ECE208/408

Lung Nodule Detection from CT scans using CNN: A Comparative Analysis of CNN Architectures with Hyperparameter Tuning

Arvin Bahreini
Jewelianna Langston
Sarah Metry

Spring 2025

Motivation

- Lung cancer is the leading cause of cancer deaths worldwide.
- Early detection of lung nodules is critical for improving survival rates.
- AI systems can serve as a "second pair of eyes" to help catch nodules that might otherwise be missed.
- This research provides a foundation for more advanced lung nodule analysis, including malignancy prediction, growth rate assessment, and 3D volumetric analysis.

Goal: Develop a CNN-based system for automated, accurate nodule detection.

1. Dataset

LUNA16 Database Includes:

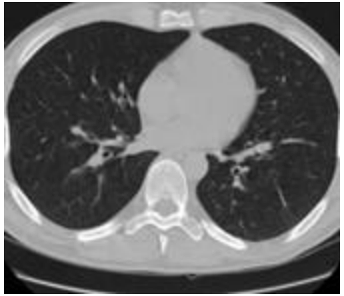
- 888 3D CT scans
  - 601 “With Nodule”
  - 287 “Without Nodule”
- Annotations with nodule cartesian-coordinate locations
- Lung segmentation masks

2. Data Preprocessing

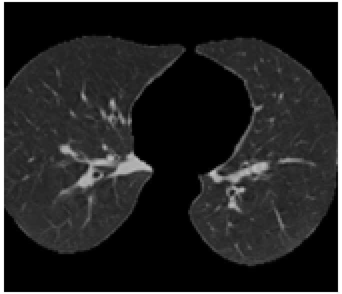
Image Preprocessing Steps:

1) Extracting the image data from the .mhd files
2) Normalizing
3) Converting 2D slices from the 3D scans to .png
4) Applying LUNA16's lung segmentation mask over images

Original



Normalized .png



Segmented & Normalized

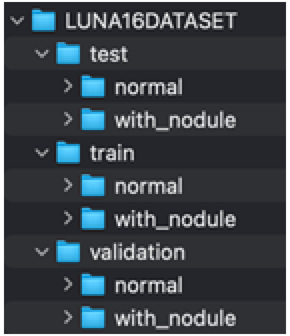
2. Data Preprocessing

Sorting Preprocessing Steps:

1) Sort all scans into training, validation, and testing (60, 20, 20)
2) Check to see if scan has a nodule

If Has a Nodule

- Get nodule location
- Take 6 slices around each nodule's "z-coordinate"



Final Dataset

If Without a Nodule

- Add to list of scans that are without a nodule

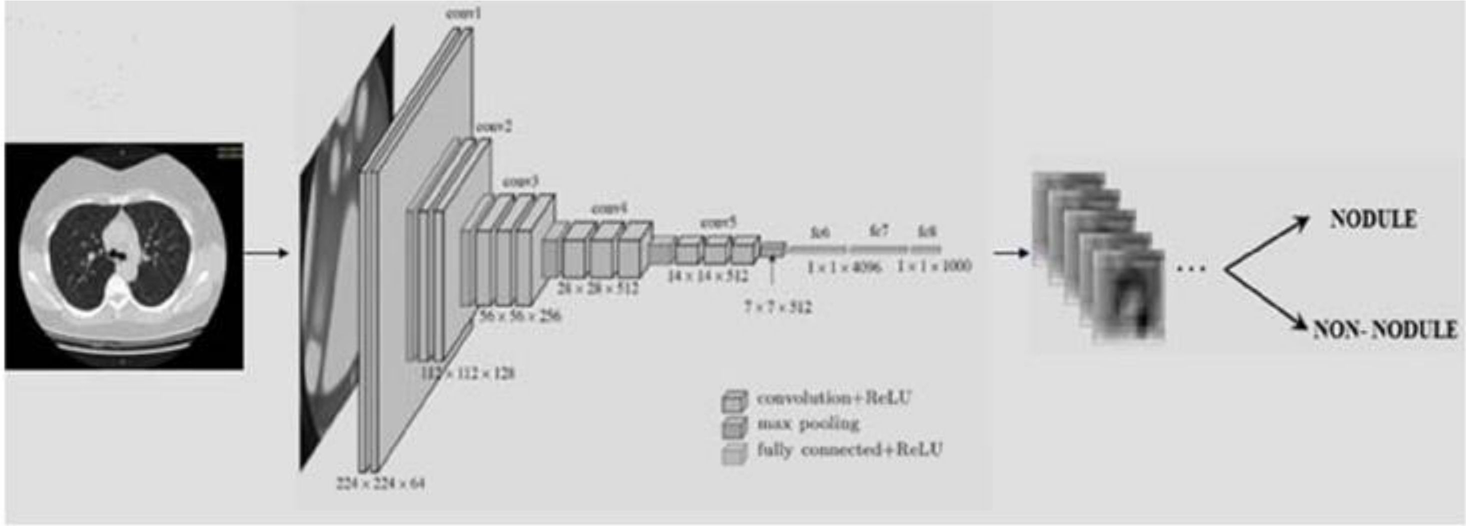
1) Take random images from "Scans Without a Nodule" to match the number of images that contain a nodule

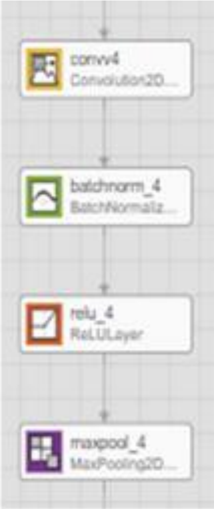
Final Dataset: 7950 training, 2912 validation, 2510 testing (13372 total images)

3. Dataset Processing

- Computed mean and standard deviation of the dataset
- Augmentation applied to the training data
  - Random horizontal flips to simulate different orientations
  - Random rotation (±10°) to add robustness to orientation variations
  - Color Jitter
  - Gaussian Blur
- Data images resized to 224*224

3.1 CNN Architecture Under Test





G. K. Abraham, P. Bhaskaran and V. S. Jayanthi, "Lung Nodule Classification in CT Images Using Convolutional Neural Network," 2019 9th International Conference on Advances in Computing and Communication (ICACC), Kochi, India, 2019, pp. 199-203, doi: 10.1109/ICACC48162.2019.8986213.

3.1.1 Hyperparameter Tuning

- Grid Search: Using Ray Tune library
- Cross-entropy loss criterion
- Adam optimizer
- ReduceLROnPlateau scheduler
- ASHA (Asynchronous Successive Halving Algorithm) scheduler for early stopping underperforming trials

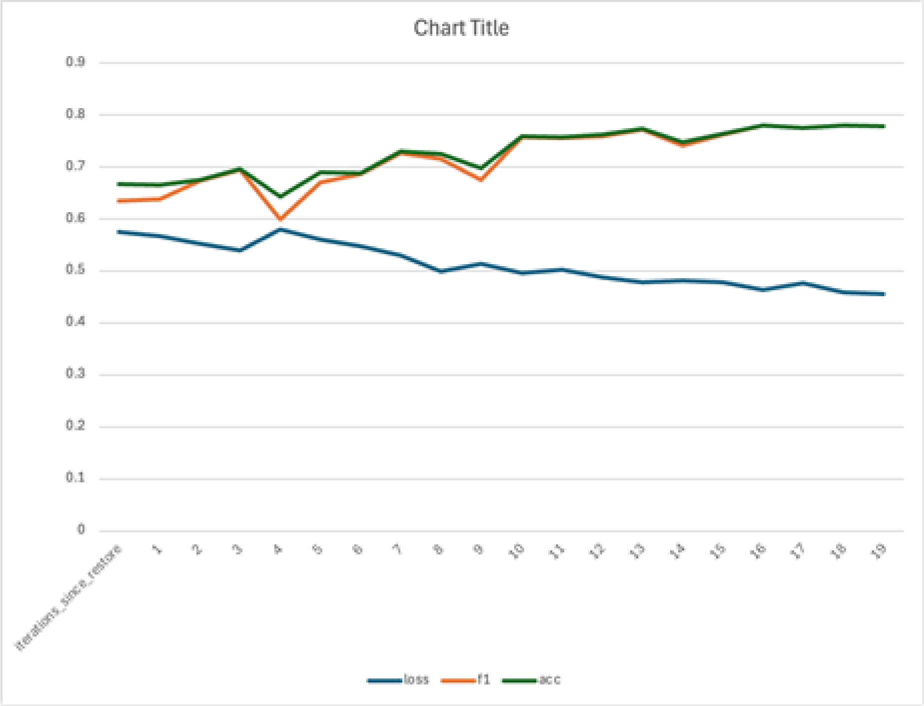
3.1.1 Hyperparameter Tuning

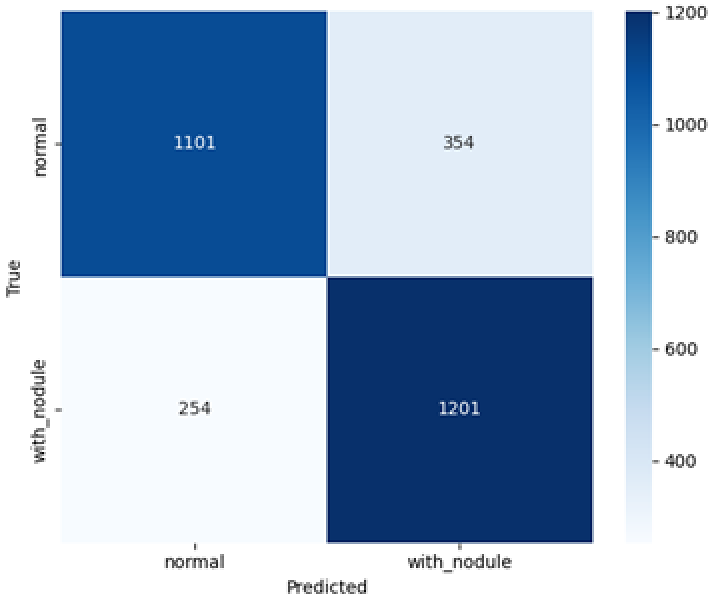
- Search Space Highlights for 20 epochs
  - Learning rate: 0.0001, 0.001, 0.01
  - Weight decay: 0.0001, 0.001, 0.01
  - Batch sizes: 16, 32, 64

Trial name	status	loc	batch_size	lr	wd	iter	total time (s)	loss	f1	acc
train_cnn_d606b_00000	TERMINATED	127.0.0.1:5265	16	0.0001	0.0001	20	7356.18	0.454701	0.778102	0.779038
train_cnn_d606b_00001	TERMINATED	127.0.0.1:9536	32	0.0001	0.0001	4	1276.28	0.561871	0.674568	0.675601
train_cnn_d606b_00002	TERMINATED	127.0.0.1:10721	64	0.0001	0.0001	1	302.364	0.572264	0.63832	0.668385
train_cnn_d606b_00003	TERMINATED	127.0.0.1:10987	16	0.001	0.0001	1	376.355	0.617305	0.589791	0.594845
train_cnn_d606b_00004	TERMINATED	127.0.0.1:11218	32	0.001	0.0001	1	312.751	0.585989	0.64033	0.664605
train_cnn_d606b_00005	TERMINATED	127.0.0.1:11312	64	0.001	0.0001	1	292.983	0.575994	0.605873	0.649485
train_cnn_d606b_00006	TERMINATED	127.0.0.1:11469	16	0.01	0.0001	1	368.516	0.694109	0.333333	0.5
train_cnn_d606b_00007	TERMINATED	127.0.0.1:11561	32	0.01	0.0001	1	318.42	0.693528	0.333333	0.5
train_cnn_d606b_00008	TERMINATED	127.0.0.1:11652	64	0.01	0.0001	4	1283.75	0.667158	0.647653	0.668041
train_cnn_d606b_00009	TERMINATED	127.0.0.1:12536	16	0.0001	0.001	20	8169.43	0.483763	0.755112	0.757732
train_cnn_d606b_00010	TERMINATED	127.0.0.1:17376	32	0.0001	0.001	1	669.895	0.581876	0.627866	0.661512
train_cnn_d606b_00011	TERMINATED	127.0.0.1:17855	64	0.0001	0.001	1	727.407	0.612162	0.602498	0.646392

3.1.1 Training

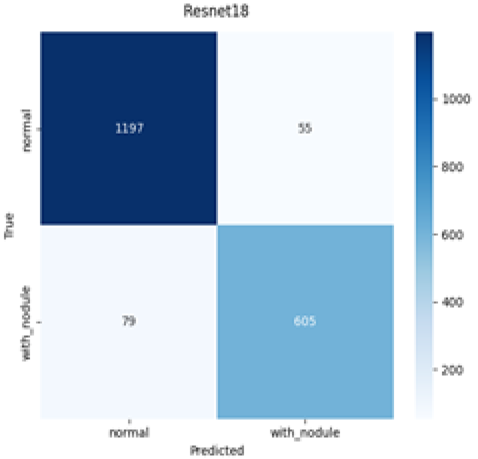
- Validation results using the best hyperparameters for 20 epochs



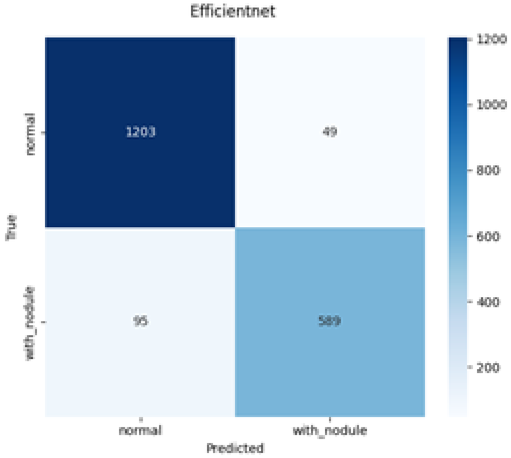


3.2 Utilizing Pre-Trained CNN structures

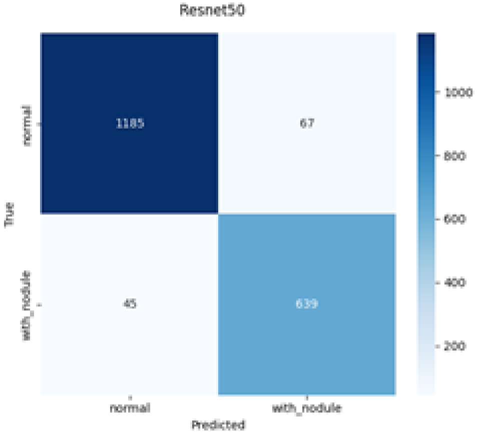
- Structures:
  a. ResNet18
  b. EfficientNet-B0
  c. ResNet50
  d. DenseNet121
- All models pre-trained on ImageNet (transfer learning)
- Can be transferred to other tasks perfectly especially in medical imaging settings
- Modified final classification layer for binary classification (normal vs. nodule)



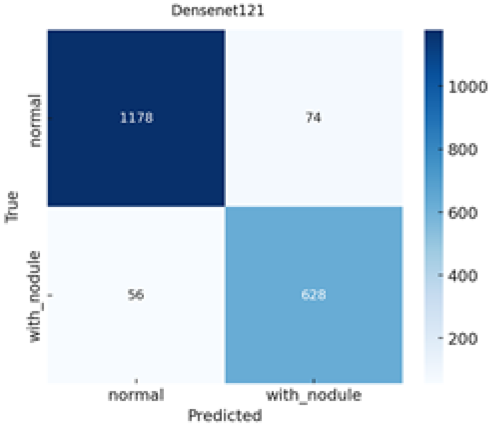
Confusion Matrix of ResNet18



Confusion Matrix of EfficientNet-B0

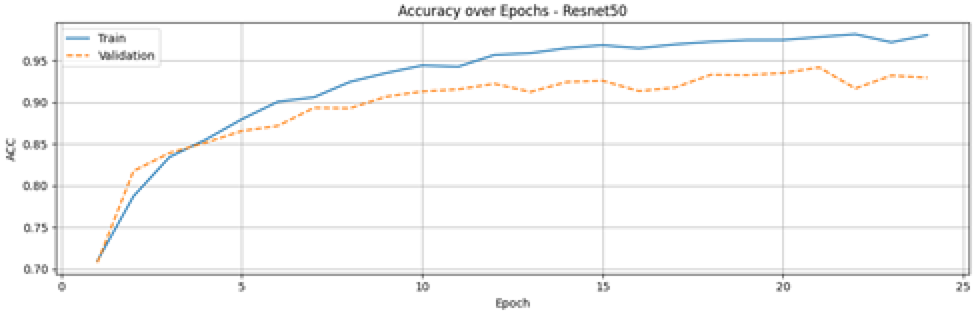


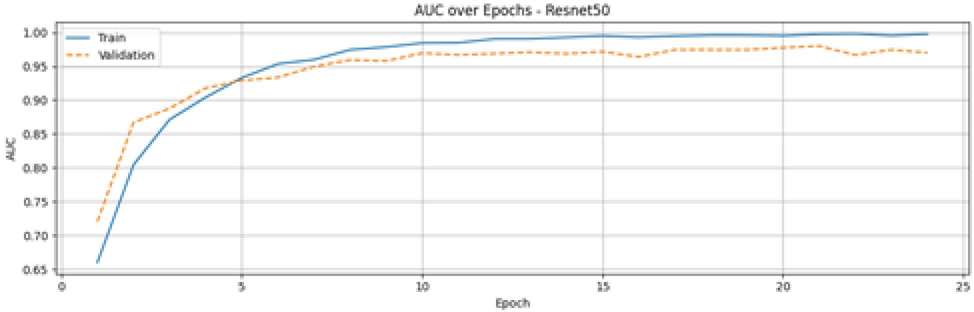
Confusion Matrix of ResNet50



Confusion Matrix of DenceNet121

```
Epoch [1/25] | Train F1: 0.628 | Val F1: 0.661 | Elapsed: 0.2m
Epoch [2/25] | Train F1: 0.767 | Val F1: 0.800 | Elapsed: 1.8m
Epoch [3/25] | Train F1: 0.826 | Val F1: 0.831 | Elapsed: 1.8m
Epoch [4/25] | Train F1: 0.850 | Val F1: 0.852 | Elapsed: 1.8m
Epoch [5/25] | Train F1: 0.877 | Val F1: 0.865 | Elapsed: 1.8m
Epoch [6/25] | Train F1: 0.900 | Val F1: 0.873 | Elapsed: 1.8m
Epoch [7/25] | Train F1: 0.905 | Val F1: 0.893 | Elapsed: 1.8m
Epoch [8/25] | Train F1: 0.925 | Val F1: 0.891 | Elapsed: 1.8m
Epoch [9/25] | Train F1: 0.935 | Val F1: 0.906 | Elapsed: 1.8m
Epoch [10/25] | Train F1: 0.944 | Val F1: 0.914 | Elapsed: 1.8m
Epoch [11/25] | Train F1: 0.943 | Val F1: 0.916 | Elapsed: 1.8m
Epoch [12/25] | Train F1: 0.957 | Val F1: 0.923 | Elapsed: 1.8m
Epoch [13/25] | Train F1: 0.959 | Val F1: 0.911 | Elapsed: 1.8m
Epoch [14/25] | Train F1: 0.965 | Val F1: 0.924 | Elapsed: 1.8m
Epoch [15/25] | Train F1: 0.969 | Val F1: 0.925 | Elapsed: 1.8m
Epoch [16/25] | Train F1: 0.965 | Val F1: 0.913 | Elapsed: 1.8m
Epoch [17/25] | Train F1: 0.970 | Val F1: 0.916 | Elapsed: 1.8m
Epoch [18/25] | Train F1: 0.973 | Val F1: 0.933 | Elapsed: 1.8m
Epoch [19/25] | Train F1: 0.975 | Val F1: 0.933 | Elapsed: 1.8m
Epoch [20/25] | Train F1: 0.975 | Val F1: 0.935 | Elapsed: 1.8m
Epoch [21/25] | Train F1: 0.979 | Val F1: 0.942 | Elapsed: 1.8m
Epoch [22/25] | Train F1: 0.982 | Val F1: 0.918 | Elapsed: 1.8m
Epoch [23/25] | Train F1: 0.972 | Val F1: 0.932 | Elapsed: 1.8m
Epoch [24/25] | Train F1: 0.981 | Val F1: 0.929 | Elapsed: 1.8m
Epoch [25/25] | Train F1: 0.977 | Val F1: 0.934 | Elapsed: 1.8m
Training complete. Best Val F1: 0.942
```





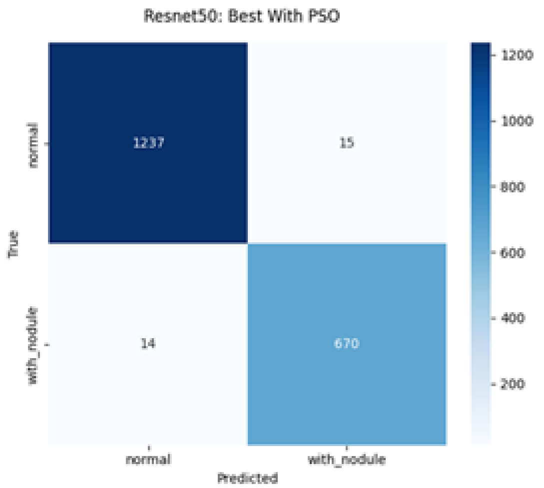
ResNet50

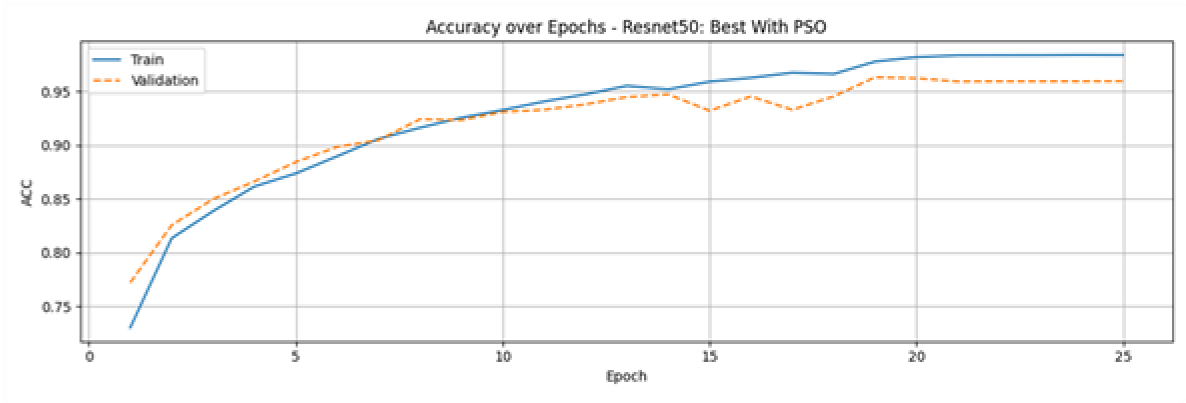
We calculated Loss, Accuracy, Precision, Recall, F1 score, Area Under the ROC Curve during 25 epochs

3.2.1 Hyperparameter Tuning

- Grid Search → used in previous approach
- Particle Swarm Optimization (PSO) → ✅

- Learning rate (lr) = 4.44e-4
- Weight decay (wd) = 1.66e-4
- Batch size (batch_size) = 44
- Beta1 (beta1) - First momentum coefficient for AdamW optimizer = 0.712
- Beta2 (beta2) - Second momentum coefficient for AdamW optimizer = 0.95
- Epsilon (eps) - Small value for numerical stability in AdamW = 3.9e-7





4. Result and Analysis

Model	Loss	Accuracy	Precision	F1	AUC	Size	Train-Time for one Epoch
Efficientnet	0.2538	92.45	0.9257	0.9249	0.968	20.5 M	1.0 minute
Resnet18	0.2495	92.56	0.9255	0.9249	0.9741	44.7 M	0.9 minute
Densenet121	0.2342	93.07	0.9311	0.9304	0.9735	30.8 M	1.9 minute
Resnet50	0.2018	94.21	0.9428	0.9423	0.9801	97.8 M	1.8 minute
Resnet50: Best With PSO	0.1393	96.32	0.9634	0.9633	0.9884	97.8 M	1.8 minute

- After PSO: reducing loss by 31%
- ResNet50 with PSO achieved the highest performance
- 2.11% accuracy improvement → thousands of true negative samples
- EfficientNet offers the best performance-to-size ratio

5. Conclusion and Future Work

Conclusion:

- Pre-trained models performed better than custom CNN model which shows transfer learning effectiveness
- The ResNet50 architecture gives the best performance especially when optimized with PSO
- Trade-offs between model size, training efficiency, and detection performance
- PSO with limited iterations provided improvement while managing computational constraints.

Future Work:

- Advanced Data Augmentation using DDGAN
- 3D CNN models
- Exploring Bayesian optimization or neural architecture search

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

Thank You!