

Interpretative medical image analysis -based on CNN, image moment and logistic regression

Boyan Tang, Peng Xiao, Yuhan Zhao, Haoran Lu, and Xuanhao Ren

Abstract—Medical image is a vital part of the medical field. With the development of computer technology, computer vision is gradually applied to medical image. Most of the medical image is used deep learning and convolutional neural networks (CNN). However, the predominant utilization of deep learning methodologies and CNN, presents challenges concerning explainability, especially in the context of medical image. This paper introduces an interpretative model, aimed at addressing these challenges by combining a CNN-based classification with Hu's moment transforms and traditional machine learning classifiers such as logistic regression. Ultimately, the model demonstrates results with a reported accuracy of 93%, in spite of being slightly lower than the accuracy of 96% achieved by VGG network. Nonetheless, the model's performance is acceptable, particularly considering its emphasis on interpretability. After conducting a small survey with professional doctors from Peking University Shenzhen Hospital, it was found that the model proves greater reliability in actual medical clinical trails.

Keyword—medical image, CNN, Hu's image moment, logistic regression

I. INTRODUCTION

Medical image is an important part of the medical field, which helps doctors to observe and analyze the internal structure of patients for diagnosis. With the development of computer technology, computer vision is gradually applied to medical image, which allows doctors to analyze images and make diagnoses more quickly and accurately. This paper we choose brain image to analysis, which is a complex and challenging task. The complexity of brain structure and the diversity of diseases make accurate analysis of brain images essential.

However, nowadays, most medical images use deep learning method and CNN, which have certain limitations. As Esteva et al. demonstrate, while deep learning models can achieve dermatologist level accuracy in cancer classification, the black-box nature of these models requires transparent and interpretable approaches to ensure they are adopted in clinical practice. ^[1] Although CNN has huge potential in medical image, its black-box mechanism makes it difficult to explain the predictive processes behind it, especially when it comes to brain tumor prediction ^[2]. Despite some advances in model interpretation, such as multi-layer convolutional sparse coding (ML-CSC), the interpretability of models still faces significant challenges when dealing with multimodal data for medical image segmentation ^[3]. If these models are applied to the

decision-making process of healthcare diagnosis and treatment, the lack of explainability may increase the risk of decision-making.

Therefore, this paper constructs a model that firstly pre-processing the data through normalization and inverse method, next use CNN and self-attention mechanism to classify the images to three sides (front, behind and besides), using Hu's moment to feature extracting and ensemble learning to vote, then use logistic regression to classification, finally evaluate and optimize the model.

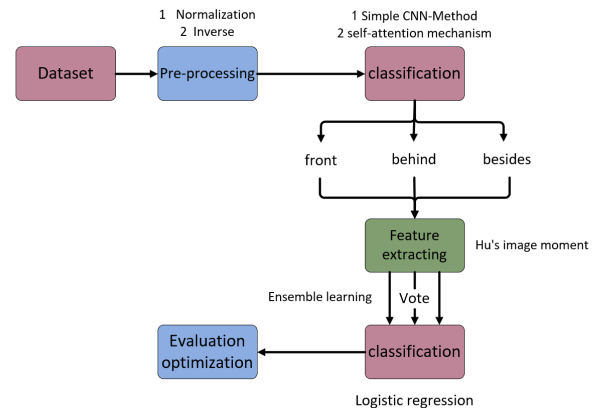


Fig. 1. Workflow

II. BACKGROUND

A. CNN

Since the discovery of convolutional neural networks, many experiments and papers have demonstrated their effectiveness. This paper also uses convolutional neural networks as the cornerstone of our view classifier. The convolutional neural network mainly employs sliding window feature acquisition, then uses the maximum pooling layer to reduce the useless data, and finally, the application of full-connected layer to fit classifier.

However, despite the effectiveness of CNN, developers suffer from limitations in extracting features using sliding window methods, as they don't know what specific features are. ^[4] This results in a lack of interpretability, as developers are unable to accurately extract features. In our proposed model, we only use CNN as a simple solution on view detection.

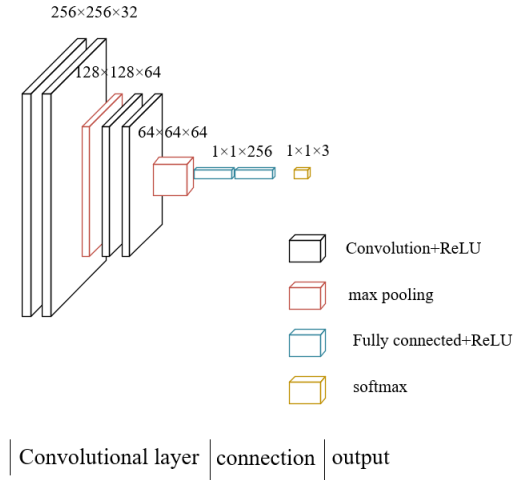


Fig. 2. CNN flowchart

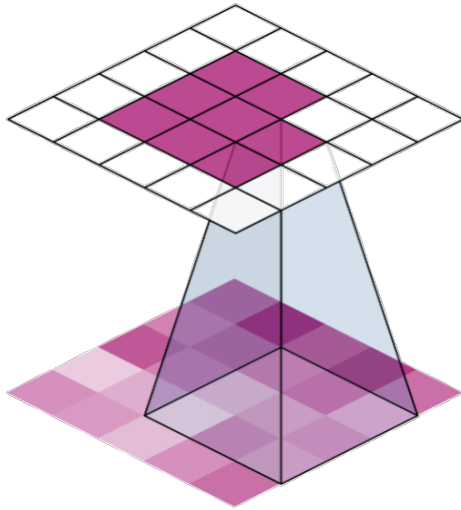


Fig. 3. Sliding window

B. Self-Attention Mechanism

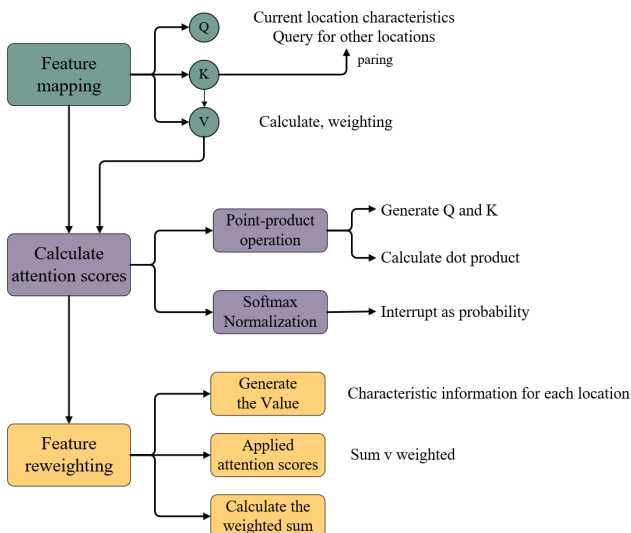


Fig. 4. Self-Attention Mechanism flowchart

In medical image analysis, self-attention mechanism can help the model focus on areas of the image that are most critical to diagnosis. For tumor detection tasks, the self-attention mechanism enables the model to identify and locate tumors more accurately in complex contexts. This not only improves the diagnostic accuracy of the model, but also improves its generalization ability.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

1) Feature mapping

In a standard CNN, the feature maps (the output of the convolution layer) usually go directly to the next convolutional layer or the pooling layer. In the self-attention model, the feature graph is first mapped to three different vector spaces, forming three components: query (Query, Q), Key (Key, K) and value (Value, V).

Query (Q) represents the feature of the current location and is used to query the feature information of other locations. Key (K) pairs with the query to calculate the importance of the features related to the current query. Value (V) contains the actual feature information for each location and is weighted by the calculated attention score.

These vectors are usually obtained by linear transformations of a series of original feature graphs.

2) Calculate attention scores

In the self-attention mechanism, the calculation of attention scores is the critical step to determine the contribution of various parts of the input image to the final output. This mechanism allows the model to focus dynamically on the most important features. The process is explained in detail below:

(a) Point-product operation:

① Generate Query and Key

For a given input feature graph, we use different convolution layers or linear layers to generate Q and K matrices respectively from the same feature graph.

② Calculating the dot product

The dot product is obtained by multiplying each vector in the query matrix with each vector in the key matrix.

(b) Softmax Normalization

After the dot product operation, the original attention score matrix obtained needs to be further processed and converted into a valid probability distribution.

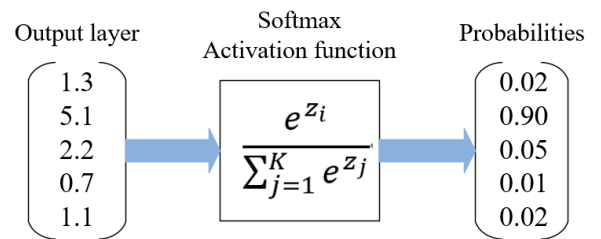


Fig. 5. Softmax activation flowchart

The Softmax function is applied to the attention score matrix for each row. For each element A_{ij} in the matrix A:

$$\text{Softmax}(A_{ij}) = \frac{e^{A_{ij}}}{\sum_k e^{A_{ij}}} \quad (2)$$

Here k traverses all columns, representing the normalization for all elements of row i . After such processing, the sum of all elements in each row is 1, and each element ' A_{ij} ' indicates the relative importance of the j th position when processing the i th position.

3) Feature reweighting

Feature reweighting involves three main steps.

① generate value

Like queries and keys, values are generated by applying a convolution or linear layer to the input feature graph. Value represents the actual feature content of each location and is weighted according to the attention score associated with those locations.

② apply attention score

The dot product of the query and the key is computed and then normalized using Softmax. The attention score at each position represents its importance relative to all other positions. These attention scores are used to guide how each position's value is weighted.

③ Calculated weighted sum

$$y_i = \sum_j A_{ij} V_j \quad (3)$$

where A_{ij} is the attention weight from position i to position j , and V_j is the value at position j . All these V_j values are summed up according to their weights A_{ij} to obtain the final feature representation at position i .

C. Image Moment

Invariant Moments, initially proposed by M.K. Hu in 1962, is a class of highly condensed image descriptors known for their inherent translation, grayscale, scale, and rotation invariance.^[5] Hereafter referred to as Hu's image moments, they serve as potent tools for describing image attributes.

The computation of Invariant Moments requires an initial derivation of the geometric and central moments, with the averages denoted as i and j respectively. These moments are then used to construct a density curve $f(i, j)$, which encapsulates the grayscale distribution on the image coordinates (i, j) . Subsequently, we get a matrix μ_{pq} , which represent image invariant feature. To correct for changes due to changes in image size or orientation, a matrix μ_{pq} is introduced to serve as the ultimate representation of image features. In the final stage of the process, the second and third derivatives of μ_{pq} are carefully extracted, resulting in a comprehensive seven-invariant eigenmatrix.

Firstly, we calculate the row moment m_{pq} , where i and j represent the pixel coordinates of the image, M and N represent the number of rows and columns of the image, and $f(i, j)$ means the pixel value of the image at coordinates (i, j) .

$$m_{pq} = \sum_{i=1}^M \sum_{j=1}^N i^p j^q f(i, j) \quad (4)$$

Next, the center moment μ_{pq} is obtained by subtracting the image center of mass, where \bar{i} and \bar{j} represents the centroid coordinate of the image, that is, the center of gravity position of the image.

$$\mu_{pq} = \sum_{i=1}^M \sum_{j=1}^N (i - \bar{i})^p (j - \bar{j})^q f(i, j) \quad (5)$$

Then, we calculate normal central moment η_{pq} , which standardize the central moment so that it remains invariant when scaled and rotated.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}}, (p+q = 2, 3, \dots) \quad (6)$$

Finally, we extract the higher-order features, to obtain 2nd to 8th moment invariants for describing shape and texture features of images.

$$\Phi_1 = \eta_{20} + \eta_{02} \quad (7)$$

$$\Phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \quad (8)$$

$$\Phi_3 = (\eta_{20} + 3\eta_{12})^2 + 3(\eta_{21} + 3\eta_{03})^2 \quad (9)$$

$$\Phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (10)$$

$$\Phi_5 = (\eta_{30} + 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + (3\eta_{21} + \eta_{03})(\eta_{21} + \eta_{03})[(3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (11)$$

$$\Phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (12)$$

$$\Phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[(3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (13)$$

D. Logistic Regression

When dealing with classification problems with multiple output classes, logistic regression proved essential and became an important cornerstone of our final classifier. Logistic regression can handle the prediction problem of multiple categories by introducing the Softmax function. In multi-class logistic regression, the goal of model training is to minimize the cross-entropy loss.

$$z(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (14)$$

$$p_k^{(i)} = \frac{e^{z_k^{(i)}}}{\sum_{j=1}^K e^{z_j^{(i)}}} \quad (15)$$

Then we count cost function $J(\theta)$, where m is the number of training samples, λ is the regularization parameters that control the complexity of the model, and n is the number of features.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(p_k^{(i)}) + \frac{\lambda}{2m} \sum_{j=1}^n \sum_{k=1}^K \theta_{jk}^2 \quad (16)$$

The cross-entropy loss function can guide the model to more accurately predict the probability of each category by measuring the difference between the probability distribution predicted by the model and the actual label distribution.

E. Stochastic gradient descent

Stochastic gradient Descent (SGD) is a very popular algorithm for optimizing machine learning models. It is especially good for large data sets because it uses only a small portion of the data in each iteration. This makes SGD faster and more efficient than batch gradient descent (gradient descent using the entire data set).

$$\theta_{t+1} = \theta_t - \eta \nabla f_i(\theta_t) \quad (17)$$

where η represents the learning rate.

F. AdaBoost

AdaBoost, which stands for “Adaptive Boosting”, is famous for its adaptive characteristic. It is adaptive in the sense that the samples misclassified by the previous basic classifier are strengthened and the weighted whole samples are used again to train the next basic classifier. [6] At the same time, a new weak classifier is added in each round until reach an error rate that is small enough or a pre-specified maximum number of iterations is reached. [7] First, we initialize the weight distribution of the training data. Each training sample is initially assigned the same weight $w_i = \frac{1}{N}$, so the initial weight distribution of the training sample set $D_1(i)$ is:

$$D_1(i) = (w_1, w_2, \dots, w_N) = \left(\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}\right) \quad (18)$$

Then, we iterate for $t = 1, \dots, T$. Firstly, after selecting the weak classifier h_t with the lowest error rate at the current step as the t th basic classifier H_t , we can calculate the error rate of this weak classifier on the distribution D_t :

$$e_t = P(H_t(x_i) \neq y_i) = \sum_{i=1}^N w_{i,t} I(H_t(x_i) \neq y_i) \quad (19)$$

As can be seen from the above formula, the error rate e_t of $H_t(x)$ on the training data set is the sum of the weights of the samples misclassified by $H_t(x)$.

Secondly, we calculate the weight of this weak classifier in the final classifier (denoted by α_t):

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - e_t}{e_t} \right) \quad (20)$$

Finally, we update the weight distribution of the training samples D_{t+1} :

$$D_{t+1}(i) = D_t(i) \exp(-\alpha_t y_i H_t(x_i)) / Z_t \quad (21)$$

where Z_t is the normalization constant:

$$Z_t = 2\sqrt{e_t(1 - e_t)} \quad (22)$$

The last step is combining the weak classifiers weighted by α_t to form a strong classifier:

$$f(x) = \sum_{t=1}^T \alpha_t H_t(x) \quad (23)$$

The strong classifier is obtained by applying the sign function sign:

$$H_{final} = \text{sign}(f(x)) = \text{sign} \left(\sum_{t=1}^T \alpha_t H_t(x) \right) \quad (24)$$

III. OTHER RELATED WORK

Medical image has witnessed significant advancements in both the fields of computer vision and healthcare, owing to the continuous exploration by experts. Waheed [8] uses the GAN-CNN network to create new data for Covid-19 detection using chest X-ray (CXR) images and to create new images to improve the performance of the VGG network in Covid-19 detection, even with limited datasets.

According to the article from Sethy [9], similar to the traditional computer vision algorithm for feature extraction and classification in steps, this paper proposes a new method to extract features from the segmented images of CNN deep neural networks. Then SVM model was used to train the extracted depth features. It chooses a balance between interpretation and representation.

In recent years, CNN has shown significant advantages in improving image classification, segmentation, and diagnostic accuracy [10]. However, due to the lack of transparency in their decision-making processes, this has become an issue as to whether neural networks can be implemented in the medical field, where explainability is crucial for maintaining the trust of patients and doctors.

Veeramuthu [11] proposed the Probabilistic Neural Network-Radial Basic Function method to improve the classification accuracy of tumor functional brain images. Multistage wavelet method is used to extract features for classification, then the morphological filtering technique is used to segmentation the process.

The above studies all used neural networks for disease monitoring, but they lack interpretability. In our study, CNN were not used for medical judgments. Instead, these challenges were addressed by combining CNN-based classification with Hu moment transformations and traditional machine learning classifiers, such as logistic regression, significantly enhancing the model's interpretability.

IV. METHODOLOGY

A. Pre-processing

We sourced an image dataset from Kaggle that covers three perspectives. This multi-perspective nature enables our proposed model to learn objects features from different perspectives, which improves the generalization ability. To

enhance the performance and stability of the model, a series of preprocessing steps were implemented.

First, we automatically filter out all non-image files, ensuring the accuracy and automation of the process. Subsequently, images were uniformly converted to RGB format and resized to 256x256 pixels to standardize the input size of the neural network and optimize calculations. Follow this, pixel values were normalized to the range [0.0, 1.0], ensuring the consistency of the network input and accelerating the learning process. Finally, color inversion was applied to enhance the contrast between the dark and the light, making the details in the image more obvious and helping professionals to better identify and analyze image features.

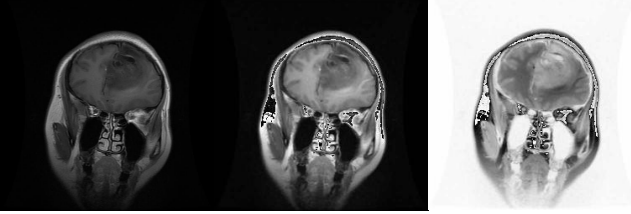


Fig. 6, 7, 8. Brain image original, standardization, inverted

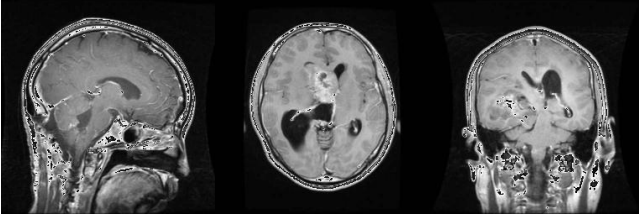


Fig. 9, 10, 11. Brain image besides, above, behind

The study aims to delineate the boundaries of image within an image array by setting a luminance threshold to identify pixels brighter than the specified threshold. Based on these identified pixels, the effective content boundary of the image is determined, thus eliminating the all-black boundary area. Ultimately, this process generates coordinates of the top, bottom, left, and the right edges of the image, making it easier to remove superfluous borders or whitespace areas. Images with different borders can be clipped automatically through a function that traverses all images.

B. Classification

1) CNN

We designed a CNN for three-view image classification. The convolutional layer utilizes its ability to automatically extract key features from the original image, while introducing ReLU as an activation function to introduce non-linearity. In addition, the pooling layer can effectively control the overfitting of the model by reducing the amount of computation and model parameters. The fully connected layer maps the learned high-level features to the final classification results, with the Softmax layer transforming these results into probability distributions, making the model output interpretable. employing the cross-entropy loss function and the Adam optimizer, we seek the optimal solution of the loss function.

To ensure the effectiveness of our proposed CNN architecture in addressing the three-view image classification problem, we plotted the loss curve. This graph shows the variation of the loss values after each training batch, aiding our comprehension of the model's learning behavior, while also demonstrating the effectiveness and stability of the model.

[<matplotlib.lines.Line2D at 0x1d8c8ee9bb0>]

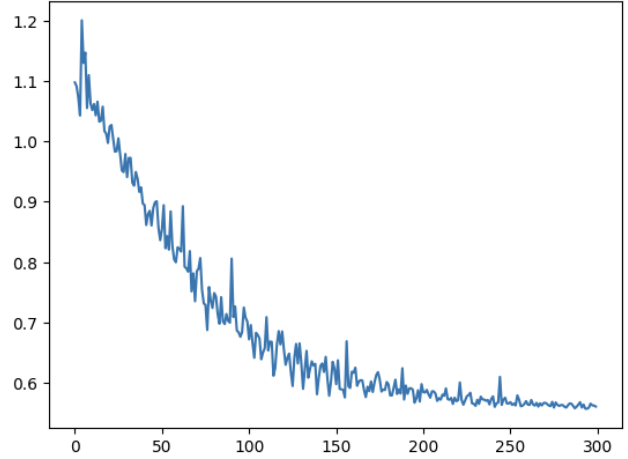


Fig. 12. Loss over Iterations

2) Self-Attention Mechanism

① Define Self-Attention Layer

First, we define a self-attention layer that enhances important features in the feature graph output by the convolutional layer. The self-attention layer uses the Query, Key, and Value components to calculate the attention weight for each location, thus adjusting the importance of features.

② Integrate the self-attention layer into the CNN model

After embedding the self-attention layers into the key convolutional layers in our CNN model, a self-attention layer is added behind each convolutional layer, especially those handling higher-level features, because their output benefits more from the attention mechanism.

③ Add self-attention mechanism to the model

Add a self-attention layer to each convolutional layer in the CNN model. First, initial features are extracted through the first convolutional layer, and then further processed through the first self-attention layer, allowing the model to focus on important regions of the image before max pooling. Next, the deeper features are extracted by the second convolutional layer and processed through the second self-attention layer. This approach helps the model focus on critical information at more complex levels of feature abstraction.

④ The problem arise in the implementation process

Computational complexity increase —The self-attention mechanism dynamically assigns attention weights to each feature by calculating the interrelationships between the input features. Specifically, for each input element, the

self-attention layer performs a series of comparisons and calculations with all other elements to determine which features are important and which can be ignored. This makes it particularly computationally expensive, especially when the dimensions of the input features are large.

Increased memory usage — The computations of the self-attention layer usually involve a large amount of intermediate data, such as matrices generated when attention weights is calculated. This not only increases computational complexity but also significantly raises memory usage. This high memory demand limits the application of the self-attention mechanism.

⑤ Influence

The increased computational burden directly affects the efficiency of the model in this paper. This means that the training and reasoning time of the model can increase significantly, which is a big problem for application scenarios that require real-time responses. In medical image analysis, fast and accurate model response is very important. However, accuracy can be improved at the expense of efficiency. Self-attention can improve our understanding of the model data of CNN and thus improve the accuracy of classification or prediction.

The trade-off between efficiency and accuracy is considered. The simple CNN model used is relatively basic. Due to its limitations of its parameter number and processing power, the performance improvement brought by the self-attention mechanism is not as significant as expected. This suggests that in relatively simpler network structures, complex mechanisms such as self-attention may not fully realize their potential advantages. Therefore, considering the trade-off between reduced efficiency and increased accuracy, the introduction of the self-attention mechanism did not significantly enhance the model's performance, although its practical value should not be overlooked.

In advanced applications, such as clinical medical image analysis, the requirement for precision far exceeds the need for processing speed. In these cases, introducing a self-attention mechanism can significantly improve the diagnostic accuracy of the model. For instance, advanced models can detect subtle lesions that standard models might miss, which is crucial for early diagnosis and treatment planning. Although these advanced models are computationally more complex, resulting in reduced processing efficiency, in the medical field, especially when dealing with complex and critical medical images such as tumors, the benefits of improved accuracy far outweigh the reduction in processing speed.

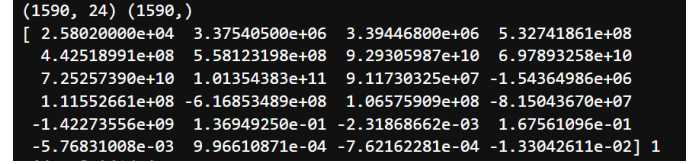
C. Image Moment

For feature extraction, non-linear transformations were applied to invariant moments and high-order moments.

How to get origin image moment? As paper mentioned before (background, image moment) Firstly get row moment

and center moment. Secondly, get function between gray scalation and regularization center moment. Then do non-linear transformations.

Through non-linear transformations, the representation of these features could be altered, achieving various image processing tasks. During the process of non-linear transformations, various methods can be used to adjust the values of invariant moments and high-order moments. For example, common non-linear functions such as exponential, logarithmic, and power functions could be make use of this purpose. This transformation can enhance or weaken specific image features to achieve the desired effect. In this way image high dimensional transfer to low dimensional image moment data. Then it will be regularization. In this way we got the image moment data from gray scale function.



```
(1590, 24) (1590,)
[ 2.58020000e+04  3.37540500e+06  3.39446800e+06  5.32741861e+08
 4.42518991e+08  5.58123198e+08  9.29305987e+10  6.97893258e+10
 7.25257390e+10  1.01354383e+11  9.11730325e+07 -1.54364986e+06
 1.11552661e+08 -6.16853489e+08  1.06575909e+08 -8.15043670e+07
-1.42273556e+09  1.36949250e-01 -2.31868662e-03  1.67561096e-01
-5.76831008e-03  9.96610871e-04 -7.62162281e-04 -1.33042611e-02] 1
```

Fig. 13. Exp Image moment

D. Classification-Logistic Regression

According to the accuracy that compared KNN, SVM and logistic regression, we chose logistic regression.

We adjusted the regularization strength to 10 and the adjusting intercept term to 10 to prevent overfitting. Also, we set a higher number of iterations (100) to ensure that the model can fully learn and converge on complex data structures.

We used cross-validation to evaluate the model's performance and adjusted the parameters based on the validation results. Through continuous adjustments and validations, we identified the optimal model configuration that achieves the highest classification accuracy. Through systematic parameter adjustments and model validation, we successfully enhanced the model's classification accuracy across all tumor types.

$$accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}} \quad (25)$$

Then, we integrated the results of multiple logistic regression models trained under different views and let each model predict independently. Finally, to prevent contingencies, we introduced a mechanism for multiple sample votes. Based on the training of multiple logistic regression models on different training datasets and their predictions, mitigate the effects of random results due to the particularity of one dataset. This ensemble method improves the reliability and robustness of our model's predictions.

$$\hat{y} = \text{majority vote}(y_1, y_2, \dots, y_n) \quad (26)$$

To ensure the effectiveness of our proposed logistic regression approach in addressing the classification problem, we plotted the learning curve. This graph shows the variation in accuracy after each training batch, aiding our

comprehension of the model's learning behavior and demonstrating the effectiveness and stability of the model.

E. AdaBoost

To increase the accuracy of our model, we apply the AdaBoost classifier. As we have mentioned before, the samples misclassified by the previous basic classifier are strengthened and the weighted whole samples are used again to train the next basic classifier. In our model, we use Logic regression as our weak classifier. Then, we divide the training sample to 10 parts, each part weight $\frac{1}{10}$. In each iteration, we select the best classifier to be the base classifier of the next iteration, and add a new weak classifier each iteration until the model gets the 100 iterations.

V. EVALUATION

In our brain image research, we are dedicated to developing a model capable of accurately identifying tumors in brain images. To evaluate the performance of our proposed model, we compare it with several that have been widely used in related fields before, including VGG16, ResNet100, and U-Net.

VGG16 performs well in tasks such as image classification on small data sets. We trained the VGG model for 100 epochs, achieving an impressive 96% accuracy, which reaches the highest accuracy among those models. ResNet100, performs well when dealing with more complex image data. We ran training for 1000 epochs, taking 6 hours, achieving 91% accuracy (for whether it is a tumor). Finally, U-Net is a classic special convolutional neural network, especially in medical image segmentation, classification and detection tasks. [12] Training it for 50 iterations we get an accuracy of 93% (for whether it is a tumor).

Compared these models against our interpretative model, evaluate them based on accuracy, precision, and F1 scores. Specific data are followings:

TABLE I

COMPARED DATA OF WHETHER IT IS A TUMOR

	VGG16	ResNet100	U-Net	Interpretative Model
Accuracy	96.21%	94.87%	95.69%	93.42%
Precision	95.5%	94.2%	95.0%	92.8%
F1-score	0.957	0.943	0.956	0.934

TABLE II

COMPARED DATA OF WHICH TYPE OF THE TUMOR

	VGG16	ResNet100	U-Net	Interpretative Model
accuracy	84.7%	82.53%	82.97%	77.82%
Precision	83.5%	81.8%	82.2%	76.9%
F1-score	0.845	0.825	0.829	0.778

Although our proposed model demonstrates a 93.42% and 77.82% accuracy rate, lower than the 96.21% and 84.7% achieved by VGG16 model. However, considering its emphasis on interpretability, its performance remains acceptable. This means that while the model may not achieve

the highest accuracy compared to more complex models, it still provides reliable results. The interpretability of the model allows for easier understanding and analysis of the decision-making process, which is crucial in fields where comprehending the underlying reasoning is as important as the accuracy of the outcome. Therefore, despite potentially lower performance accuracy, the model's transparency and explanatory power make it a valuable tool in our application.

In interpretative area, according to Explaining Explanations: An Overview of Interpretability of Machine Learning, Leilan and his team introduced an idea, he believes that it is better to come up with models that can be explained than to try to explain them after perfecting them[1].

Our team support this idea, and based this theory to propose all the model, after simple classification, we tried to avoid deep learning model used and use classical machine learning model based on mathematical like image moment. We team also try our best to improve our model accuracy to make our model accuracy close to deep learning one. It is a really hard task without no-linear transfer and multiple hidden layer design. We finally keep our accuracy is acceptable and have a well imperative.

In medical area, any method cannot be applied without understanding by patient and doctor. In this paper, we compare the clusters of tumor and conventional brain CT scans, highlighting the area's most likely to be problematic.

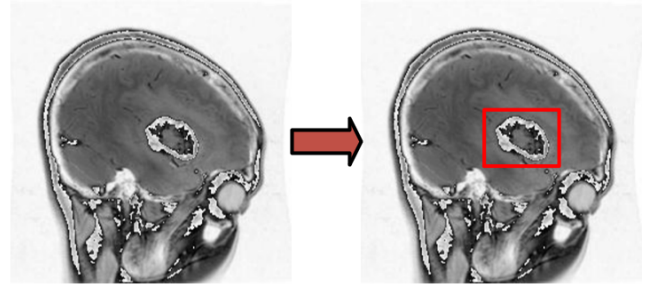


Fig. 14. The image has been processed by the model

At last, a questionnaire was prepared (The questionnaire is attached in the appendix). With help of professional doctor, we make sure this model has the potential to help doctors diagnose brain tumors in the future.

VI. FUTURE WORK

In the section on Hu moments, we only utilized 24 features, but the model can support more, providing higher-dimensional data for a more comprehensive analysis. Additionally, we can employ PCA and the Ray algorithm to reduce the dimensionality of Hu moments, assigning them weights to make image moments more effective and powerful.

Our team hope we can make have close connection with clinical medical institution, such as, hospital or medical lab, it still have a long way to build connection between theoretical models and clinical medicine. Maybe one day we can A wide range of use of questionnaires to consult professional doctors to complete the interdisciplinary. At last, interpretability medical image brings another possible in medical image.

VII. CONCLUSION

This paper builds a model that solve the challenge of the black-box nature of CNNs, making the prediction process more explainable by combining a CNN-based classifier with Hu's moment transformation and traditional machine learning classifiers like logistic regression. Given the critical

importance of this task for patient safety, our goal is to ensure that our model reaches the high level of both accuracy and reliability. Ultimately, the model demonstrates a 94% accuracy rate, slightly lower than the 97% achieved by the VGG network. However, considering its emphasis on interpretability, its performance remains acceptable.

APPENDIX

The questionnaire:



Figure 15 shows the first page of a questionnaire titled "解释性医学肿瘤分类模型评估报告" (Interpretative Medical Tumor Classification Model Evaluation Report). The page is illustrated with a background of medical icons like a syringe, virus, and bandage. It contains three numbered sections: 1. Interviewee Information (姓名: 汤惠茹, 专业领域: 靶向治疗药物研究所), 2. Model Introduction (模型名称: 解释性医学肿瘤分类模型, 目的: 对肿瘤进行分类), and 3. Validity Evaluation (准确性评估: 您认为模型在肿瘤分类方面的准确性如何?).

解释性医学肿瘤分类模型评估报告
an interpretative medical tumor classification model's evaluation report

1 受访医生信息 Information of interviewed doctors
姓名: 汤惠茹 Name: Huiru Tang
专业领域: 靶向治疗药物研究所
Professional field: Institute for Targeted Therapeutics
临床经验: 21年 (曾任职北京大学深圳医院妇产科副主任)
Clinical experience: 21 year (worked as deputy director of Obstetrics and Gynecology Department of Peking University Shenzhen Hospital)

2 模型介绍 Model Introduction
模型名称: 解释性医学肿瘤分类模型
Model name: Interpretative Medical tumor classification model
目的: 对肿瘤进行分类 Goal: To classify the tumor
输入特征: 描述您使用的特征 (例如, 影像特征、临床数据等)
Enter features: Describe the features you use (e.g., imaging features, clinical data, etc.)
输出结果: 分类结果 (例如, 良性/恶性、不同类型的肿瘤等)
Output results: Classification results (e.g., benign/malignant, different types of tumors, etc.)

3 有效性评估 Validity evaluation
准确性评估: 您认为模型在肿瘤分类方面的准确性如何?
Accuracy assessment: How accurate do you think the model is in classifying tumors?
准确性根据提供的测试集来看, 能达到90%左右 (辨认是否有肿瘤), 75%左右 (辨认不同类型的肿瘤)
According to the test set provided, it can reach about 90% (to identify whether there is a tumor), about 75% (to identify different types of tumors).
临床应用: 您认为该模型是否适用于实际临床应用?
Clinical application: Do you think this model is suitable for practical clinical application?
汤医生: 我认为目前来看, 这个模型并没有在临床验证过, 数据量级也较小, 作为学生项目, 它有一定的潜力在医学领域应用, 相对与过往实施的实验室模型也量级较小相对易于部署, 但医学临床应用是一个很复杂, 对精度要求很高的工作。在这条路上学生团队, 甚至是专家团队还有很长的路要走。
Dr. Tang: From what I see currently, this model hasn't been clinically validated yet and the scale of data is relatively small. As a student project, it holds certain potential for application in the medical field. Compared to past laboratory models implemented in hospitals, it's relatively easier to deploy due to its smaller scale. However, medical clinical application is highly complex and demands high precision. On this path, both student teams and even expert teams have a long way to go.
如果是, 您会在哪些情况下使用它?
If so, in what circumstances would you use it?
我可能会先在实验室应用, 防止由于疲劳导致的误判
I might use it in the lab first, to prevent misjudgments due to fatigue

Fig. 15. Page 1 of the questionnaire



Figure 16 shows the second page of the questionnaire. It contains two numbered sections: 4. Interpretability Evaluation (特征重要性: 您认为哪些输入特征对模型的分类决策最重要?) and 5. Model Improvement Suggestion (您是否有任何关于模型改进的建议?). The page also includes a signature line with the name "汤惠茹" (Tang Huiru).

解释性医学肿瘤分类模型评估报告
an interpretative medical tumor classification model's evaluation report

4 可解释性评估 Interpretability evaluation
特征重要性: 您认为哪些输入特征对模型的分类决策最重要?
Feature importance: Which input features do you think are most important for classification decisions in the model?
汤医生: 影像特征尤其重要, 例如肿瘤的形态、边界清晰度等。
Dr. Tang: Imaging features are particularly important, such as tumor morphology and boundary definition.

5 模型改进建议 Model improvement suggestion
您是否有任何关于模型改进的建议? 这有助于您进一步优化模型并提高其可用性。
Do you have any suggestions for model improvements? This helps you further optimize your model and improve its usability.
我建议模型可以进一步优化对不同类型肿瘤的识别能力, 特别是对罕见肿瘤的分类。同时, 结合使用其他诊断工具, 提升整体诊断的准确性和效率。可能在未来可以作为一个辅助的判断手段。
I suggest that the model can further optimize its ability to identify different types of tumors, especially the classification of rare tumors. At the same time, the combined use of other diagnostic tools to improve the accuracy and efficiency of the overall diagnosis. It may be used as an auxiliary judgment tool in the future.

签名 signature 汤惠茹

Fig. 16. Page 2 of the questionnaire

REFERENCES

- [1] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017).** Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118.
<https://doi.org/10.1038/nature21056>
- [2] Ashwath, V. A., Sikha, O. K., & Benitez, R. (2023). TS-CNN: A three-tier self-interpretable CNN for multi-region medical image classification. IEEE Access, 11, 78402–78418. <https://doi.org/10.1109/access.2023.3299850>
- [3] Srdan Lazendić, Janssens, J. L., Huang, S., & Aleksandra Pižurica. (2022). On interpretability of CNNs for multimodal medical image segmentation. 2022 30th European Signal Processing Conference (EUSIPCO).
<https://doi.org/10.23919/eusipco55093.2022.9909776>
- [4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25, 1097-1105.
<https://doi.org/10.1145/3065386>
- [5] Cao, Y., Miao, Q., Liu, J., & Gao, L. (2013). Advance and Prospects of AdaBoost Algorithm. Acta Automatica Sinica. [https://doi.org/10.1016/S1874-1029\(13\)60052-X](https://doi.org/10.1016/S1874-1029(13)60052-X).
- [6] Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms. CRC Press. <https://doi.org/10.1201/b12207>
- [7] Hu, M. K. (1962). Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory, 8(2), 179-187.
<https://doi.org/10.1109/TIT.1962.1057692>
- [8] Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., & Pinheiro, P. R. (2020). CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. IEEE Access, 8, 91916-91923. <https://doi.org/10.1109/ACCESS.2020.2994762>.
- [9] Sethy, P. K., Behera, S. K., Ratha, P. K., & Biswas, P. (2020). Detection of coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine. International Journal of Mathematical, Engineering and Management Sciences, 5(4), 643-651.
<https://doi.org/10.33889/UMEMS.2020.5.4.052>.
- [10] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.
<https://doi.org/10.1016/j.media.2017.07.005>
- [11] Veeramuthu, A., Meenakshi, S., & Darsini, V. P. (2015). Brain image classification using learning machine approach and brain structure analysis. Procedia Computer Science, 50, 388-394.
<https://doi.org/10.1016/j.procs.2015.04.030>
- [12] Ronneberger, O., Fischer, P., & Brox, T. (2015).** U-Net: Convolutional Networks for Biomedical Image Segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28
- [13] Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.
<https://arxiv.org/abs/1412.6980>
- [14] Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. Springer.
<https://doi.org/10.1007/978-1-4757-2440-0>
- [15] Guo, M., Liu, Z., Mu, T., & Hu, S. (2021). Beyond Self-Attention: External Attention Using Two Linear Layers for Visual Tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45, 5436-5447.
<https://doi.org/10.1109/TPAMI.2022.3211006>.
- [16] Moritz, N., Hori, T., & Roux, J. (2021). Capturing Multi-Resolution Context by Dilated Self-Attention. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5869-5873.
<https://doi.org/10.1109/ICASSP39728.2021.9415001>.

CONTRIBUTION

Equal contribution, Tang as the group leader proposed and implemented initial model and most of code. Zhao and Lu finished all the data preprocessing. Zhao raise three view voting idea and design origin CNN classification model. Then Ren proposed and add self-attention mechanism, implemented it and add into CNN network. Tang finish Image Moment's formula derivation and code implementation. Lu design logistic regression model does the grid search to optimize it. Xiao completed the final design of noise adjustment and Ada-boosting. She also organized other members to lead the writing of the final paper and completed the questionnaire for professional doctors. Last and most importantly, each member of this team well finished their own task. We are the best team. This paper writes for all members.