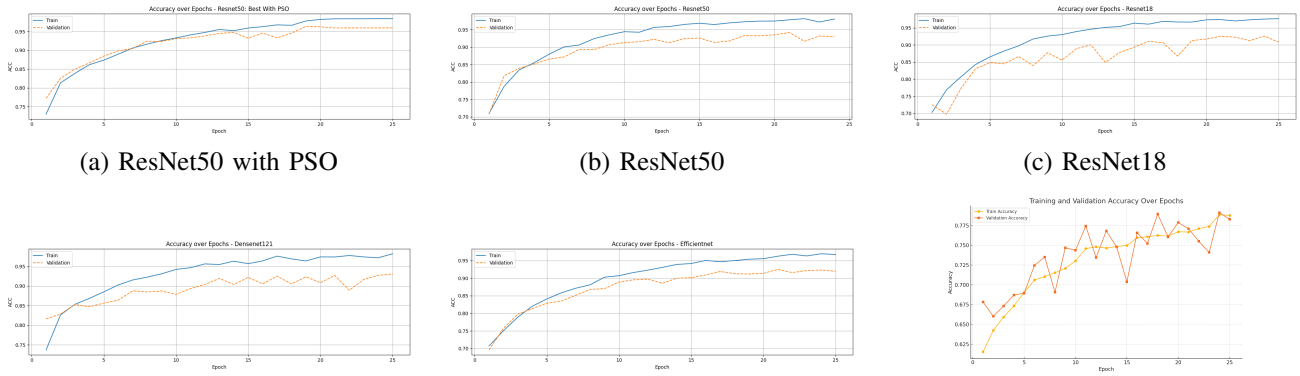
1) We can use advanced data augmentation instead of randomly flipping or rotating. For instance, implementing DDGAN (Diffusion Denoising Generative Adversarial



(a) ResNet50 (b) ResNet50 with PSO (c) ResNet18

(d) DenseNet121 (e) EfficientNet (f) Custom CNN

Fig. 3. Confusion matrices of the different CNN architectures. The matrices show the number of true positives, false positives, true negatives, and false negatives for each model.



(a) ResNet50 with PSO (b) ResNet50 (c) ResNet18

(d) DenseNet121 (e) EfficientNet (f) Custom CNN

Fig. 4. Accuracy curves of different CNN architectures showing training (blue) and validation (orange) accuracy over epochs.

Networks) could significantly enhance data diversity beyond traditional methods. DDGANs can generate realistic synthetic nodule images, addressing class imbalance and improving rare nodule-type recognition.

2) Extending the comparison to 3D CNN architectures that can leverage volumetric information
3) Exploring hybrid optimization approaches that combine PSO with other techniques such as Bayesian optimization or neural architecture search.

The methods and findings presented in this report contribute significantly to the advancement of computer-aided diagnosis (CAD) systems for lung cancer screening. The work addresses several important challenges in the clinical implementation of deep learning for medical image analysis. First, by systematically evaluating multiple CNN architectures using standardized metrics and a common dataset, we provide clear evidence-based recommendations for architecture selection. Second, the PSO optimization framework demonstrates that thoughtful

hyperparameter tuning can yield substantial performance gains (2.11% improvement in accuracy), which could translate to hundreds of more accurately detected nodules in large-scale screening programs.

From a clinical perspective, the higher precision achieved by our optimized model (96.34%) could reduce false positive rates, potentially decreasing unnecessary follow-up procedures that create patient anxiety and burden healthcare systems. Simultaneously, the improved recall suggests better sensitivity for detecting actual nodules, which is critical for early intervention. The PSO-optimized ResNet50 model achieved an AUC of 0.9884, approaching the performance of expert radiologists while offering the benefits of consistency and scalability inherent to automated systems.

REFERENCES

[1] Wang, L., 2022. Deep learning techniques to diagnose lung cancer. Cancers, 14(22), p.5569.

[2] Abraham, G.K., Bhaskaran, P. and Jayanthi, V.S., 2019, November. Lung nodule classification in CT images using convolutional neural network. In 2019 9th International conference on advances in computing and communication (ICACC) (pp. 199-203). IEEE.

[3] A. A. A. Setio et al., "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge," Medical Image Analysis, vol. 42, pp. 1-13, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770-778.

[5] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 6105-6114.

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4700-4708.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., 1995, pp. 1942-1948.

[8] Z. Wang et al., "Artificial intelligence in lung cancer: Current achievements and future challenges," Translational Lung Cancer Research, vol. 11, no. 5, pp. 863-886, 2022.