Pneumonia Image Classification: Deep Learning and Machine Learning Fusion

Jiarui Tang
Computer Science and Engineering
UESTC
Chengdu, China
2021080903012@std.uestc.edu.cn

Bohua Zhang
Electronic Science and Engineering
UESTC
Chengdu, China
2022360902021@std.uestc.edu.cn

Jinzhou Liu
Electronic Science and Engineering
UESTC
Chengdu, China
2021190506013@std.uestc.edu.cn

Zhuoling Dong
Electronic Science and Engineering
UESTC
Chengdu, China
2022360902008@std.uestc.edu.cn

Yangbin Zhou
Spatial Information and Digital Technology
UESTC
Chengdu, China
2021070903011@std.uestc.edu.cn

Xingyu Meng
Electronic Science and Engineering
UESTC
Chengdu, China
2021190501002@std.uestc.edu.cn

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

***Abstract*—This paper introduces a comprehensive method for diagnosing pneumonia in chest X-ray (CXR) images. By employing deep learning models for feature extraction and machine learning methods for classification, we evaluated the effectiveness of feature extraction using ResNet50, VIT, and Inception v3 models, and compared the performance of random forest and logistic regression classifiers based on these models. A comprehensive assessment of accuracy, F1 score, and AUC values was conducted to determine the most effective model. This study addresses the challenge of manual pneumonia diagnosis and emphasizes the urgency of automated solutions. Ultimately, the combination of Inception v3 and logistic regression achieved higher accuracy (79.32%), a higher F1 score (0.6239), and a more ideal AUC value (0.94). These results enhance the clinical efficiency and accuracy of pneumonia detection, benefiting patient treatment outcomes.**

***Index Terms*—ResNet50, VIT, Inception v3, Machine Learning, Logistic Regression, Random Forest, pneumonia**

I. Introduction

A. Background

Since several decades, pneumonia has been a worldwide significant health concern, often requiring prompt and accurate diagnosis for effective treatment. Chest X-ray (CXR) pictures are manually inspected by radiologists in traditional techniques of detecting pneumonia. This process can be laborious and prone to human error [1]. Automation of this diagnostic process has become possible because of the development of deep learning and machine learning techniques in recent years.

UESTC means University of Electronic Science and Technology of China.

B. Main method

Using machine learning algorithms for classification and deep learning models for feature extraction, this research presents a complete approach for diagnosing pneumonia from chest X-ray pictures. To be more precise, the performance of random forest and logistic regression classifiers based on ResNet50, VIT, and Inception v3 models is compared, and the efficacy of feature extraction using these models is assessed. The difficulty of diagnosing pneumonia by hand is discussed in this paper, which also emphasizes the need for automated alternatives.

In order to identify important patterns and information in images, feature extraction is a critical step in the image classification process. For feature extraction, deep learning models like ResNet50, VIT, and Inception v3 are used. These models each have specific benefits in capturing picture features at various scales and complexities.

Furthermore, classification algorithms are essential elements in the image recognition process, allowing a huge number of images to be automatically classified using features that have been retrieved. In this study, random forest classifiers and logistic regression are taken into consideration for the diagnosis of pneumonia.

C. Literature review

The literature that is currently available supports the introduction of automated diagnosis techniques. Using deep learning techniques, Wang et al. [2] have established automated diagnosis of pneumonia using CXR pictures. A review of deep learning methods for X-ray image-based pneumonia

	LeNet	AlexNet	GoogLeNet	IVGG13
Training time	49 s	284 s	1270 s	231 s
Accuracy	76.6%	73.6%	73.3%	**77.5%**
Precision	73.0%	70.6%	78.8%	**73.7%**
Recall	99.2%	98.9%	78.4%	**99.4%**
F1-Measure	84.1%	82.4%	78.6%	**84.6%**

Fig. 1. Prediction Result of Each Network Model

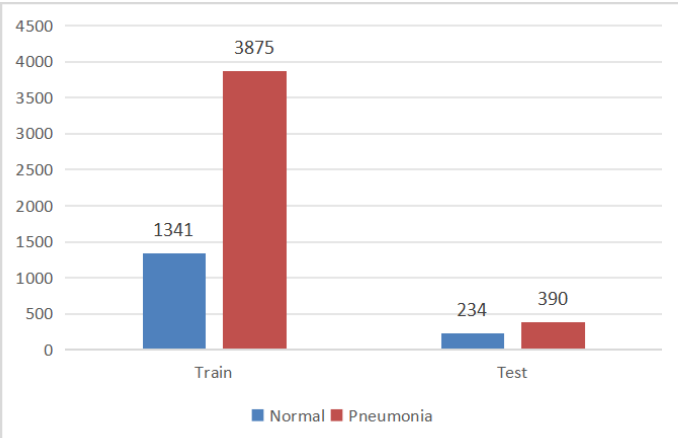

Fig. 2. Arrangement of Dataset

identification was presented by Chen et al. [3]. Moreover, the use of stacking generalization of models for tuberculosis (TB) detection in chest radiographs has been investigated by Rajaraman et al. [4]. These investigations highlight the continuous progress in medical image processing, especially in the area of pneumonia detection. According to [5], various training models were used to recognize CXR pictures and "Figure 1" shows the prediction result of these models. Thus, this paper aims to attempt more strategies to get better results.

D. Dataset

The dataset used for training is anterior-posterior chest X-ray images chosen from retrospective cohorts of pediatric patients from Guangzhou Women and Children's Medical Center, Guangzhou. It has subfolders for each image category (Pneumonia/Normal) and is arranged into three folders (train, test, and val). There are two categories (Pneumonia/Normal) and 5,863 X-ray images (JPEG). "Figure 2" illustrates the number of samples of different categories in the dataset.

All chest radiographs were first screened for quality control by eliminating any low quality or unreadable scans before being subjected to the analysis of chest x-ray pictures. Before the photos' diagnoses could be used to train the AI system, they were evaluated by two board-certified medical professionals. In order to account for any grading errors, the evaluation set was also checked by a third expert.

II. METHODOLOGY

"Figure 3" illustrates the overall process of this study. Initially, chest X-ray images are divided into a training set and a test set, with the training set serving as the training

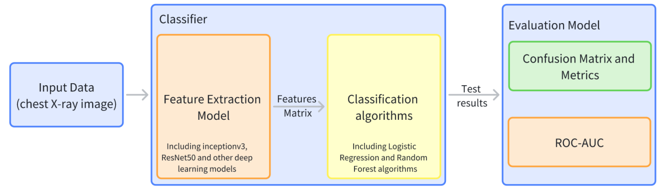

Fig. 3. Pipeline of Proposed Method

data for the classifier. The classifier consists of a feature extraction model and a classification algorithm, where the feature extraction model transfers the extracted feature matrix from the images to the classification algorithm. Ultimately, running the test data through the classifier yields classification results, which are then evaluated using an evaluation model. The evaluation model utilizes confusion matrix along with its parameters and the ROC-AUC method, comprehensively considering the accuracy and stability of the model to derive a comprehensive evaluation result.

A. Extracting features using deep learning methods

Feature processing is crucial in image classification as it involves operations such as filtering, transforming, and dimensionality reduction to extract representative feature vectors, which are used to characterize key information in images. Feature extraction helps to represent images as more abstract and distinguishable features, thereby improving the accuracy and robustness of the model in recognizing images. Moreover, appropriate feature processing can reduce data dimensionality, thereby lowering the computational complexity of subsequent classifiers or detectors, thus enhancing processing efficiency. Additionally, good feature processing can extract common information from images, making the model more generalizable and adaptable to various scenarios and transformation conditions in image recognition tasks.

In this paper, deep learning models such as ResNet50, VIT, and Inception v3 are respectively used to extract features from pneumonia images, followed by employing various classification algorithms to derive classification results, thereby comprehensively comparing their effectiveness.

To ensure the quality of the features, this study starts with standardize the images. In the procedure of normalization, Image and ImageOps modules are first imported from the PIL library to preprocess images. The Image.open(img_path) function is used to read image content, resize((150, 150)) function adjusts the size of input images for the deep learning models. ImageOps.colorize(img, white="white", black="black") function is employed to adjust image color. Additionally, during the retraining process, if the deep learning model cannot extract features from the image, which means an error is reported in the process of extracting features, the corresponding data will be discarded in this experiment.

1) ResNet50: As the depth of the convolutional network gets deeper, the extracted features become more advanced and perform better. However, traditional convolutional neural

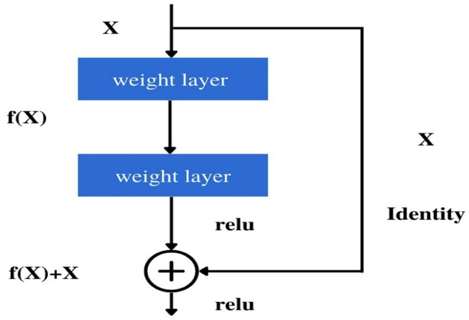

Fig. 4. Basic Structure of residual units

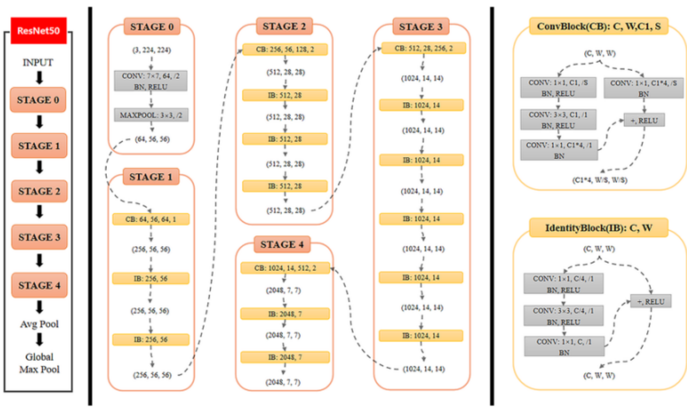

Fig. 5. the Structure of the ResNet50

networks suffer from network degradation, gradient vanishing, and gradient explosion as the depth of layers increases. To solve these problems, the ResNet model was proposed. The key to the ResNet network lies in the residual units in the structure as "Figure 4" shown. [6]

The output signal of the residual block consists of two parts: the output of the jump connection and the output of the residual learning part. The main problem with traditional CNN is that they have to learn the entire feature map, which means they need a large number of parameters. This, in turn, means they are very expensive to train and slow to run. The residuals block is designed so that the network can learn the residuals between the input and the target output, rather than learning the entire map directly. This deep residual learning method has the advantages of fewer parameters, simple backpropagation and less computation. [6] Suppose the input image is x, the output is H(x), and the output after convolution is a nonlinear function of F(x), then the final output is H(x) = F(x) + x. The whole of ResNet consists of multiple residual blocks, each of which contains multiple convolutional layers and bulk normalization layers. The classical ResNet50 convolutional neural network is shown in "Figure 5"

Five stages are required to proceed from input to output in ResNet50. In stage 0, the input is preprocessed. The next four stages consist of 3, 4, 6, and 3 bottlenecks, respectively. Bottleneck residual unit is a kind of convolutional neural network (CNN) specially designed for image recognition. The unit consists of three layers: 1×1 convolution layer, 3×3 convolution layer and 1×1 convolution layer. The 1×1 convolution layer reduces the dimension of the input, while the 3×3 convolution layer is responsible for learning features. The 1×1 convolution layer then recovers the dimensions of the output. [6] Stage 0 contains two layers. The first layer contains sixty-four 7×7 convolution kernels (step size is 2), the BN layer and the RELU activation function. The second layer is the MAXPOOL layer, whose kernel size is 3 x 3 and step size is 2. The last four stages in all consist of ConvBlock(CB) and IdentityBlock(IB). In IB, the input of shape (C,W,W) is x. The three convolutional blocks on the left side are the function F(x), which are summed up and passed through the activation function to get the output. In addition, CB has one more convolutional layer on the right side compared to IB, making its function G(x). The final output obtained is F(x) + G(x). In this experiment, ResNet50 is a model for classifying images of pneumonia.

In this study, ResNet50 served as the feature extractor. It was initialized with pre-trained weights from ImageNet using the ResNet50(weights="imagenet", input_shape=(150,150,3), include_top=False) function. This enabled effective feature learning. The input images were resized to 150x150 pixels with RGB channels. By excluding the top layers (include_top=False), custom classifiers could be added later. Finally, the predict() and flatten() functions were employed to obtain and flatten the image features into a one-dimensional array.

2) VIT: VIT (Vision Transformer) is an image processing model based on attention mechanism, which differs from traditional Convolutional Neural Networks (CNNs). The key innovation of the VIT architecture lies in its utilization of self-attention mechanisms, which allow for capturing long-range dependencies in the input data. This is in contrast to traditional CNNs, where feature extraction is performed through convolutions. The self-attention mechanism enables VIT to effectively process both spatial and positional information within the input sequences, thereby enhancing its ability to capture complex patterns.

In the VIT model, firstly, it divides the input image into blocks and adds positional encoding to each image block to preserve spatial information. Then, these image blocks pass through multiple layers of Transformer encoders, which include multi-head self-attention mechanism and fully connected feed-forward networks, to capture both global features and local relationships in the image. Finally, the output of the last Transformer encoder is mapped to category labels predictions via a fully connected layer. [7] "Figure 6" illustrates its basic structure and principle.

The method of using VIT for feature extraction involves first loading a pre-trained model, then removing the model's top classification layer. Subsequently, the preprocessed images are fed into the VIT model to obtain the output of a specific layer (usually the penultimate layer) as the feature representation of the images. Finally, the extracted features are saved for use as input data for subsequent tasks.

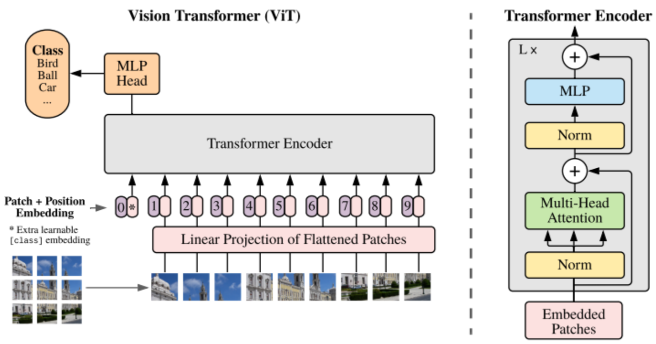

Fig. 6. the Basic Structure and Principle of VIT

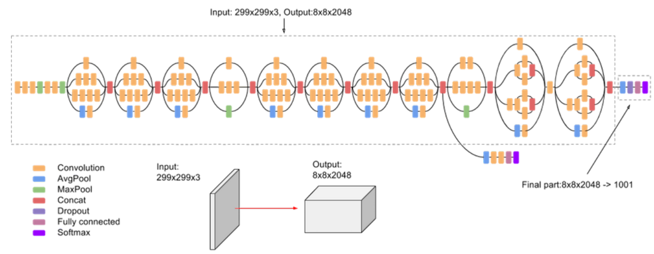

Fig. 7. the Overview Structure of Inception v3

In this study, ViTFeatureExtractor is imported from transformers, and the VIT model is loaded using the from_pretrained("google/vit-base-patch16-224") function. Using the previously imported VIT model, the image data is passed as input to enable the model to extract features from the image. Subsequently, by using the statement $img = img["pixel\_values"]$, the pixel values of the preprocessed image are extracted to facilitate subsequent feature extraction and processing. Here, the value corresponding to the key $pixel\_values$ represents the processed image's pixel values. By using the np.array(img) function, the image is converted into an array, facilitating further processing and analysis of the image data. The flatten() function is used to transform the image from a two-dimensional matrix form into a one-dimensional vector form. Finally, the vector is divided by 255, which normalizes the image. Since pixel values in images typically range from 0 to 255, dividing each pixel value by 255 converts them into floating-point numbers between 0 and 1. This normalization aids in model training and convergence.

3) Inception v3: Inception v3 is a convolutional neural network model developed by Google in 2015 to tackle challenges related to the proliferation of parameters and computational intricacies within deep neural networks. This model comprises both symmetric and asymmetric architectural components, incorporating various operations such as convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Additionally, batch normalization plays a pivotal role in the model, being extensively employed across its architecture and applied to activation inputs. The loss function in Inception v3 is computed utilizing Softmax [8]. "Figure 7" illustrates the overview of the Inception v3.

Inception v3 can serve as a method for feature extraction. It simultaneously extracts features at different scales by employing convolutional kernels of various sizes (e.g., 1x1, 3x3, 5x5) and pooling operations, and concatenates these features in the depth dimension. This enables the capture of more image information across different levels, thereby enhancing the model's performance.

This paper performs feature extraction on standardized images using Inception v3 through the following steps. The transformed images are vectorized using the array(img) function from the numpy library. Subsequently, the function preprocess_input(img_array) from the Keras library in TensorFlow is called. This function preprocesses images for the Inception v3 model, converting them into a format suitable for model input.The invocation of the np.expand_dims(img_array, axis=0) function inserts an additional dimension at the foremost position of the array img_array, thus transforming it into a batch comprising a solitary sample, making these images suitable for feature extraction by the Inception v3 model.

Afterwards, this experiment initializes the foundational portion of an InceptionV3 model using the statement "base_model = InceptionV3(weights = 'imagenet', include_top = False)", while also loading pre-trained weights. The parameter weights='imagenet' signifies the incorporation of pre-trained weights, trained on the extensive ImageNet dataset, thereby augmenting the model's performance. With the parameter include_top=False, the model's top layers (fully connected layers) are omitted, exclusively loading the convolutional segment of the model, which is undertaken specifically for feature extraction, capturing the feature representation of input images post convolutional layer processing, and excluding the final classification segment. Subsequently, using the base_model.predict(img_array) function, the input image data (img_array) is passed through a pre-trained model (base_model) to obtain the image's feature representation within the model. Subsequently, the flatten() function is called to flatten this feature map (multidimensional array) into a one-dimensional vector, facilitating further processing or passing to other modules.

B. Classification Algorithm

The classification algorithm is a core component in the process of image recognition, enabling automatic classification and identification of images by inputting image features into the classifier. Through classification algorithms, automatic classification of a large number of images can be achieved, saving time costs, and improving work efficiency. Classification algorithms in the image recognition process learn the relationship between image features and labels to accurately classify images, making it a crucial part of image recognition. In this study, only two machine learning classification algorithms, namely Random Forest and Logistic Regression, are considered. Logistic Regression and Random Forest models are introduced using the sklearn library. Model training is conducted using the fit() function, and the predict_proba() function is utilized to classify test images, ultimately obtaining the model's classification results and related evaluation parameters.

1) Logistic Regression(LR): Logistic Regression is a widely used statistical method for classification tasks, especially binary classification problems. Despite the term "regression" in its name, logistic regression is actually a classification technique. It utilizes a logistic function, often the Sigmoid function"(1)", to estimate probabilities, thereby predicting the likelihood of an event's occurrence. Logistic regression maps the output of a linear equation to probability values between 0 and 1 through the Sigmoid function [9].

$$\text{Sigmoid Fungtion: g(z)} = \frac{1}{1+e^{-z}} \tag{1}$$

$$\text{input: z} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{2}$$

Where $g(z)$ is the sigmoid function that maps the input z to an output within the probability range and z means the output of the linear equation in logistic regression, typically represented as "(2)", β represents the model parameters and $x_1, x_2, \ldots, x_n$ are input features.

These probability values naturally represent the likelihood of belonging to the positive class (usually denoted as "1"). Given that the probability values are continuous from 0 to 1, they are aptly suited for expressing the probabilities of two categories (positive and negative classes), making it an ideal choice for binary classification problems. The decision boundary of a logistic regression model is linear, meaning it divides the feature space into two parts [9]. For simple binary classification problems, this linear division is typically sufficient. Logistic regression employs maximum likelihood estimation to determine the optimal parameters, a method that is particularly effective in binary classification problems because the objective function (the likelihood function) is a convex function with respect to the model parameters, ensuring that the solution process can find the global optimum. [10]

2) Random Forest(RF): Random Forest is an ensemble learning algorithm extensively used for both classification and regression tasks. It operates by constructing a multitude of decision trees at training time and outputting the mode of the classes (in the case of classification) or mean prediction (in the case of regression) of the individual trees. Random Forest corrects for decision trees' habit of over-fitting to their training set [10].

Key Features of Random Forest:

Ensemble Approach: Random Forest combines multiple decision trees to form a more robust and accurate model. The ensemble approach reduces the risk of over-fitting, a common problem with individual decision trees [9]. "Figure 8" illustrates the principal structure of a random forest.

Bootstrap Aggregating (Bagging): Each tree in the Random Forest is built on a bootstrap sample, a random sample with replacement from the training dataset. This technique introduces diversity among the trees, enhancing the model's accuracy.

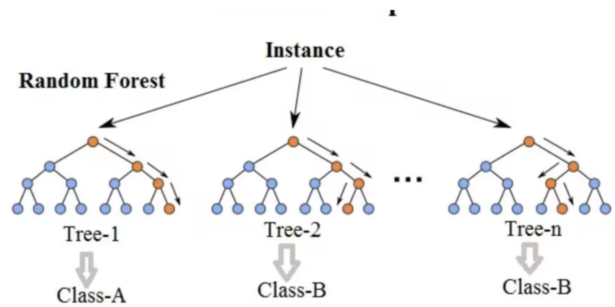

Fig. 8. Random Forest Simplified

Random Feature Selection: When splitting nodes during the construction of trees, Random Forest randomly selects a subset of features at each split. This randomness helps in making the model more robust and less prone to over-fitting.

Versatility: Random Forest can handle both numerical and categorical data. It can also manage missing values and maintain accuracy even when a large proportion of the data are missing [10].

Interpretability: Despite being an ensemble model, Random Forest can provide insights into the importance of each feature in prediction, making it relatively interpretable.

III. RESULT AND EVALUATION

A. Evaluation Method

This paper optimizes the classification model by considering the following criteria comprehensively to ensure the accuracy and stability of models.

1) Confusion matrix and related parameters, including accuracy(CA), F1 score(F1) and recall
2) Receiver Operating Characteristic (ROC) curves and the Area Under the ROC Curve (AUC)

B. Classifier Evaluation: Confusion Matrix and Metrics

To evaluate the performance of these models, this paper introduced the Confusion Matrix and Metrics. Confusion matrix is a table generally used in machine learning and statistical classification to assess the performance of a classification algorithm [11]. And (3) describes its principle. Confusion Matrix summarizes the predicted and actual class labels for a set of data [12]. The matrix has four entries:

1) True Positive (TP): Instances where the model correctly predicts the positive class.
2) True Negative (TN): Instances where the model correctly predicts the negative class.
3) False Positive (FP): Instances where the model incorrectly predicts the positive class (Type I error).
4) False Negative (FN): Instances where the model incorrectly predicts the negative class (Type II error).

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

(3)

According to the above criteria, we classified the test set data using each classification model, and obtained the corresponding confusion matrix for each model. The results are shown in "Fig. 9", which demonstrates the confusion matrices of ResNet50+LogisticRegression (a), ResNet50+RandomForest (b), VIT+LogisticRegression (c), VIT+RandomForest (d), Inception v3+LogisticRegression(e) and Inceprion v3+RandomForest(f) Based on these factors and

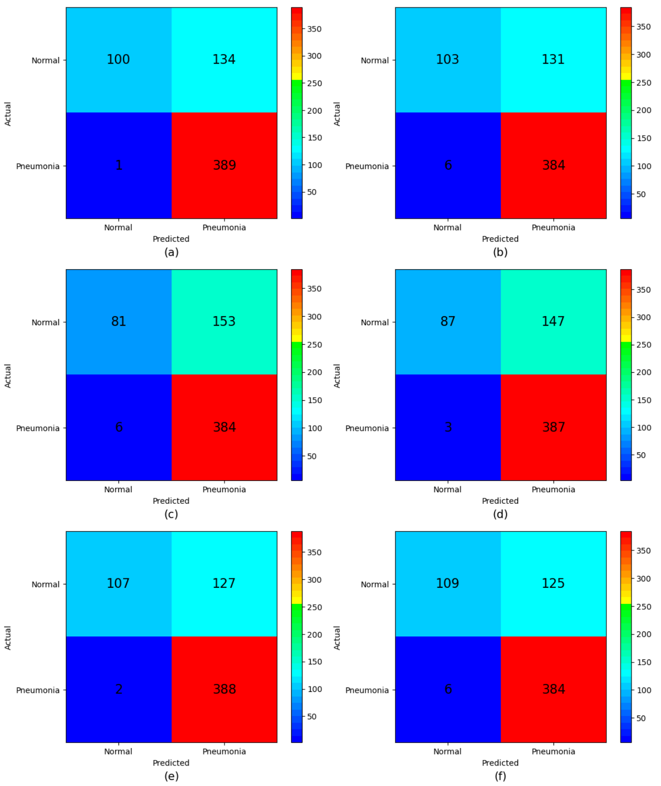

Fig. 9. Confusion Matrix

the formula including Equation (4), Equation (5), Equation (6), and Equation (7), this study calculates the accuracy, precision, recall and F1 score of each classifier model. And the results are presented in "Table. I" and some key parameters are visualized in the "Fig. 10.

According to the "Table. I", this study observes that the accuracy of all models is above 74.5%, with Inception v3 + LR achieving the highest accuracy at 79.33%, and the lowest accuracy among all models is observed with VIT + LR, at 74.52%. Although all models exhibit precision exceeding 93%, the recall rates are comparatively lower, resulting in F1 scores ranging from 50% to 65%. A comparison with results from other studies reveals that, models utilizing feature extraction with ResNet50 and Inception v3 exhibit superior accuracy when compared to shallower deep learning networks like LeNet. However, it is noted that their F1 scores do not match the performance achieved by more complex deep networks such as GoogLeNet. Additionally, upon comparison of "Fig. 10", it is evident that, under the same classification algorithm, the Inception v3 model outperforms ResNet50, which, in turn, surpasses the VIT model among the three feature extraction models. Furthermore, for a given feature extraction model, the Logistic Regression classification algorithm demonstrates superior accuracy and F1 score compared to the Random Forest classification algorithm.

$$\mathrm{CA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}} \tag{4}$$

$$\mathrm{Prec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP+TP}} \tag{5}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{7}$$

TABLE I
DIFFERENT CLASSIFICATION MODELS PERFORMANCE METRICS TABLE

Model	Performance Metrics			
	CA	***Prec***	***Recall***	***F1***
ResNet50+LogisticRegression	0.7837	0.9901	0.4274	0.5971
ResNet50+RandomForest	0.7804	0.964	0.4402	0.6006
VIT+LogisticRegression	0.7452	0.9310	0.3462	0.5047
VIT+RandomForest	0.7596	0.937	0.3718	0.5370
inception v3+LogisticRegression	0.7932	0.9817	0.4573	0.6239
inception v3+RandomForest	0.79.01	0.9478	0.4658	0.6246

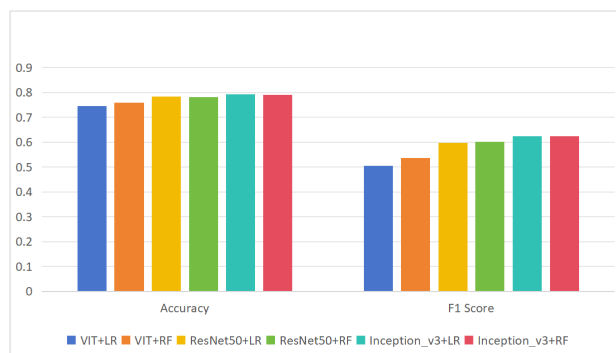

Fig. 10. Accuracy and F1 Score of Different Models

C. *Classifier Evaluation via ROC-AUC*

Given that the ability of a classification model to maintain a consistent level of accuracy across diverse environments, while remaining unaffected by class distribution and the choice of classification threshold, is a pivotal factor in a comprehensive assessment of classification performance [13].

In this study, the ROC-AUC method was employed to evaluate the robustness of the model.ROC-AUC method is widely used in machine learning to evaluate the performance of classification models. ROC-AUC provides insights into the trade-off between sensitivity and specificity across different classification thresholds.

The ROC curve is a graphical representation of the true positive rate (sensitivity) against the false positive rate (1-specificity) for various threshold settings. It is generated by plotting these rates as the classification threshold changes [14]. It uses False Positive Rate (FPR) as the horizontal axis and True Positive Rate (TPR) as the vertical axis to depict the model's performance on positive and negative classes at different thresholds."Fig. 11","Fig. 12" and "Fig. 13"respectively display ROC curves for classification using Logistic Regression and Random Forest algorithms on top of ResNet50, VIT and Inception v3 as feature extraction models.

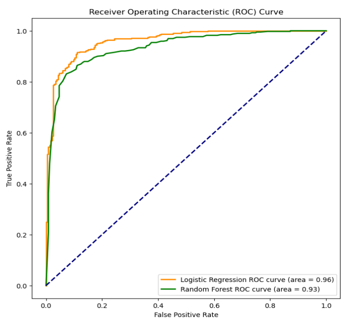

Fig. 11. ROC Curve of the ResNet50

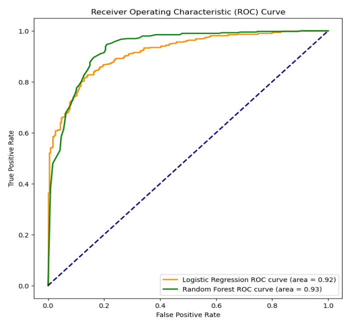

Fig. 12. ROC Curve of the VIT

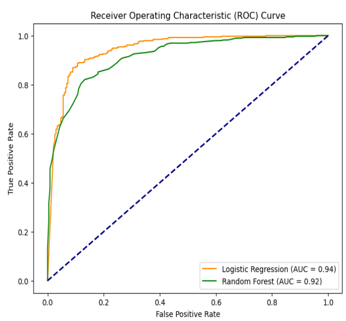

Fig. 13. ROC Curve of the Inception v3

The AUC is the area under the ROC curve. It ranges from 0 to 1, where 0.5 indicates a random classifier and 1.0 represents a perfect classifier. A higher AUC value suggests a better model performance [15].

"Fig. 14" shows the AUC of different models.According to the images, it is evident that the AUC parameters for all models surpass 90%. Notably, among these models, the one utilizing ResNet for feature extraction and LR as the classification algorithm achieves the highest AUC value, reaching 96%. Conversely, models employing VIT for feature extraction with LR as the classification algorithm, and Inception v3 for feature extraction with RF as the classification algorithm, exhibit the lowest AUC values, both at 92%.Based on the AUC parameters of each model, except for the case where VIT serves as the feature extraction model, the AUC value of the LR classifier is higher than that of RF among all models with the same feature extraction. Under the same classifier, the feature extraction performance of the ResNet model surpasses that of Inception v3 and VIT.

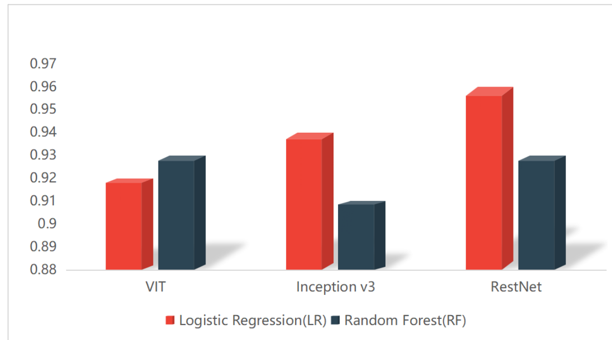

Fig. 14. AUC of Different Models

IV. CONCLUSION

This paper employs deep learning models for feature extraction from pneumonia images and utilizes machine learning algorithms for classification based on these features. After training the models on the training set, the trained classification models are used to classify images in the test set, generating confusion matrices and ROC curves. Comprehensive evaluation of the classification models is conducted by calculating metrics such as accuracy, F1 score, and AUC. Ultimately, the Inception v3+LR model emerges as the one with the highest training set accuracy (79.32%), the second-highest F1 score (0.6239), and the second-highest AUC value (0.94), demonstrating both accuracy and stability. Therefore, through this experiment combining deep learning models with machine learning algorithms, this model is considered the most effective classification model for pneumonia image classification in the medical domain, effectively improving the accuracy and speed of pneumonia diagnosis for doctors.

REFERENCES

[1] E. Avşar, "Effects of Image Preprocessing on the Performance of Convolutional Neural Networks for Pneumonia Detection," 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1-5, Aug 2021.

[2] L. Wang et al., "Automated Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning," IEEE Transactions on Medical Imaging, vol. 38, no. 2, pp. 404-415, Feb 2019.

[3] H. Chen et al., "Deep Learning Techniques for Pneumonia Detection from X-Ray Images: A Review," IEEE Access, vol. 7, pp. 172145-172162, 2019.

[4] Rajaraman S, Candemir S, Xue Z, Alderson PO, Kohli M, Abuya J, Thoma GR, and Antani S, "A Novel Stacked Generalization of Models for Improved TB Detection in chest radiographs," Annu Int Conf IEEE Eng Med Biol, pp. 2073-2073, Jul 2018.

[5] Z.-P. Jiang, Y.-Y. Liu, Z.-E. Shao, and K.-W. Huang, "An Improved VGG16 Model for Pneumonia Image Classification," Applied Sciences, vol. 11, no. 23, p. 11185, Jan 2021.

[6] M.Shafiq, and Z.Gu, "Deep Residual Learning for Image Recognition: A Survey," Applied Sciences, vol.12, no.18, pp.8972, Sep 2022.

[7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby,"An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv preprint arXiv:2010.11929, Jun 2021.

[8] Google Cloud. "Inception-v3 Advanced Usage - TPU Documentation," Cloud TPU, Feb 2024.

[9] Breiman L, "Random Forests," Machine Learning, 45, pp. 5-32, Oct 2001.

[10] Liaw Andy, and Wiener Matthew, ”Classification and Regression by randomForest,” R News, vol.2/3, pp.18-22, Dec 2002.
[11] Gareth James, Daniela Witten, Trevor Hastie, and Rob Tibshirani, ”An Introduction to Statistical Learning. Springer,” Springer Texts in Statistics, vol.103, 2013.
[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, ”The Elements of Statistical Learning.” Springer Series in Statistics, 2009.
[13] David M. W. Powers, ”Evaluation: From precision, recall and F1 to ROC, informedness, markedness and correlation,” Journal of Machine Learning Technologies, pp. 37-63, Oct 2020.
[14] Tom Fawcett, ”An introduction to ROC analysis,” Pattern Recognition Letters, Vol. 27,pp. 861-874, 2006.
[15] Hanley JA, and McNeil BJ, ”The meaning and use of the area under a receiver operating characteristic (ROC) curve,” Radiology, 143(1), pp.29-36, Apr 1982.