Deep Learning on Medical Computer Vision

Introduction

Microscopy is essential in medicine, revealing critical biological details. Computerized image analysis enhances research and clinical applications like robotic surgery and diagnosis, demanding accuracy and speed due to their impact on health. Machine learning uncovers vital image features, aiding medical staff by improving understanding and treatment outcomes.
The main aim of this article is to analyze and research the application of the algorithms we learned in class based on our task images and background. The following content mainly analyzes the structure of the algorithms, optimizes the model through data preprocessing and hyperparameter tuning, and compares algorithm running time, algorithm interpretability, and algorithm accuracy and robustness.

Data

Data Description

There are 4 datasets in this data: X-train, y_train, X_test and y_test. The training set has 13673 rows of data, the dimension is 28*28*3, and it is RGB data. y has 8 classes
The dataset used is from BloodMNIST, containing 28x28 color images of normal blood cells, each highlighting a specific cell type.

Data Exploration

In order to analyze the data distribution of the training and test set, I draw a histogram.

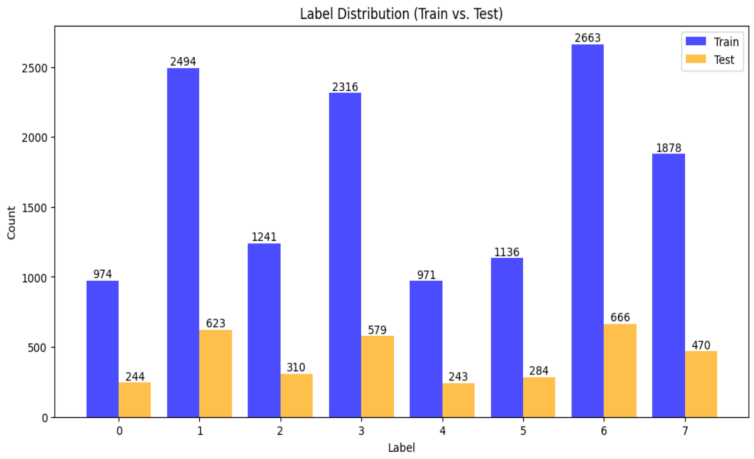

Figure 1. Distribution of training and test set

As can be seen, the data volume of the test set is smaller, but the distribution characteristics are roughly the same as the training set. Also, the dataset is imbalanced.
I randomly selected 4 pictures from each class to see how the pictures looked.

Figure 2. Image examples

Figure 2 shows the features across different classes are different, but similar within the same class. Attributes like brightness, contrast, and cell size, and shape differ across categories. Some images show incomplete cell nuclei at the edges.

Challenges

Incomplete images challenge our image data processing and analysis. Similarities in color, shape, and structure also complicate classification. Classifying cell nuclei requires expertise, posing challenges for non-medical professionals.

Data Preprocessing

Observing the data, I first created a validation set from the training set for model validation and parameter tuning. Next, I standardized the images to improve algorithm convergence.

Data Augmentation

For image data, special preprocessing like data augmentation (flipping, adjusting brightness, hue, saturation, contrast, adding noise) can expand limited data. However, not all methods suit our dataset, as explored in Table 1. Ultimately, only left-right and up-down flipping were used as our augmentation strategies.

Table 1. Validation of data augmentation on our dataset. Flip 1 and 2 denote left-right and up-down flipping. The accuracy is based on CNN.

	Accuracy	Training Time	Test Time
rgb	0.9143	1min 34s	1.21s
rgb+normalization	0.9228	1min 40s	1.34s
rgb+normalization+flip1	0.9421	2min 9s	1.26s
rgb+normalization+flip1+flip2	**0.9471**	2min 24s	1.04s
rgb+normalization+flip1+flip2+brightness	0.9280	3min 47s	1.29s
rgb+normalization+flip1+flip2+Gaussian noise	0.9418	3min 57s	1.05s
rgb+normalization+flip1+flip2+hue	0.9263	4min 26s	1.12s

rgb+normalization+flip1+flip2+saturation	0.9351	4min 49s	1.02s
rgb+normalization+flip1+flip2+contrast	0.9333	3min 40s	1.31s

Methods

Model 1 - Fully Connected Neural Network

The Principle of MLP:

MLP is a multi-layer perceptron that handles complex non-linear problems like computer vision and natural language processing using multiple linear layers, as shown in Figure 3. It transforms i inputs into n outputs through multiple linear transformations [1].

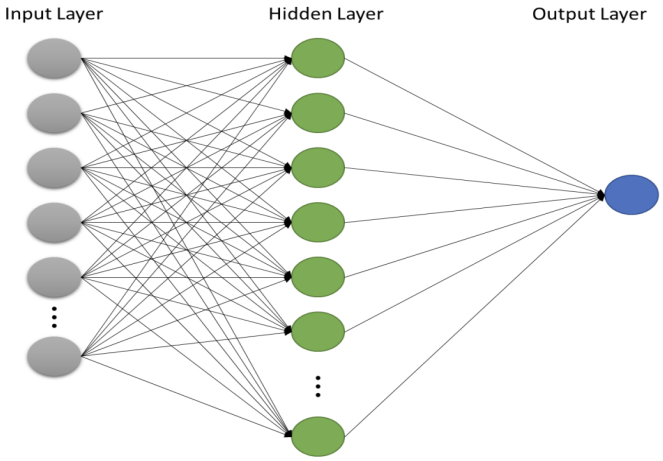

Figure 3: The design of MLP.

The calculation of output, input, and bias variables is as follows:

$$C_i = \sum_{i=1}^{n} W_{im} X_i + B_m \quad (1)$$

where W_{im}, X_i , *and* B_m refers to the weights, input variables, and bias variables.

Our task involves image classification. Images, that are complex, are not effectively processed by linear models. MLP extracts non-linear features, representing images better and benefiting downstream tasks like classification.

The input RGB image is flattened to match MLP's input. After transformation, the output Y dimension is K, representing 8 output categories in our task.

$$\sigma(Z) = 1/(1 + e^{(-z)}), \quad (2)$$

$$Y=\sigma(f(X)), \quad (3)$$

where $f(X)$ is MLP, $\sigma(\bullet)$ is activation function, which is $sigmoid$ function.

The MLP model itself may be overfitted, but dropout can prevent overfitting problems.

Strength: Suitable for complex data structure: MLP can handle nonlinear problems. For complex data types such as images, MLP can automatically learn their features and complete classification. The MLP model is a multi-layer structure that can process and learn high-level abstract features. The MLP model can learn by itself and adapt to complex structures without defining model features in advance.

Weakness: MLP's unsuitability for image processing lies in its one-dimensional input requirement. Images, being two-dimensional with pixel spatial relationships, must be flattened to one dimension, destroying these inherent spatial relationships. In order to process complex image data, MLP requires more parameters, so that it requires a longer calculation time. Poor interpretability. Because the model structure is complex, it is difficult to understand the internal logic and learned features.

Architecture Design

I designed an MLP model for our task. It starts with a flattening layer, converting 2D images to 1D. Then, it passes through N-dense layers with M neurons and an activation function $\sigma(\cdot)$, interspersed with Dropout layers to reduce overfitting, discarding 10% and 50% of neurons. Finally, a 10-neuron output layer with Softmax activation outputs a probability distribution over 10 categories.

Model 2 - Convolutional Neural Networks

The Principal of CNN:

CNNs are similar to MLP but convolution to extract features. Comprising convolution, pooling, and fully connected layers, CNNs extract local image features, select significant pixels through pooling, and output desired results via the fully connected layer, akin to traditional neural networks.

$$S(i,j) = \Sigma_m \Sigma_n I_{(i-m,j-n)} * K_{(m,n)} \quad (4)$$

Strength: MLP struggles with two-dimensional image data, unlike CNN, which preserves spatial information through convolution, matching image data better. CNNs effectively extract image features, transferring edge pixel characteristics towards the center as model depth increases, allowing feature maps to represent global information in reduced dimensions. Translation invariance: The translation invariance of CNN means that it can identify the position of the object, no matter where the object is in the image, it can be recognized. This is useful when working with objects of different scales and locations. CNN can handle complex image data while maintaining high accuracy. CNN can automatically learn image features through convolutional layers and pooling layers without manually designing feature extractors, which can improve classification performance.

Weakness: CNN computations are intricate, producing two-dimensional feature maps that can't directly yield classification results, necessitating a classifier like MLP for this purpose. CNN requires a large amount of data. If the amount of data is too small, it is easy to overfit or fail to converge. Parameter adjustment is complicated and the amount of calculation is too large[2].

Architecture Design

I designed a CNN model for our task. It includes N blocks, each with three convolution layers followed by an activation function and average pooling, reducing spatial dimensions by 50%. channel numbers double after each block. Post-convolution, features are flattened, and passed through an MLP with a 128-dimensional hidden layer and Softmax activation, outputting an 8-category probability distribution. A 50% Dropout layer before the MLP reduces overfitting risks.

Model 3 - PCA+SVM

The Principle and Reason for Choosing PCA:

Images are high-dimensional and noisy. PCA linearly projects this data to lower dimensions, removing noise and retaining meaningful classification features. With 13,673 training data points and 2,352 features (width * height * 3), satisfying n>=m, the original image data X can be represented more succinctly[3].

By PCA, original data X can be transformed to

$$X_{reduced} = \lambda_1 u_1 v_1 + \lambda_2 u_2 v_2^T + \cdots + \lambda_k u_k v_k^T \quad , \tag{6}$$

The principles and reasons for selection of SVM:

SVM maximizes the decision boundary margin for more accurate classification. Originally for binary classification, its calculation is as follows:

$$f(X) = sign(W * X + b) \tag{7}$$

W and b are the feature vector and bias, respectively, forming the hyperplane.

Training multiple SVMs extends its application to multi-class classification. Regularization further enables SVM to nonlinear decision boundaries, enhancing its suitability for images.

Strength: The data after PCA reduces the dimension, which can simplify the process and time of SVM image processing. Moreover, PCA can avoid overfitting. The PCA can help extract the main features of the image and distinguish primary and secondary features.

Weakness: The dimensionality reduction process of the PCA model may lose information of the image data, some of which may be important. Using PCA for classification sacrifices SVM model interpretability, making it unclear which pixels and channels most influence the classifier. Sensitive to outliers and noise, more affected.

Architecture Design

I designed a PCA+SVM model for our task. Initially, PCA, configured to retain N principal components, reduces image data dimensions, fitting on training and applied on validation and test data. An SVM classifier is then trained using PCA-processed training data and subsequently predicts the validation and test data.

Model 4 - Random Forest

The Principle of Random Forest:

Random Forest is an integrated method that randomly samples data and builds multiple decision tree models through bootstrap. At the same time, only a part of the features are used to build the model each time to enhance the generalization of the model. For inference, it inputs the input data X into the constructed N decision trees and obtains N classification results Ypred. Through voting, the result with the most votes is selected as the final prediction y_{pred} [5].

The Reason for Choosing Random Forest:

Image data is typical nonlinear data, and the random forest model can handle nonlinear problems very well. For high noise, Random Forest can reduce the interference of noise on classification results by randomly sampling features.

Strength: When dealing with image problems, the random forest model is insensitive to outliers and noise and has good robustness. Strong generalization ability. Because random

forest is a set algorithm composed of multiple decision trees, it can handle unfamiliar pairs of data very well. Suitable for processing large amounts of complex data.
Weakness: The random forest model is complex and consumes a lot of computing resources. The training time will be very long, Random Forest will not be as interpretable as the decision tree. There are many parameters, and the model parameter adjustment will be more complicated.
Architecture Design
I designed a random forest model for our task. Image data is reshaped into 2D arrays and flattened for model input. A random forest classifier, with N estimators and a set random state for result consistency, is trained using this data. The model then predicts validation data, calculating and outputting accuracy.

Comparison of 4 Algorithms

CNN: Considering the 2D nature of the data set given by the task, although the running speed is very slow, the translation invariant performance of CNN is best adapted to image data and can greatly improve the performance of the model.
MLP: Compared with CNN, MLP's inference time is faster, but MLP ignores the 2D features of image data and cannot extract the spatial information of the image, making MLP's performance slightly lower than CNN.
PCA+SVM: Compared with the deep learning model, PCA+SVM has faster inference speed, but it cannot fully extract the features of the image, resulting in inferior performance than the deep learning algorithm.
Random Forest: Random forest can avoid overfitting of the model through the integration method. However, due to the nature of the decision tree, it cannot extract the characteristics of the image data, resulting in that it cannot obtain a satisfactory performance [6, 7].

Hyperparameter tuning

To analyze parameter impacts on the model, I adjusted each using the control variable method, enhancing interpretability and understanding of the model structure. Using only validation data for hyperparameter tuning, I identified optimal parameters and analyzed their effects on model accuracy.

1. Ablation experiment on MLP

I added a 0.5 dropout layer before the MLP output to prevent overfitting and manually tuned parameters: depth, neuron number, and activation function. Depth influences complexity and overfitting risks, neurons affect model capacity, and activation functions like ReLU add nonlinearity. Keeping neuron numbers and activation constant, I varied the depth between 1 and 6, obtaining the following results:

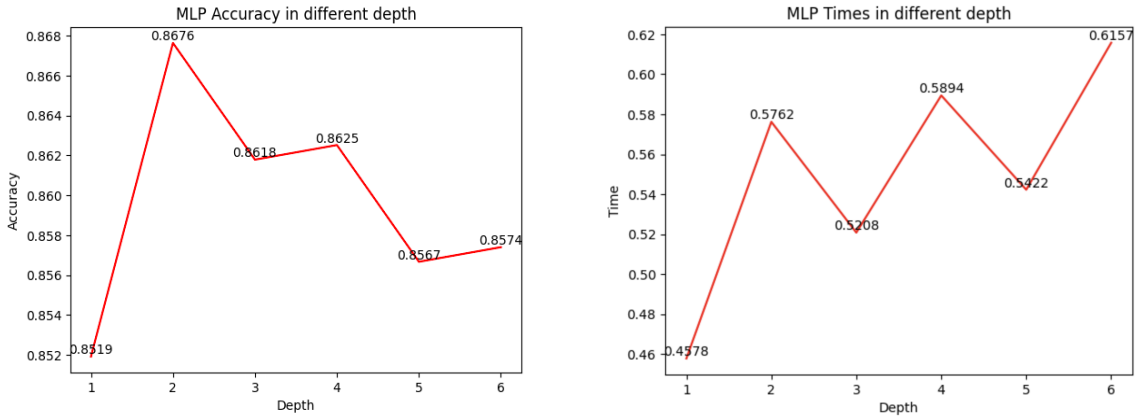

Figure 4. The performance and efficiency of MLP with different depth.

As can be seen from the above figure, the accuracy of the MLP model first increases and then decreases as the depth increases, so the best depth is 2.

Secondly, keep the depth and the number of neurons unchanged, and change the activation function: [‘relu’, ‘tanh’, ‘leaky_relu’’], the following results are obtained:

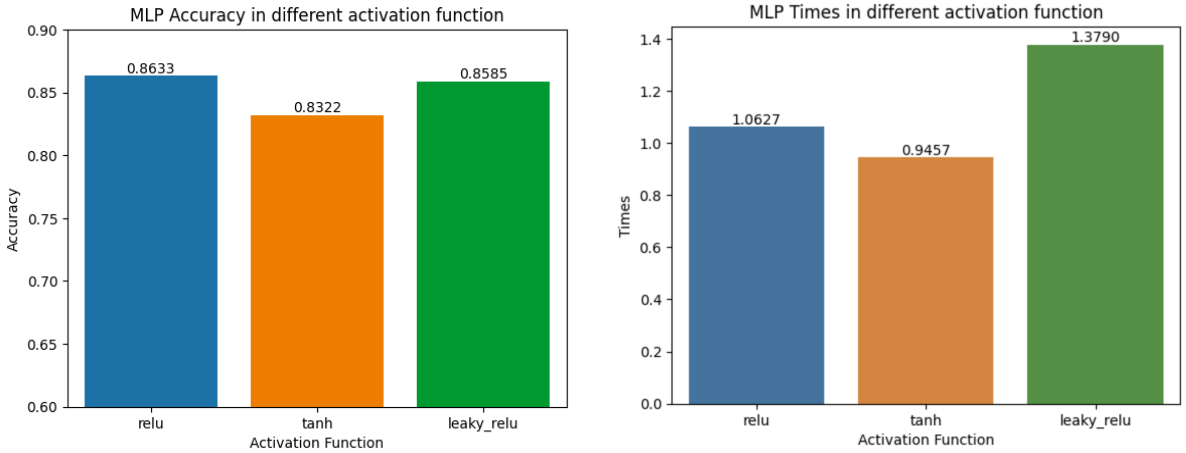

Figure 5. The performance and efficiency of MLP with activation function.

We can see from the figure above, that the activation function with the highest accuracy is "relu".

Finally, keep the depth and activation function of the MLP model unchanged, adjust the number of neurons in the model [64, 128, 256, 512], and obtain the following results:

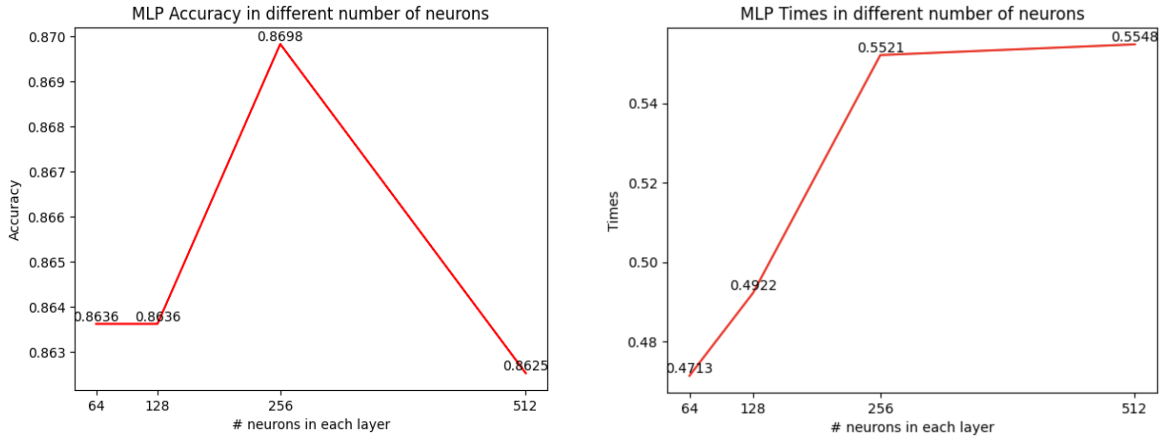

Figure 6. The performance and efficiency of MLP with different numbers of neurons.

We can see from the figures above, that the number of neurons with the highest accuracy is 256.
To sum up, the best parameters of the MLP model are: [‘2’, ‘relu’, ‘256’]

2. Ablation experiment on CNN

I added a 0.5 dropout layer before the CNN output to prevent overfitting. After ablation experiments in data enhancement, the best parameters were manually adjusted in the CNN model. Parameters such as depth, neuron numbers, kernel size, and activation function were individually tuned. Kernel size, a crucial parameter, refers to the filter size used in convolution operations.
First, I controlled the number of channels, kept the kernel size and activation function unchanged, modified the model to depth [3, 6, 9], and obtained the following results:

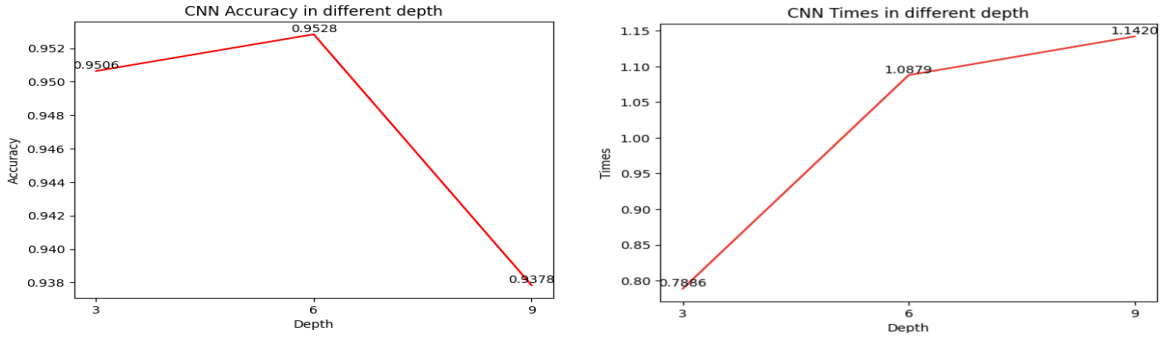

Figure 7. The performance and efficiency of CNN with different depths.
As can be seen from the above figure, as the depth increases, the accuracy of the CNN model first increases and then decreases, so the best depth is 6.
Secondly, I controlled the depth and number of channels of the CNN model, kept the activation function unchanged, and adjusted the parameter kernel size [3, 5, 7], and the results were as follows:

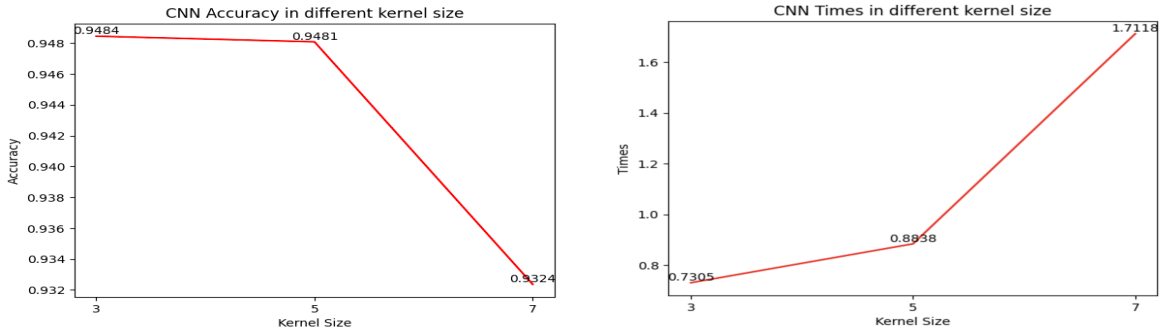

Figure 8. The performance and efficiency of CNN with different kernel size.
As can be seen from the above figure, the best kernel size is 3. Because our image dimensions are small, a smaller kernel can extract all the information from the image.
Then, I controlled the depth of the CNN model, the number of channels, and the kernel size. I adjusted the parameter activation functions: ‘relu’, ‘tanh’, and ‘leaky relu’. The results are as follows:

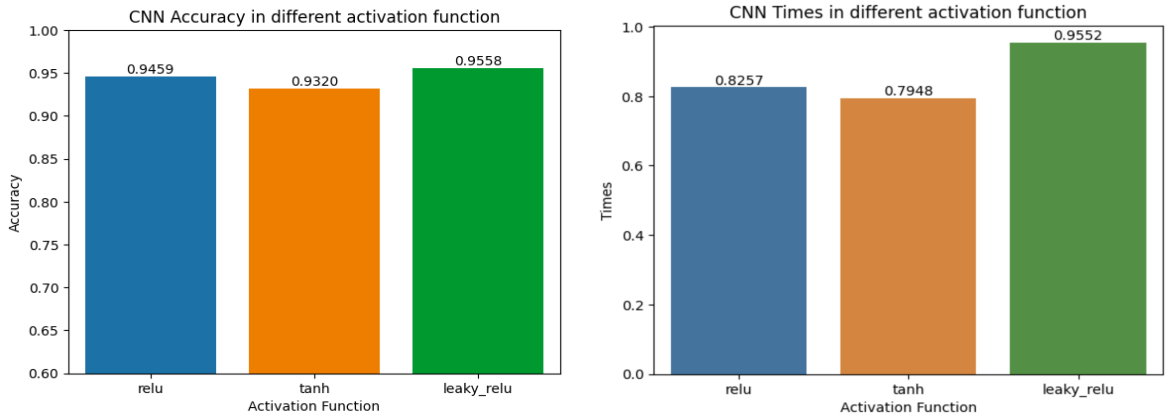

Figure 9. The performance and efficiency of CNN with different activation function.

As can be seen from the above figure, the best activation function is ‘leaky relu’.

Finally, I controlled the depth and kernel size of the CNN model, kept the activation function unchanged, and adjusted the number of parameter channels. The results are as follows.

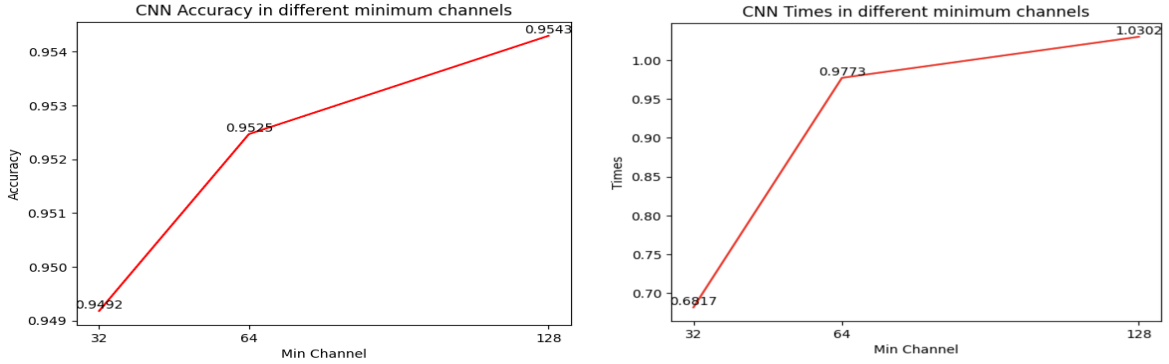

Figure 10. The performance and efficiency of CNN with different minimum channels.

We can see that the number of channels with the highest accuracy is 128.

To sum up, the best parameters of the CNN model are: [‘6’, ‘3’, ‘leaky relu’,’128’]

3. Ablation experiment on PCA+SVM

In this model, I tuned parameters: 'pca__n_components', 'svc__C', and 'svc__kernel' individually. 'pca__n_components' determines the principal components retained by PCA, a dimensionality reduction technique. 'svc__C' is SVM’s regularization parameter, controlling model complexity. 'svc__kernel' specifies the SVM’s kernel function type.

First, I controlled and kept svc__C and svc__kernel unchanged, modified pca__n_components [200, 400, 800], and obtained the following results:

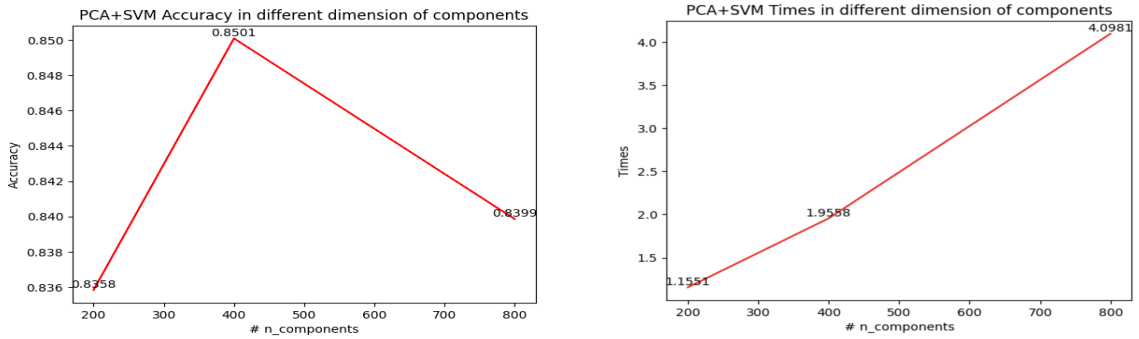

Figure 11. The performance and efficiency of PCA+SVM with different dimensions of components.

We can see from the above figure, that as the pca__n_components increases, the accuracy of the CNN model first increases and then decreases, so the best pca__n_components is 400.

Secondly, I controlled the pca__n_components and svc__kernel unchanged and adjusted the parameter svc__C [0.01, 0.1, 1, 10], and the results were as follows:

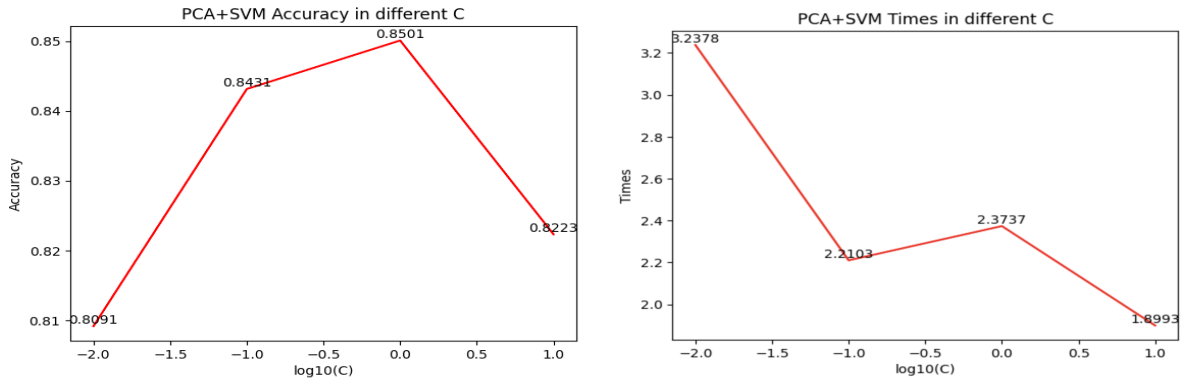

Figure 12. The performance and efficiency of PCA+SVM with different C.

As can be seen from the above figure, the best svc__C is 0.1.

Finally, I controlled the pca__n_components and svc__C unchanged, and adjusted the parameters svc__kernel: 'linear', 'rbf', and 'sigmoid'. The results are as follows.

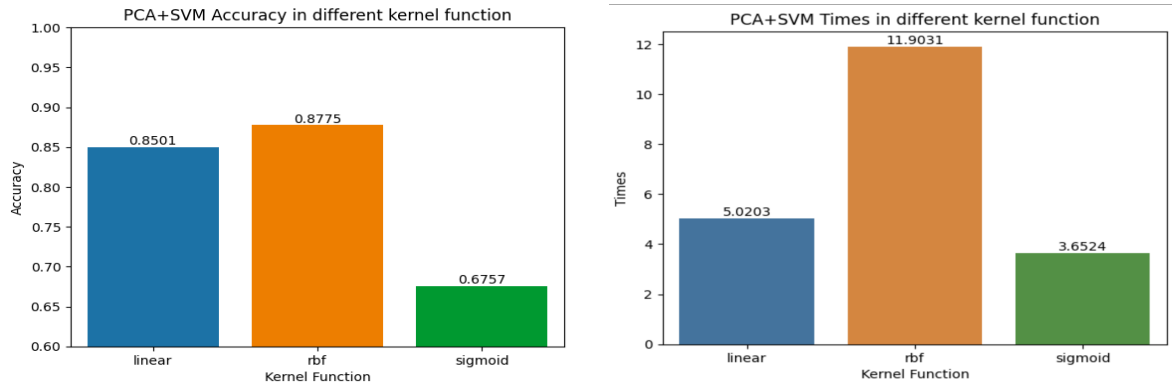

Figure 13. The performance and efficiency of PCA+SVM with different kernel functions.

We can see that the best svc__kernel with the highest accuracy is rbf.

To sum up, the best parameters of the CNN model are: [‘400’, ‘0.1’, ‘rbf’]

4. Ablation experiment on random forest

In this model, I tuned parameters: 'n_estimators', 'max_depth', and 'criterion'. 'n_estimators' defines the forest's tree count, affecting model complexity and computational cost. 'max_depth' limits tree size, and 'criterion' determines the split quality evaluation function, influencing feature and split point selections in tree building.

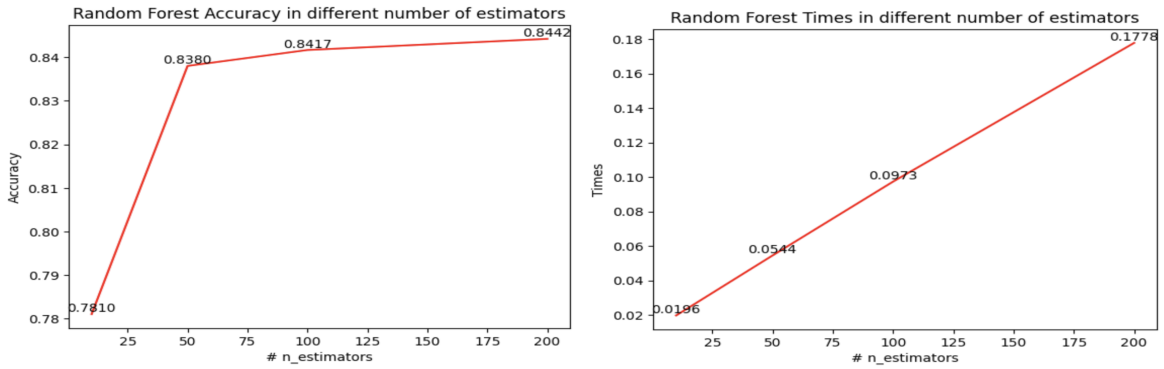

Figure 14. The performance and efficiency of random forest with different estimators.

We can see from the above figure, that as the n_estimators increase, the accuracy of the random forest model will rise greatly first, and then, it will increase slowly. However, its inference efficiency also decreases. Finally, considering the efficiency of the model, we choose 200 as the number of estimators.

Secondly, I fixed n_estimators as 200 and tuned the max depth of trees.

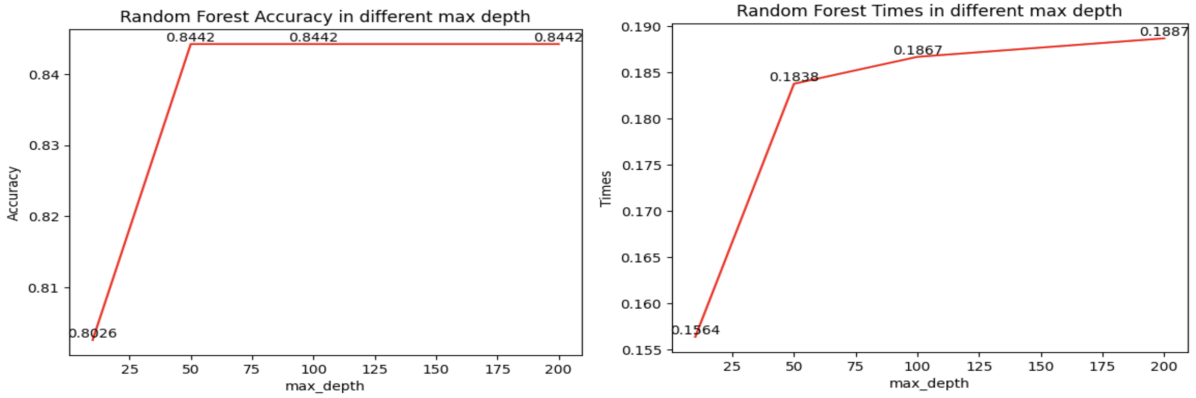

Figure 15. The performance and efficiency of random forests with different max depths.

As can be seen from the above figure, with the increase of max depth, the performance can be improved greatly, while with a further increase, it will not change, which may be because the trees have achieved their depth and will not be affected by these parameters. Meanwhile, its running time will also rise with the increase of max depth. Thus, we use 50 as max depth as last.

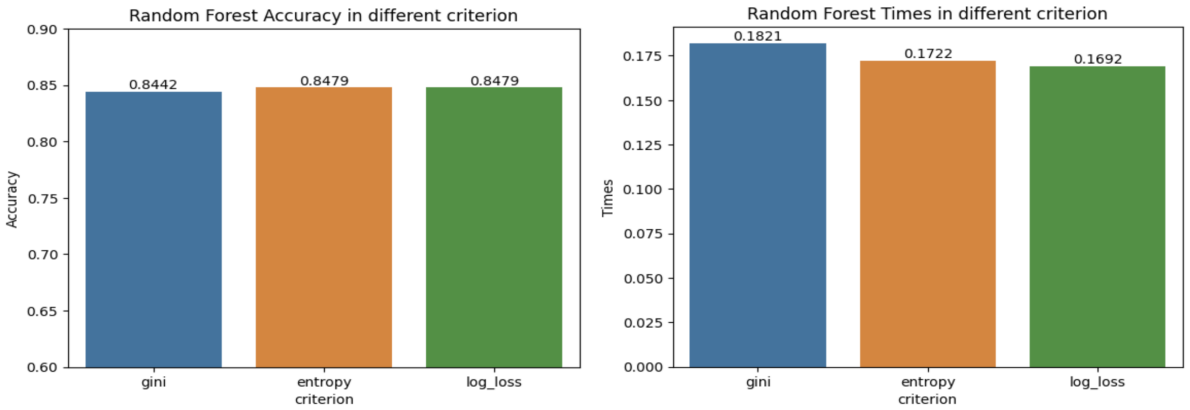

Figure 16. The performance and efficiency of random forest with different criteria.

Different criterions will cause different accuracy. It can be seen the performance and efficiency of gini, entropy, and log_loss are not too much different. Among them, entropy performs best. Thus, it is used as the final criterion.

Results

Table 2. Comparison of different methods.

model	accuracy	macro f1	runtime (s)
MLP	0.8488	0.8283	0.4847
CNN	0.9491	0.9408	1.5787
PVA+SVM	0.8663	0.8431	8.9472
Random Forest	0.8321	0.7997	0.2336

Results vary across models; accuracy initially increases, then decreases. The CNN model performs best with 0.9459 accuracy, while the random forest is lowest at 0.8321. MLP and PCA+SVM show average performances, aligning with prior predictions. Due to the image data's 2D nature, CNN models adapt best, yielding superior results, while the inherent traits of decision trees limit the random forest model. In terms of runtime, PCA+SVM is slower due to its two-stage nature, while the random forest is the fastest, and CNN takes the longest.

Conclusion and future work

conclusion: Comparing performance, overfitting, runtime, and interpretability, machine learning models offer better interpretability but lower performance. Deep learning excels in image data but lacks in learning efficiency and interpretability, and is prone to overfitting due to our task's smaller dataset size.

study limitations: Due to the relatively small size of our data set and the small dimensions of the images, for deeper neural networks, our data cannot meet the training requirements and cannot explore the deeper layers of the model.

Future work suggestions: In future work, we can use larger images and more data, such as ImageNet [8], to explore the limits of the model.

Reflection

Completing this assignment, I learned about neural networks, distinguishing between machine learning and deep learning in processing image data. I realized that lecture knowledge needed to be improved, and more practical projects are necessary for improvement.

Reference

[1] Abdel-aziem, A. H., & Soliman, T. H. A Multi-Layer Perceptron (MLP) Neural Networks for Stellar Classification: A Review of Methods and Results.
[2] Firdaus, M., & Arief, M. R. (2023). Impact of Data Augmentation Techniques on the Implementation of a Combination Model of Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) for the Detection of Diseases in Rice Plants. *Journal of Scientific Research, Education, and Technology (JSRET)*, *2*(2), 453-465.
[3] Ning-min, S., & Jing, L. (2015). A literature survey on high-dimensional sparse principal component analysis. *International Journal of Database Theory and Application*, *8*(6), 57-74.
[4] Al-Saqqar, F., AL-Shatnawi, A. M., Al-Diabat, M., & Aloun, M. (2019). Handwritten Arabic text recognition using principal component analysis and support vector machines. *International journal of advanced computer science and applications*, *10*(12).
[5] Zhang, J., Li, M., Feng, Y., & Yang, C. (2020). Robotic grasp detection based on image processing and random forest. *Multimedia Tools and Applications*, *79*, 2427-2446.
[6] Wen, S., Kurc, T. M., Hou, L., Saltz, J. H., Gupta, R. R., Batiste, R., ... & Zhu, W. (2018). Comparison of different classifiers with active learning to support quality control in nucleus segmentation in pathology images. *AMIA Summits on Translational Science Proceedings*, *2018*, 227.
[7] Klibi, S., Mestiri, M., & Farah, I. R. (2021, July). Emotional behavior analysis based on EEG signal processing using Machine Learning: A case study. In *2021 International* Congress of Advanced Technology and Engineering (ICOTEN) (pp. 1-7). IEEE.
[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.