Improved convolutional neural network lung cancer classification detection method based on transfer learning and model compression

Liu Fangxing   
*Computer Science and Technology  
Huazhong University of Science and Technology*Wuhan, China  
1198971412@qq.com

Chen Ju  
*School of Optical and Electronic Information*  
*Huazhong University of Science and Technology*Wuhan, China  
1064314670@qq.com Pi Yaxin  
*School of Economics*  
*Huazhong University of Science and Technology*Wuhan, China  
2710324892@qq.com

Teoh Teik Toe  
*NTU Business AI Lab  
Nanyang Technological University*Singapore  
ttteoh@ntu.edu.sg

Abstract—Lung cancer, one of the most life-threatening malignancies, has witnessed a significant increase in its incidence and mortality rates over the past few decades. Accurate determination of the histological type of lung cancer is crucial for developing appropriate treatment strategies. However, traditional medical imaging analysis, such as magnetic resonance imaging and computed tomography (CT), faces challenges due to sensor noise, patient variations, and pathological states, which may lead to misinterpretation and errors. In this regard, deep learning approaches can assist in rapidly distinguishing different types of lung cancer by learning complex patterns and features from medical imaging data, aiding clinicians in precise diagnosis. This research aims to address the limitations of existing lung cancer classification models, including suboptimal accuracy, slow inference speed, high computational and storage requirements, and vulnerability to overfitting. To achieve this, we propose an improved lung cancer classification and detection method based on deep learning. Our approach leverages transfer learning by utilizing pre-trained models, data augmentation techniques, and model compression methods, including global unstructured pruning and model distillation. Experimental results demonstrate that the teacher model achieves 100% accuracy, recall, precision, and F1-score on the dataset, with a loss of 0.00007575038. The distilled student model, after pruning 55.15% of its parameters, achieves a 52.5% reduction in size. It also attains 100% accuracy, recall, precision, and F1-score, with an average loss of 0.004158932.

Keywords—Lung cancer, deep learning, transfer learning, model compression, CNN

# Introduction

One of the most dangerous types of malignant tumors that can seriously threaten human health and life is lung cancer, which develops from either the bronchial mucosa or glands in the lungs[1]. Over the last 50 years, numerous countries have observed a substantial rise in both the occurrence and fatality rate of lung cancer. The tissue type of lung cancer is closely related to its etiology, biological characteristics, clinical manifestations, prognosis, and treatment response. It is important to determine the tissue type of lung cancer before formulating a treatment plan[2]. Lung cancer can be classified into two main types: small cell lung cancer (SCLC) and non-small cell lung cancer (NSCLC). The majority of patients, accounting for more than 75%, are diagnosed with non-small cell lung cancer. The histological subtypes of non-small cell lung cancer can be further divided into adenocarcinoma (ADC), squamous cell carcinoma (SqCC), etc.[3]. There are different clinical treatment methods and prognostic effects for different subtypes. Accurate classification of lung cancer subtypes helps to develop better clinical treatment methods and strategies, and also provides patients with more suitable diagnosis and treatment plans.

The rapid advancements in deep learning have brought about significant enhancements in the processing of image data through the emergence of convolutional neural networks. These networks have revolutionized image classification and pattern recognition by providing efficient end-to-end systems[4,5]. In this study, our focus is on constructing a simple yet highly accurate model. Therefore, we have selected ResNet-18[6] as the foundational model for our improvements. This dataset we used is sourced from Kaggle. The dataset is categorized into three groups: Lung benign tissue (Lung\_n), Lung adenocarcinoma (Lung\_aca), and Lung squamous cell carcinoma (Lung\_scc).

In this paper, we present an improved lung cancer classification and detection method based on deep learning, transfer learning, and model compression. We leverage transfer learning by utilizing pre-trained models to enhance the learning process on a relatively small lung cancer dataset. Data augmentation techniques are applied to increase the dataset's diversity and mitigate overfitting. Model compression techniques, including global unstructured pruning and model distillation, are employed to reduce the model size and computational complexity while maintaining or even improving the model's performance.

We have shown the effectiveness of our proposed method through extensive experimentation. The teacher model achieves 100% accuracy, recall, precision, and F1-score on the lung cancer dataset, with significantly reduced loss. The distilled student model achieves a remarkable reduction in model size while maintaining 100% accuracy, recall, precision, and F1-score. The results highlight the potential of our approach to address the limitations of existing lung cancer classification models, including their accuracy, inference speed, computational and storage requirements, and overfitting.

In conclusion, our proposed method, combining transfer learning and model compression techniques, offers an improved approach for lung cancer classification and detection. By enhancing the accuracy, reducing computational requirements, and addressing overfitting, our approach has the potential to assist clinicians in making more reliable and efficient diagnoses, ultimately aid in the advancement of lung cancer research and diagnosis.

To obtain the experimental code or further experimental details related to this paper, please visit https://github.com/hmny123/IEEE-paper.

# METHODOLOGY

## Transfer Learning

Transfer learning is a technique in machine learning that utilizes knowledge acquired from one task to another related task, with the aim of accelerating the learning process, increasing the accuracy of the model, and improving its generalization performance. In traditional machine learning, training a new model often requires a large amount of data and computing resources. However, obtaining such resources can be difficult in practice, leading to high costs and training difficulties. Transfer learning can help overcome this challenge by leveraging existing models and data. There are three types of transfer learning:

* Parameter-based transfer learning: uses the parameters of a pre-trained model as a starting point for the new model, accelerating the learning process.
* Feature-based transfer learning: uses the learned features from a pre-trained model in a new model to improve generalization performance.
* Model-based transfer learning: uses the entire pre-trained model in a new model to improve accuracy and generalization performance. This is also the method we use in the article. In practical use, there are two different ways to improve the pre-trained model after it is selected. One is called Fixed Feature Extractor, which freezes the first few layers of the network and trains only the fully connected classification layers. The advantage of this approach is that it can fully reuse the pre-training set weights and features. Another method is called Finetune, which initializes its own network weights with the pre-trained network weights instead of the original random initialization, which is suitable for cases where the pre-trained set does not match as well. The specific choice of method depends on the fit of the migration training set to the pre-training set.

Transfer learning has several advantages, including:

* It reduces training time and costs by utilizing existing models and data.
* It enhances model accuracy and generalization performance, reducing overfitting and underfitting issues.
* It solves the problem of data sparsity by using existing data to improve learning performance. In other words,transfer learning is well suited to small sample data, which is the case we are going to use in this article.

## RESNET

In traditional CNN models, increasing the depth of the network does not necessarily lead to a decrease in training error. On the contrary, it may result in an increase in training error. This is largely due to the fact that during the training process of networks that use stochastic gradient descent, the back propagation of error signals through multiple layers can cause the gradients to either vanish or explode, leading to unstable training. In order to effectively solve this decline phenomenon in the network, the ResNet network structure was proposed. This structure is relatively easy to optimize, and its network performance far surpasses traditional network models.

The main feature of ResNet is the introduction of the concept of the residual block, as shown in Figure 1, which is used to solve the problems of gradient vanishing and gradient explosion in deep convolutional neural networks. In residual blocks, shortcut connections can directly connect inputs to outputs, allowing the network to learn residual information and perform better in feature extraction and processing.

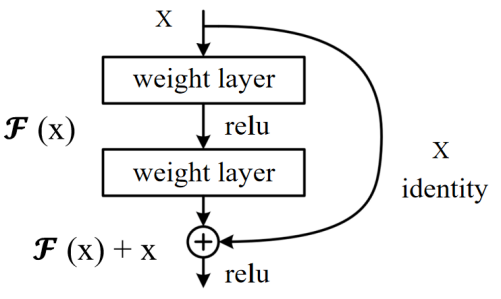


Fig. 1. The structure of the residual block

As shown in Figure 2, each module in this ResNet model has 4 convolutional layers. Adding the first 7x7 convolutional layer and the last fully connected layer, there are a total of 18 layers. Therefore, this model is commonly referred to as ResNet-18. ResNet-18 is a very classic and effective deep convolutional neural network model with excellent feature extraction and classification capabilities, which can be applied to computer vision tasks such as image classification and object detection.

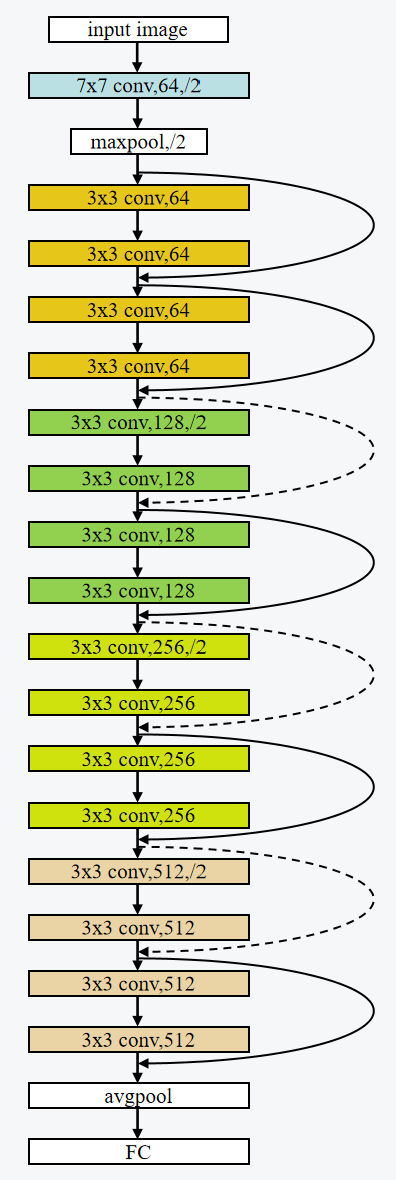


Fig. 2. The structure of ResNet-18

## Global Unstructured Pruning and Fine-tuning

Global Unstructured Pruning, as referenced in [7,8,9], is an advanced model compression strategy that uniformly targets all model parameters, including weights and biases, eschewing the specificity of individual layers or structures. The core strategy is to create a sparse model by nullifying a pre-determined fraction of parameters, generally guided by a pruning ratio. This procedure involves sorting weights by their absolute magnitude, and setting the smallest percentile to zero, thereby effectively reducing computational overhead and storage requirements while aiming to preserve model performance.

Nevertheless, the pruning process may instigate a decrease in model performance. Hence, a crucial step that ensues pruning is fine-tuning. During this stage, the weights of the pruned model are recalibrated using a reduced learning rate to prevent large fluctuations that may result in overfitting. Fine-tuning is pivotal in facilitating the pruned model's adaptation to its new architecture, ensuring it remains proficient in tackling the original task while maintaining minimal model size and computational complexity. In the context of this experiment, fine-tuning effectively reinstated the model's performance to a level comparable to the original, pre-pruned model.

## Knowledge Distillation

Following the pruning and fine-tuning processes, Knowledge Distillation is enacted as proposed in [7,10]. This methodology necessitates the pruned model to function as a student, with the original model serving as the teacher. Specifically, ResNet18 from torchvision models is utilized as the student model. Through Knowledge Distillation, the pruned model learns to emulate the performance of the original, larger model, effectively distilling the "knowledge" of the expansive model into the more compact, pruned model.

# EXPERIMENTAL STUDY

## Data Augmentation

Data augmentation is a technique of generating new data by applying various transformations to the existing data set. The goal is to increase the diversity and quantity of data, which can improve the performance and robustness of machine learning models. The common data augmentation methods we use are as follows.

* Flipping: horizontally or vertically flipping images to create new ones. This method is often used in image classification, object detection, and other related tasks.
* Cropping: cropping images into different sizes and positions to create new ones. This method is often used in image classification and object detection.
* Scaling: scaling images by a certain ratio to create new ones. This method is often used in image classification and object detection.

In addition to the methods we use, common methods include rotation, translation, noise addition, contrast or colour adjustment, etc.

## Teacher Model Compilation

* Optimizer

Our training methodology employs the Adam optimizer, an acronym for Adaptive Moment Estimation. Specifically designed for deep learning applications, Adam extends the stochastic gradient descent method. This optimizer brilliantly amalgamates the salient features of two other well-established optimization algorithms - AdaGrad and RMSProp, thereby facilitating an optimization strategy that can adeptly handle sparse gradients in noisy problem scenarios.

Adam's computational strategy involves calculating an exponential moving average of the gradient and the squared gradient. These moving averages decay at rates determined by the hyperparameters beta1 and beta2. The initial step size, which is tantamount to the learning rate, can be regulated using the hyperparameter alpha.

Adam offers several distinguishing characteristics:

* It is simple to implement and demonstrates high computational efficiency with minimal memory requirements.
* Its parameter updates exhibit invariance to the rescaling of the gradient, which aids in maintaining the relative updates of model weights.
* Its hyperparameters have intuitive implications and generally require minimal tuning.
* The step sizes are effectively bounded by the learning rate, which assists in preventing excessively large parameter updates.
* It exhibits robust capability in handling sparse gradients and noisy data, making it particularly suitable for scenarios with extensive data and parameters.
* It is effective for non-stationary objectives
* It implements a form of step size annealing, which aids in stabilizing the learning process and reduces the need for manual tuning of the learning rate.

In practical applications, Adam often outperforms SGD in terms of convergence speed and can yield superior performance on the test set.

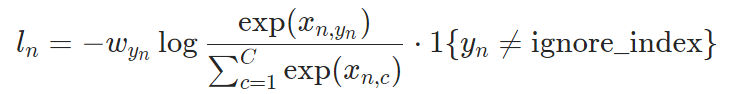
* Loss

For our multi-class problem, we deploy Cross-Entropy Loss as our objective function. This loss function is particularly suitable for multi-class classification problems where the target labels are one-hot encoded. Cross-Entropy Loss comprises a Softmax activation and a Cross-entropy Loss.

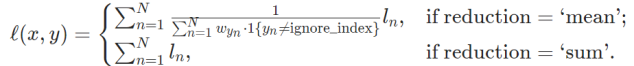
The computation of Cross-Entropy Loss involves two integral parts:

* Softmax activation: This activation function transforms raw score vectors, or logits, into probabilities by exponentiating them and then normalizing them.
* Cross-Entropy Loss: The Cross-Entropy Loss is calculated by taking the negative logarithm of the likelihood of the correct class. To use this criterion, the target should consist of class indices ranging from 0 to C, where C represents the total number of classes. Additionally, if an "ignore index" is specified, this loss can also accept that particular class index, which may not necessarily fall within the class range. The unreduced loss for this case, without any reduction applied, can be represented as shown in equations (1) and (2).

C:\Users\hmny\AppData\Roaming\Tencent\Users\1589575290\QQ\WinTemp\RichOle\FA4{RI]R1KBWG{9W5U{UZ)M.png 

 

Here, x denotes the input, y represents the target, w is the weight, C denotes the number of classes, and N spans the minibatch dimension, as well as d1,...,dk for the K-dimensional scenario. If is not the default reduction method, then equation (3) applies.

 

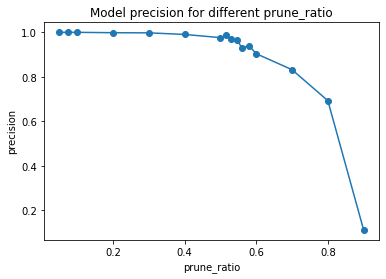
It is worth noting that this case is equivalent to the combination of LogSoftmax and NLLLoss.

The Softmax operation ensures the sum of the output probabilities across all classes is unity, with the output probability of each class being influenced by other classes. This property makes Cross-Entropy Loss suitable for multi-classification problems, as it penalizes the model heavily when it assigns high probability to the incorrect class. Therefore, the Cross-Entropy Loss guides the model to refine its predictions by iteratively adjusting the parameters to minimize the loss.

## Model Compression Process

* Global Unstructured Pruning and Fine-tuning

In alignment with the methodologies proposed in [7,8,9], our initial step involved applying Global Unstructured Pruning to the baseline model. A critical decision in this procedure was the determination of an appropriate pruning ratio. Following extensive experimentation with a spectrum of ratios (See Figure 3), we identified that a pruning ratio of 0.515 struck an optimal balance between model sparsity and computational performance. With this ratio, we proceeded to nullify the least significant 51.5% of weights, as sorted by their absolute value, within the model parameters.



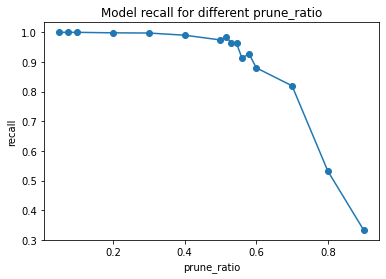
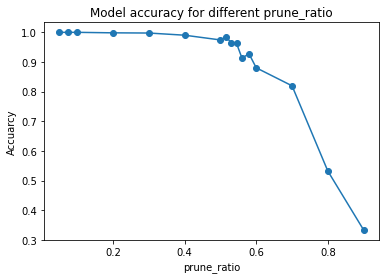
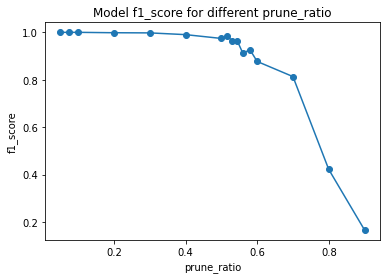
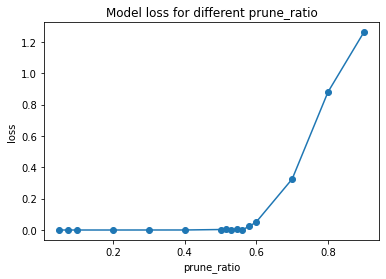


Fig. 3. Evaluations for different prune ratio

* Knowledge Distillation

Following the pruning and fine-tuning processes, Knowledge Distillation is enacted as proposed in [7,10]. This methodology necessitates the pruned model to function as a student, with the original model serving as the teacher. Specifically, ResNet18 from torchvision.models is utilized as the student model.

Through Knowledge Distillation, the pruned model learns to emulate the performance of the original, larger model, effectively distilling the "knowledge" of the expansive model into the more compact, pruned model.

# RESULT AND EVALUATION

## Evaluation Indicators

* Test accuracy: The proportion of correctly classified samples in the test set.





* Test recall: The proportion of true positive samples among all samples that belong to a certain class.





* Test loss: We deploy Cross-Entropy Loss as our objective function.
* Test precision: The proportion of true positive samples among all samples that are predicted to belong to a certain class.





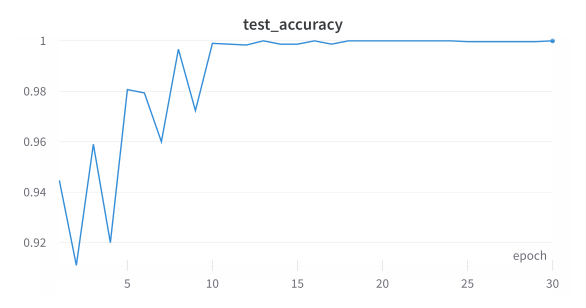
* F1 score: A commonly used performance metric that combines both precision and recall into a single score. It is the harmonic mean of precision and recall, and it ranges from 0 to 1, with a higher score indicating better performance.

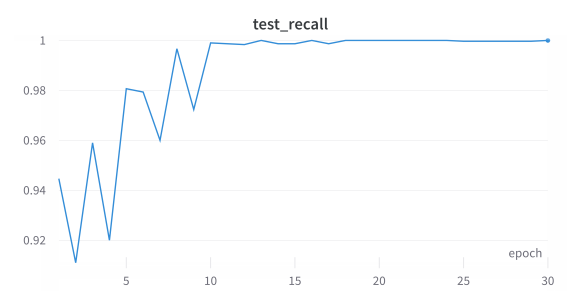


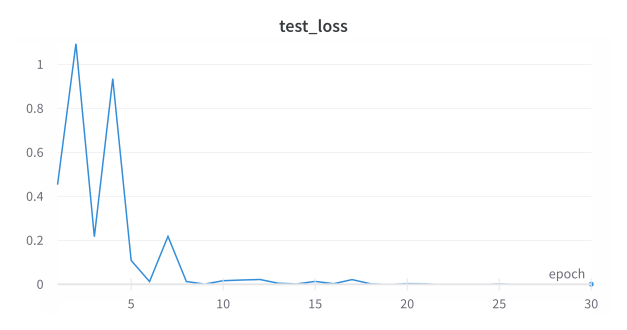
## Test Results

* Resnet50

As shown in Table I, The accuracy rate of the test set using resnet50 is 1. Under our evaluation metrics, the precision rate of the model is 1, the recall rate and F1 value are both 1, and the test loss is 7.575\*10^-5. Compared with the currently available articles performing lung cancer classification, our model performs better in test accuracy and does not lose a significant amount of test loss.







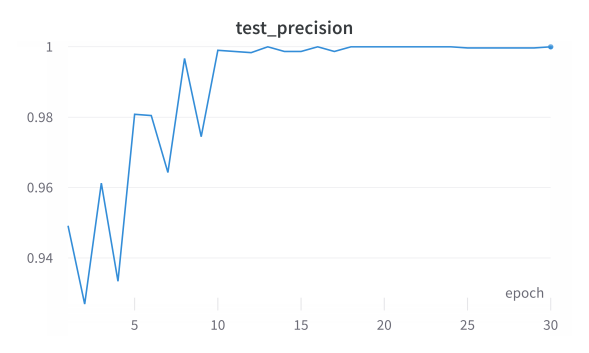
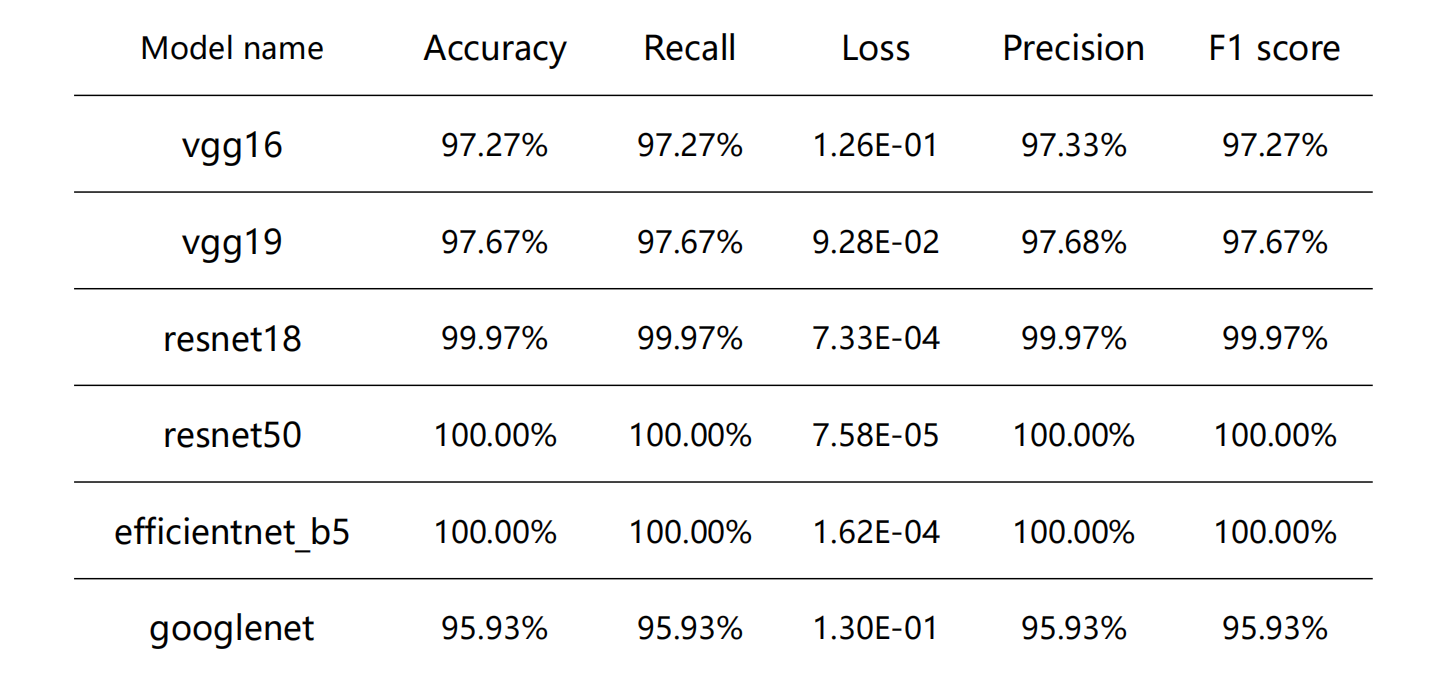


Fig. 4. Evaluations for Resnet50

* Teacher Model Choices

In addition to resnet50, we also tried several of the more widely used pre-trained network models, including vgg16, vgg19, resnet18, efficientnet\_b5, and googlenet. Comparing the results of resnet18 using Fixed Feature Extractor and Finetune respectively, we found that Finetune's results were significantly better, probably because our dataset for lung cancer classification did not match as well with the pre-training set of resnet18. Finetune also performs significantly better than Fixed Feature Extractor in the vgg16 network, so we choose to fine-tune all parameter layers in all pre-training networks to improve the model. The final results of different models are compared as shown in Table 1.



1. Comparison Results

By comparison, resnet50 and efficientnet\_b5 perform better, but from the test loss data, resnet50 is slightly better.

* Model Compression Results

The student model was subjected to a 50-epoch training regime, during which it learnt to replicate the behavior of the teacher model. The overarching aim was to distill the 'knowledge' encapsulated in the pruned model into the more compact ResNet18 model. Upon the completion of the distillation process, the ResNet18 model not only matched the performance of the pruned model but occasionally outperformed it, all while maintaining a significantly reduced model size.

The original model size was 92180 KB, while the compressed model size after pruning and distillation was 43750 KB. The compression rate was calculated as:

Compression rate = (Original model size - Compressed model size) / Original model size = (92180 - 43750) / 92180 ≈ 52.5%

This model compression method, inspired by the PEEL technique mentioned in [7], resulted in a compression rate of 52.5% with only a minor trade-off in terms of accuracy. These results demonstrate the efficacy of this model compression method.

# CONCLUSION

In this study, we proposed an improved convolutional neural network (CNN) lung cancer classification and detection method based on transfer learning and model compression. The motivation behind this research was the need for accurate and efficient classification of lung cancer types, considering the significant threat it poses to human health and the increasing incidence and mortality rates worldwide.

Through extensive experimentation, we demonstrated the effectiveness of our approach. The teacher model, trained using transfer learning and data augmentation techniques, achieved remarkable results on the dataset. It attained 100% accuracy, recall, precision, and F1-score, with a minimal loss of 0.00007575038. This indicates the robustness of the model in accurately identifying different types of lung cancer.

To address the challenges associated with computational resource requirements, storage limitations, and slow inference speed, we applied model compression techniques. Specifically, we employed global unstructured pruning, resulting in a reduction of 55.15% in model parameters. Despite the significant compression, the student model achieved outstanding performance, with 100% accuracy, recall, precision, and F1-score, along with an average loss of 0.004158932.

Our improved CNN lung cancer classification and detection method offers several advantages. Firstly, it achieves high accuracy, fulfilling the requirements for precise medical classification. Secondly, by incorporating model compression, it significantly reduces the model's size and computational complexity, enabling its deployment on resource-constrained devices and applications. Moreover, the compressed model exhibits faster inference speed, which is crucial for real-time diagnosis. Lastly, the approach effectively mitigates the issue of overfitting that arises with large-scale CNN models.

In summary, our improved CNN lung cancer classification and detection method, incorporating transfer learning and model compression, effectively addresses the limitations of existing models. It offers accurate classification, fast inference speed, reduced computational and storage requirements, and mitigates overfitting. This research contributes to the advancement of computer vision in the medical domain and holds promise for assisting healthcare professionals in the diagnosis and management of lung cancer.

# FUTURE WORK

Future research directions include expanding the dataset and exploring additional data enhancement methods to address sample insufficiency issues. This will enable us to improve the generalizability and robustness of the model. Additionally, further investigations into model compression techniques can be pursued to optimize the size and computational efficiency of the model. By combining these efforts, we aim to provide medical professionals with more reliable support in diagnosing and treating lung cancer.

Acknowledgment

Liu Fangxing, Pi Yaxin, and Chen Ju would like to express our sincere gratitude to Professor Teoh Teik Toe, the director of the AI lab at Nanyang Technological University, for his invaluable guidance, constant encouragement, and insightful instructions throughout the course of this research. The corresponding author of this paper is Liu Fangxing, who is also the first author. For any inquiries or correspondence regarding this research, please contact Liu Fangxing at 1198971412@qq.com.

##### References

1. L. A. Torre, R. L. Siegel, and A. Jemal, “Lung Cancer Statistics,” Advances in Experimental Medicine and Biology,2016,893.
2. Emmanuel Rios Velazquez, Chintan Parmar, Ying Liu, Thibaud P. Coroller, Gisele Cruz, Olya Stringfield, Zhaoxiang Ye, Mike Makrigiorgos, Fiona Fennessy, Raymond H. Mak, Robert Gillies, John Quackenbush, Hugo J.W.L. Aerts, “Somatic Mutations Drive Distinct Imaging Phenotypes in Lung Cancer,” Cancer Res, vol. 77, pp. 3922– 3930, July 2017.
3. C. Gridelli, A. Rossi, D. Carbone, and et al. “Non-small-cell lung cancer,” Nat Rev Dis Primers 1, May 2015.
4. Nora yahia Ibrahim,Amira Samy Talaat. An Enhancement Technique to Diagnose Colon and Lung Cancer by using Double CLAHE and Deep Learning[J]. International Journal of Advanced Computer Science and Applications (IJACSA),2022,13(8).
5. Masud Mehedi,Sikder Niloy,Nahid AbdullahAl,Bairagi Anupam Kumar,AlZain Mohammed A.. A Machine Learning Approach to Diagnosing Lung and Colon Cancer Using a Deep Learning-Based Classification Framework[J]. Sensors,2021,21(3).
6. H. Kaiming, Z. Xiangyu, R. Shaoqing, et al. “Deep residual learning for image recognition,” IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp:770-778.
7. Hou, Y., Ma, Z., Liu, C., Wang, Z., & Loy, C. C. (2021). Network pruning via resource reallocation. arXiv preprint arXiv:2103.01847.
8. Iofinova, E., Peste, A., & Alistarh, D. (2023). Bias in pruned vision models: In-depth analysis and countermeasures. arXiv preprint arXiv:2304.12622.
9. Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In Advances in neural information processing systems (pp. 1135-1143).
10. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.