IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 November 2024, accepted 4 December 2024, date of publication 9 December 2024, date of current version 18 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3514319

RESEARCH ARTICLE

A Novel Federated Learning-Based Image Classification Model for Improving Chinese Character Recognition Performance

MIN-SUN KIM[1], CHANG-HO SON[2], AND SEOUNG-HO CHOI[3]

[1]Department of AI Application, Hansung University, Seoul 02876, South Korea
[2]Department of System Engineering, Korea Army Academy at Youngcheon, Yeongcheon, Gyeongsangbuk 38900, South Korea
[3]Faculty of Basic Liberal Art, College of Liberal Arts, Hansung University, Seoul 02876, South Korea

Corresponding authors: Chang-Ho Son (c13981@kaay.ac.kr) and Seoung-Ho Choi (jcn99250@naver.com)

This work was supported by the National Research Foundation of Korea (NRF) Grant, funded by the Korean Government (MSIT), under Grant NRF-2022R1F1A1062959.

ABSTRACT Chinese characters are an essential means of communication in the East Asian cultural regions. Chinese characters are characterized by many strokes and complex structures, some of which are very similar. However, the misrecognition of messy writing can significantly decrease the accuracy of Optical Character Recognition (OCR) systems. Therefore, filtering the messy writing is critical. We propose a novel federated learning-based deep learning model to classify messy Chinese handwriting, addressing the challenges posed by complex character structures and variations in writing quality. To validate the proposed method, we conducted experiments to compare our approach with global, local, FedAVG, and IPA federated learning methods. We evaluated our approach using the CNN, ResNet50V2, NASNetLarge, EfficientNetV2B0, EfficientNetB0, and swin transformer models. The results showed a 301.2% improvement in the test accuracy and a 241.54% increase in the area under the curve (AUC) score with the proposed method, particularly when EfficientNetB0 was trained using the global method. These findings confirm that the proposed model effectively classifies Chinese messy handwriting. Additionally, when the data were fed into the OCR system, the match rate between the image and the recognized characters improved by up to 7.14%.

INDEX TERMS Chinese character recognition, deep learning, explainable method, federated learning, handwriting classification model.

I. INTRODUCTION

Chinese characters are complex and sophisticated writing systems that have been used in East Asian cultures for thousands of years. In modern society, Chinese characters remain a crucial means of communication in several countries including China, Japan, and Korea. However, Chinese characters are composed of intricate structures with numerous strokes, and many characters resemble each other closely, such as 日 (sun) and 目 (eye), or 未 (not yet) and 末 (end) [1]. As society becomes increasingly digitized, the digital conversion of documents has become essential, driving the need for advancements in Optical Character Recognition (OCR) technology.

Handwritten Chinese Character Recognition (HCCR) has been a consistently studied technology since the 1980s [2]. However, the errors in this technology can be attributed to factors such as the similarity between characters [3], unstable stroke recognition [4], and the diverse handwriting styles of users, including messy handwriting [5]. Therefore, when characters are misrecognized, issues such as document errors and medical record inaccuracies can arise, making it crucial to enhance the performance of character recognition systems.

However, the complexity of Chinese characters and the diversity of handwriting styles are key factors that can reduce the performance of such technology. Therefore, it is

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

critical to address these challenges from several perspectives. This accurate recognition of Chinese characters improves the quality of document digitization, thereby ensuring data reliability. It also enhances the efficiency of information retrieval and utilization, thereby meeting the demands of digital society. Therefore, our approach involves classifying the input handwritten Chinese characters as neat or messy, thereby preventing misrecognition. The contributions of this study are as follows.

- For the first time, we proposed a system that can classify messy written Chinese characters.
- We introduced a new federated learning-based model to classify messy and neat written Chinese characters. We conducted experimental analyses using the four federated learning methods and the proposed method.
- We performed visual analysis using CCA, histogram-based statistical analysis, t-SNE, and UMAP to analyze messy written and neat written data.
- We experimented with precise classification methods for messy written Chinese characters using six image classification deep learning models and our new proposal.

The remainder of this paper is organized as follows. In Section II, we explain previous research related to Chinese character recognition. Section III describes the data composition and the system and model training methods used in this study. Section IV presents the experimental results obtained using the proposed method. Finally, Section V concludes the paper and discusses future research directions.

II. RELATED WORKS

Xu et al. [6] proposed to overcome the limitations of existing deep learning-based character recognition methods that require large data and resources. Unlike traditional image representation-based character recognition methods, the proposed method uses a meta-stroke library to build a conceptual model of Chinese characters based on prior knowledge. It extracts the strokes of characters and uses Bayesian Program Learning (BPL) to learn the conceptual model. In the character recognition process, Monte Carlo Markov Chain (MCMC) sampling is used to obtain a generative model of each character concept, which performs classification by calculating the probability that the target and training characters belong to the same category. Experiments demonstrate that the proposed method outperforms state-of-the-art deep learning-based methods on the ICDAR-2013 competition dataset, using minimal samples to build conceptual models for character classification prediction.

Zhuang et al. [7] proposed a convolutional neural network (CNN)-based model for recognizing handwritten Chinese characters, utilizing the CASIA-HWDB1.1 dataset, specifically training on 20 out of 3755 classes. The input images underwent median filtering to reduce salt-and-pepper noise before resizing to 28 $\times$ 28 pixels for CNN input. The model was implemented in MATLAB and involved several steps: data reading, image matrix transformation, random partitioning into training and testing sets, and constructing a 6-layer CNN. The architecture featured two convolutional layers, with the first using six filters and the second utilizing twelve, producing feature maps that were subsequently processed by a fully connected layer to classify characters. During training, the error or cost function was calculated, and weights were updated via mini-batch gradient descent, facilitating faster convergence and mitigating overfitting. After 5000 training iterations, the model achieved an overall recognition accuracy of 90.91% and a mean squared error of 0.0079. Character recognition accuracy varied, ranging from a high of 96.67% to a low of 80%. The combination of CNN and median filtering demonstrated significant recognition accuracy and potential for practical applications.

Zhao et al. [8] proposed a similarity-based method for recognizing and evaluating handwritten Chinese characters, combining text content recognition with signature authentication. The primary aim was to quantitatively assess the normative level of handwritten characters, specifically focusing on the Chinese phrase "同意办理" (agree to handle), commonly used in banking transactions. To achieve this, a dataset was created by collecting 400 handwritten images from five individuals, which were then preprocessed using a Z-S skeleton extraction algorithm to normalize handwriting thickness. The study employed various quantitative measures, including correlation coefficients, pixel match rates, and cosine similarities, to evaluate the resemblance between the handwritten samples and template characters. The results indicated that standardizing handwritten characters significantly enhances recognition rates, with the proposed method demonstrating exceptional performance in evaluating the normative levels of handwritten Chinese characters. The study achieved a remarkable recognition accuracy ranging from 98% to 100%, highlighting the effectiveness of using similarity metrics in character recognition tasks.

Bhowmik et al. [9] focused on distinguishing text from non-text in handwritten documents, addressing a crucial step in the automatic processing of such documents. This study developed a dilated spatial attention-based network to tackle various complex issues encountered in the process of separating non-text from text, particularly in the detection of touching components. Furthermore, a realistic dataset was prepared for the task, and the proposed model achieved an accuracy of 87.85% on this dataset. To compare the performance, the model was evaluated against seven feature-engineering-based methods and six deep learning-based methods, and the proposed model outperformed the existing methods in most cases.

Bhowmik [10] proposed a multi-scale dilated Convolutional Neural Network to classify text and non-text regions in document images. This network effectively captures local patterns and relative contexts across various scales, enabling accurate classification of document areas. The proposed method achieved an accuracy of 97.91% on the AUTNT dataset, which exceeds the existing performance by 1.63%. Additionally, it demonstrated applicability to multi-class problems by achieving accuracies of 99.31%

and 90.68% on the MNIST and Fashion-MNIST datasets, respectively.

Federated learning is a technique that enables learning to be conducted using data distributed across multiple devices while reducing the risk of data leakage. In traditional machine learning, users must send data to a cloud server, which increases the risk of leakage sensitive information. Federated learning was introduced to solve this problem. In [11] the performance of federated learning (FL) algorithms such as-FedAVG, FedProx, FedMA, FedAMP, and MOON-for image classification based on the Tiny-ImageNet dataset was evaluated. FedAVG is a simple algorithm that addresses the problem of data imbalances in mobile devices. FedProx addresses system and statistical heterogeneity issues but performs similarly to FedAVG. FedMA enhances the performance and reduces the communication overhead using a Bayesian non-parametric approach that adapts to data heterogeneity. FedAMP strengthens the connections between clients to solve the depersonalization problem. Finally, MOON introduces contrastive learning to reduce the differences between the models and exhibit the best performance. The experimental results showed that MOON and FedAMP effectively addressed the non-IID problem and exhibited excellent performance, FedMA exhibited moderate performance, and FedAVG and FedProx exhibited similar performances.

Tanmoy et al. [12] proposed a new approach to improve the existing FedAVG algorithm in federated learning. This enables efficient learning while protecting the privacy of the data. However, the existing FedAVG algorithm suffers from performance degradation due to data imbalance and asynchrony among clients. To address these issues, the Globally Shared Model-FedAVG(GSM-FedAVG) algorithm, which introduces a global shared model, was proposed. GSM-FedAVG enhances learning performance by incorporating a global shared model during local model training. The process includes the following steps: Initially, the global model was distributed to clients. Each client then trains a local model using local data. Complementary learning was simultaneously conducted using a global shared model. Local and global shared models were integrated at the central server to generate a new global model. Various experiments have been conducted to evaluate the performance of the GSM-FedAVG algorithm. The proposed algorithm demonstrated a higher accuracy and stability than the existing FedAVG algorithm. In particular, GSM-FedAVG maintained excellent performance even in situations with severe data imbalance.

Selvaraju et al. [13] introduced Gradient-weighted Class Activation Mapping (Grad-CAM), a method for providing visual explanations for CNN-based model decisions. Grad-CAM generates coarse localization maps that identify crucial image regions relevant to predicting specific concepts, such as 'dog' in image classification. This technique utilizes gradients from the target concept flowing into the final convolutional layer, making it applicable to various CNN architectures without the need for retraining or modifications. Grad-CAM excels at generating high-resolution, class-discriminative visualizations, particularly when combined with existing fine visualization techniques, resulting in Guided Grad-CAM. It has proven effective in applications such as image classification, image captioning, and visual question-answering (VQA). Importantly, Grad-CAM enhances model interpretability by analyzing CNN failure modes, identifying biases in datasets, and showcasing the spatial focus of models. The study highlights Grad-CAM's success in the ILSVRC-15 weakly supervised localization task and its role in evaluating model fidelity. Through human studies, it has been shown that Guided Grad-CAM helps users understand model decisions, fostering trust in automated systems by enabling untrained users to distinguish between strong and weak networks. Overall, Grad-CAM is positioned as a vital tool for improving the transparency and reliability of AI systems, contributing significantly to the field of explainable AI.

Lundberg [14] proposed a new framework called SHapley Additive exPlanations(SHAP) to interpret complex model predictions. SHAP assigns importance to each feature during the prediction process by introducing the concept of additive feature importance. This framework identifies a unique solution within this class of desirable properties and provides a consistent measure of feature importance across various methods (Local interpretable model-agnostic explanation(LIME), DeepLIFT, Layer-Wise Relevance Propagation, Shapley regression values, Shapley sampling values, and Quantitative Input Influence). SHAP enhances computational performance and produces results that are more aligned with human intuition. The key contributions of this study include introducing the concept of explanation models, proposing SHAP values as a unified measure of feature importance, and introducing new methods to estimate SHAP values. SHAP values are the only additive feature importance measures that satisfy local accuracy, missingness, and consistency. Although the exact computation is difficult, efficient estimation can be achieved using two model-agnostic approximation methods (Shapley sampling values and Kernel SHAP). This study balances the interpretability and accuracy of complex models, thereby providing users with more reliable model interpretations. SHAP integrates various feature attribution methods and presents a consistent approach for interpreting the model predictions.

As in previous studies on handwriting recognition [6], [7], [8], research on recognizing handwritten text exists; however, no studies have yet focused on filtering out illegible handwriting from handwritten Chinese character inputs, ensuring that only neat writing images are processed by OCR. Therefore, we propose a first messy writing classification system and introduce a novel federated learning approach to enhance the model performance.

Federated learning is characterized by utilizing local data from each client to train the model, which has the advantage of protecting data privacy and enabling learning from distributed data sources [11], [12], [15], [16]. However, in existing approaches, performance degradation of the global model or overfitting to specific client data can occur due to differences in data distributions between clients. To address these issues and improve the overall performance of the model, we propose a new federated learning approach. In this method, the same data is input into two independent local models, which are then trained separately. The weights of these two models are averaged and integrated into a new global model. This approach allows the individual models to maintain their distinct characteristics while ensuring the integrated model learns more balanced weights. We expect that this approach will enhance learning performance by leveraging data diversity and contribute to improving the generalization capability of the global model.

III. PROPOSED METHOD

We propose a Chinese character messy writing classification solution that classifies input characters as either neat or messy writing, ensuring that only images classified as neat writing are processed through the OCR.

The overall system structure of the proposed program is shown in Fig. 1. During the data collection phase, the data necessary for training were collected. Images of Chinese characters typed on a computer and converted into images were collected as neat images, whereas images of Chinese characters written by hand on a smartphone and converted into images were collected as messy images. The images were resized to 256 × 256 pixels and loaded in color format. To stabilize the neural network training and improve performance, the pixel values of the images were normalized to values between 0 and 1. After normalization, the images were labeled 0 for neat and 1 for messy to complete the preprocessing step. We used six models for training: a CNN, four pre-trained models, and a swin transformer. The output layer was set to sigmoid to produce probabilities for each class. Model evaluation was conducted using the receiver operating characteristic (ROC) Curve, performance metrics, confusion matrix, SHAP, LIME, and GRAD-CAM to visualize and interpret the performance of the models.

The dataset for the Chinese characters consisted of 14,969 images. We collected 9,998 images of neatly written characters and manually wrote Chinese characters to gather a variety of messy writing images, resulting in 4,971 messy writing characters.

Fig. 2 illustrates the proposed federated learning method. In this method, the same dataset is input to two local models. The weights of the two models are summed and averaged to create a new model. The difference is that FedAVG [11], [12] inputs different datasets, while the proposed method inputs the same data. Algorithm 1 is a pseudocode for our proposed method.

Algorithm 1 Pseudocode for Our Proposed Method

Require: Dataset D, Initial global model M_{global}, Number of epochs E
Ensure: Updated global model with averaged weights
1: **function** ModelWeightAveraging
2: Initialize local model M_{local}
3: **for** epoch = 1 to E **do**
4: $model\_A \leftarrow$ train_model(M_{local}, D)
5: $model\_B \leftarrow$ train_model(M_{local}, D)
6: $averaged\_weights \leftarrow$ average_weights($model\_A$, $model\_B$)
7: $M_{local} \leftarrow$ update_local_model($averaged\_weights$)
8: Evaluate M_{local} performance
9: **end for**
10: **return** M_{local}
11: **end function**
12: **function** train_model($model$, $dataset$)
13: Train $model$ on $dataset$ using forward pass and optimizer
14: **return** trained $model$
15: **end function**
16: **function** average_weights($model\_A$, $model\_B$)
17: Initialize empty $weights\_avg$
18: **for** each layer L **do**
19: $weights\_avg[L] \leftarrow \frac{model\_A[L]+model\_B[L]}{2}$
20: **end for**
21: **return** $weights\_avg$
22: **end function**
23: **function** update_global_model($averaged\_weights$)
24: Set M_{global} weights to $averaged\_weights$
25: **return** updated M_{global}
26: **end function**

The pseudo-code presented in Algorithm 1 details our proposed method for model weight averaging in a federated learning framework. The algorithm begins by requiring dataset D, an initial global model M_{global}, and the number of epochs E to train the model (line 1).

The primary function, ModelWeightAveraging, is defined to perform the weight averaging process (line 4). It initializes the global model (line 5) and iteratively trains two local models $model\_A$ and $model\_B$ on the same dataset for E epochs (lines 6-7). The weights of each model were then averaged using the average_weights function (line 8). This function calculates the average weights layer by layer, which helps to mitigate the risk of overfitting by combining the knowledge gained from both models (lines 12-14).

The averaged weights wwre then used to update the global model (line 9), which was evaluated for its performance at each epoch (line 10). The final output of the ModelWeightAveraging function is the updated global model with average weights (line 11).

The auxiliary functions train_model (lines 15-19), average_weights (lines 21-25), and update_global_model (lines 27-31) facilitate the model training, weight averaging, and global model updating, respectively.

IV. EXPERIMENTED METHODS

To analyze Chinese character image data, we employed data visualization methods (t-SNE and UMAP), correlation analysis (CCA), and t-test statistical comparison. In the

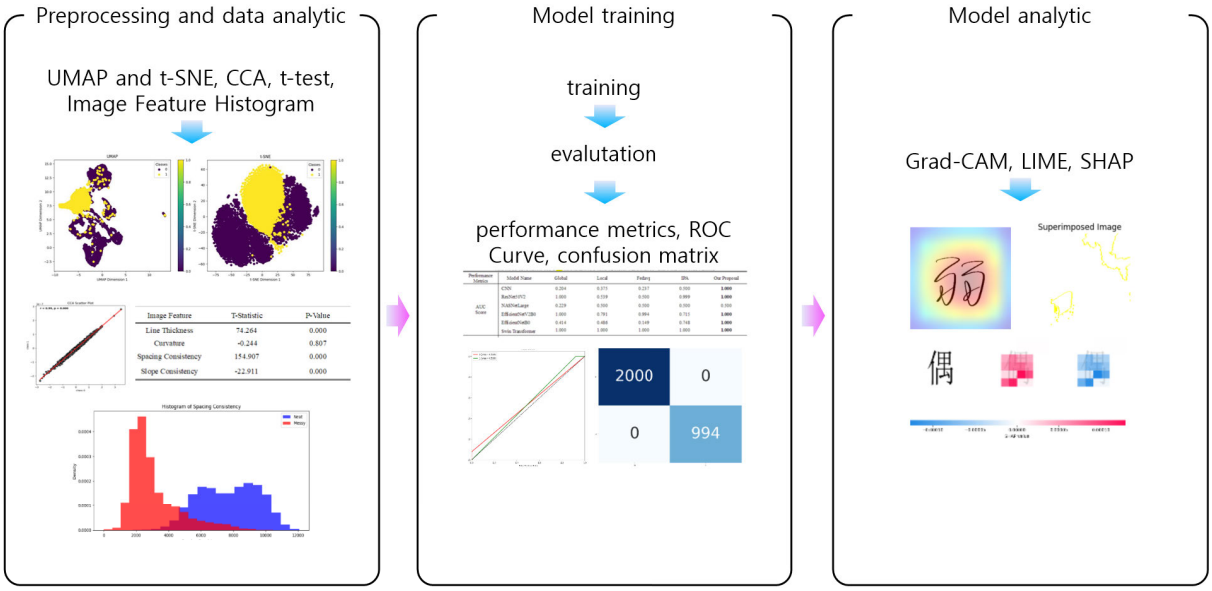

FIGURE 1. Proposed chinese character classification system.

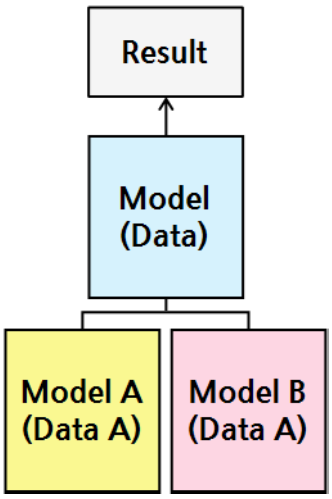

FIGURE 2. Structural diagram of the new proposed federated learning approach.

TABLE 1. Configuration of experimented dataset.

	Neat writing	Messy writing	Overall
Train dataset	5,999	2,982	8,981
Validation dataset	1,999	995	2,994
Test dataset	2,000	994	2,994
Overall	9,998	4,971	14,969

TABLE 2. Results of t-statistic analysis based on histogram of image features.

Image Feature	t-statistic	p-value
Line Thickness	74.264	0.000
Curvature	-0.244	0.807
Spacing Consistency	154.907	0.000
Slope Consistency	-22.911	0.000

t-SNE visualization, the input data x are first converted into a 2-dimensional array, and then the t-SNE is used to embed the data into two dimensions. The embedded data were plotted as a scatter plot using a color map. The UMAP visualization follows a similar procedure. The input data were converted into a 2-dimensional array, embedded into two dimensions using the UMAP, and visualized as a scatter plot.

CCA (Canonical Correlation Analysis) is a method used to analyze the relationship between two datasets. We classified neat written characters as Class 0 and messy written characters as Class 1. Each data class was unfolded into a 2-dimensional array and standardized before analysis using the CCA model. The data for each class were transformed to obtain the canonical correlation scores. After calculating the correlation coefficient and p-value, the results were visualized using scatter plots and regression lines.

The t-test statistical comparison involves the t-statistic and p-value, which are indicators of statistical test results. Four image features were examined, namely line thickness, curvature, character spacing consistency and inclination consistency. Line thickness was calculated as the inverse of the average pixel value. The stroke curvature was determined by the ratio of the curved parts after edge detection. The consistency of character spacing was measured using the standard deviation of the vertical projection. Consistency of inclination was assessed by the standard deviation of HOG(Histogram of Oriented Gradients) features.

The dataset used in the experiments was structured as shown in Table 1. The dataset was divided into training,

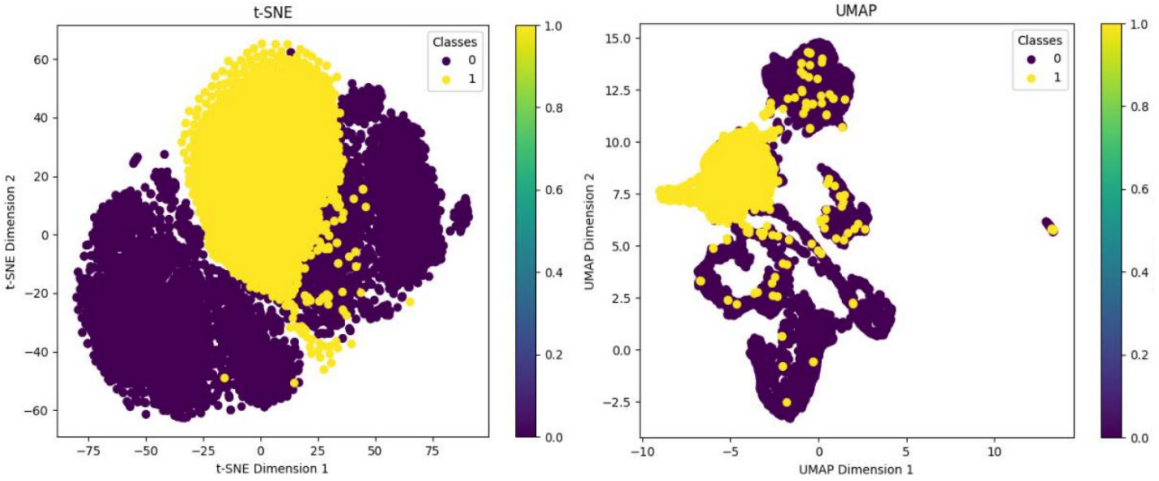

FIGURE 3. Visualization of data distribution between messy and neat writing, (a) t-SNE and (b) UMAP.

TABLE 3. Performance analysis using six classifiers and six evaluation measures to verify the proposed method.

Performance Metrics	Model Name	Global	Local	FedAVG [11], [12]	IPA [23]	Our Proposal
AUC Score	CNN [7], [17]	0.204	0.375	0.237	0.500	**1.000**
	ResNet50V2 [18]	1.000	0.539	0.500	0.999	**1.000**
	NASNetLarge [19]	0.229	0.500	0.500	0.500	0.500
	EfficientNetV2B0 [20]	1.000	0.791	0.994	0.715	**1.000**
	EfficientNetB0 [21]	0.414	0.486	0.149	0.748	**1.000**
	Swin Transformer [22]	1.000	1.000	1.000	1.000	**1.000**
F1 Score	CNN [7], [17]	0.668	0.633	0.668	0.668	**0.790**
	ResNet50V2 [18]	1.000	0.534	0.332	0.668	0.711
	NASNetLarge [19]	0.668	0.406	0.332	0.668	0.332
	EfficientNetV2B0 [20]	1.000	0.301	0.332	0.332	**1.000**
	EfficientNetB0 [21]	0.332	0.486	0.332	0.385	**1.000**
	Swin Transformer [22]	1.000	0.999	0.999	0.999	**1.000**
Recall	CNN [7], [17]	0.668	0.633	0.668	0.668	**0.790**
	ResNet50V2 [18]	1.000	0.534	0.332	0.668	0.711
	NASNetLarge [19]	0.668	0.406	0.332	0.668	0.332
	EfficientNetV2B0 [20]	1.000	0.301	0.332	0.332	**1.000**
	EfficientNetB0 [21]	0.332	0.486	0.332	0.385	**1.000**
	Swin Transformer [22]	1.000	0.999	0.999	0.999	**1.000**
Precision	CNN [7], [17]	0.668	0.633	0.668	0.668	**0.790**
	ResNet50V2 [18]	1.000	0.534	0.332	0.668	0.711
	NASNetLarge [19]	0.668	0.406	0.332	0.668	0.332
	EfficientNetV2B0 [20]	1.000	0.301	0.332	0.332	**1.000**
	EfficientNetB0 [21]	0.332	0.486	0.332	0.385	**1.000**
	Swin Transformer [22]	1.000	0.999	0.999	0.999	**1.000**
Loss	CNN [7], [17]	52.457	91.005	2.331	411.554	**0.606**
	ResNet50V2 [18]	0.000	10222.134	0.989	3.505	1.630
	NASNetLarge [19]	417963.781	444591534955724.800	268191236096.000	16616029184.000	40685383680.000
	EfficientNetV2B0 [20]	0.000	3.962	2.561	3.383	**0.000**
	EfficientNetB0 [21]	7.360	2.050	3.21	0.925	**0.000**
	Swin Transformer [22]	0.002	0.007	0.003	0.005	**0.002**
Accuracy	CNN [7], [17]	0.668	0.633	0.668	0.668	**0.790**
	ResNet50V2 [18]	1.000	0.534	0.332	0.668	0.711
	NASNetLarge [19]	0.668	0.406	0.332	0.668	0.332
	EfficientNetV2B0 [20]	1.000	0.301	0.332	0.332	**1.000**
	EfficientNetB0 [21]	0.332	0.486	0.332	0.385	**1.000**
	Swin Transformer [22]	1.000	0.999	0.999	1.000	**1.000**

validation, and test datasets, which were allocated at 60%, 20%, and 20% respectively.

Deep learning is a subset of machine learning (ML), which has the ability to automatically learn representations from raw

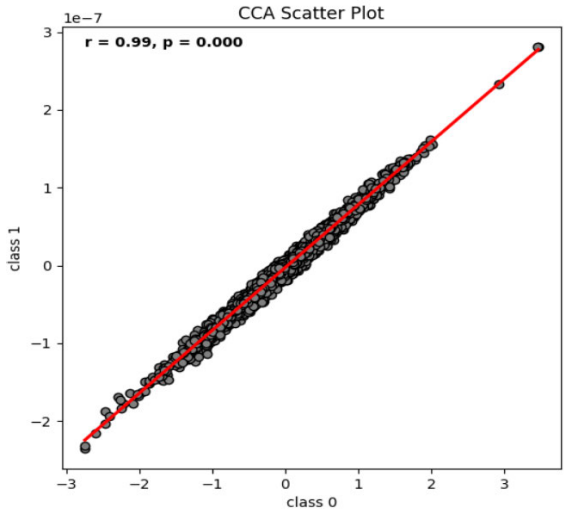

FIGURE 4. Correlation analysis between neat and messy writing using CCA.

data, and has emphasized its importance in computer vision. CNNs serve as the core of DL models in computer vision and consist of interconnected layers of neurons arranged in a manner similar to the human visual cortex. CNNs learn to extract important features from pixels, thereby reducing the need for manual feature engineering. This capability allows CNNs to capture complex patterns and structures within images, which is crucial for enhancing the performance of various computer vision tasks. The models used included CNN [7], [17], as well as pre-trained ResNet50V2 [18], NASNetLarge [19], EfficientNetV2B0 [20], EfficientNetB0 [21], and Swin transformer [22].

The top layers were excluded, and ImageNet weights were loaded. Pooling was performed using a Global Average Pooling two-dimensional (2D) layer to reduce the dimensionality of the output of the base model. A Fully Connected Layer with 1024 units was added, using ReLU as the activation function. The final output is a dense layer with two units using the sigmoid activation function to output the class probabilities. The deep learning models were trained using the Adam optimization method, and the loss function was the categorical cross-entropy. The model with the lowest validation loss was saved, and early stopping was set to terminate the training if the validation loss did not improve for two consecutive epochs. The models were trained for 10 epochs with a batch size of 32 for performance evaluation.

The experiment used a lightweight version of a swin transformer tailored for this task. This model processes input images by dividing them into patches and applying attention on a window basis. The core component, the SwinTransformerBlock, includes multi-head attention and MLP layers, which reduce spatial information loss by partitioning and recombining the input through window-based attention. Patch embedding transforms the input into a lower-dimensional representation, improving both learning speed and performance. By streamlining the original swin transformer, the model efficiently extracted features through Conv2D-based patch embedding and window-shifted self-attention.

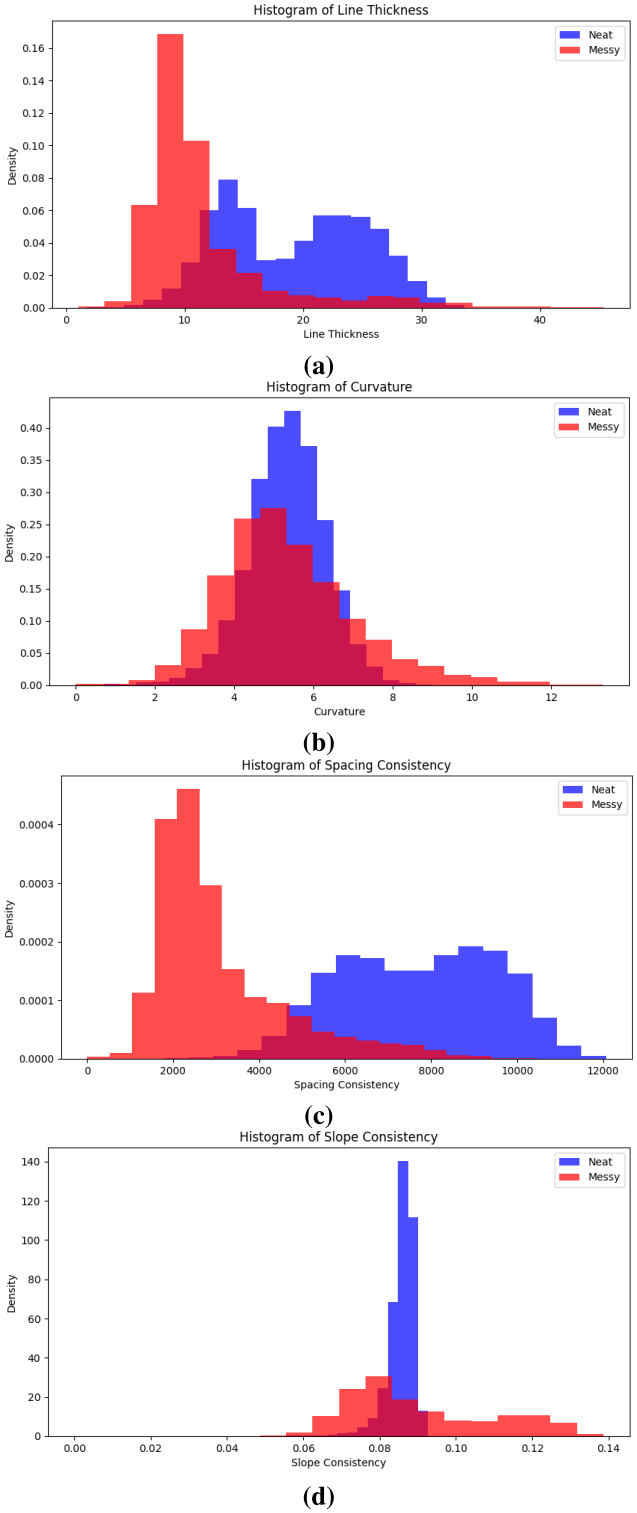

FIGURE 5. Histogram visualization analysis for frequency analysis of image features, (a) Line Thickness, (b) Curvature, (c) Spacing Consistency, and (d) Slope Consistency.

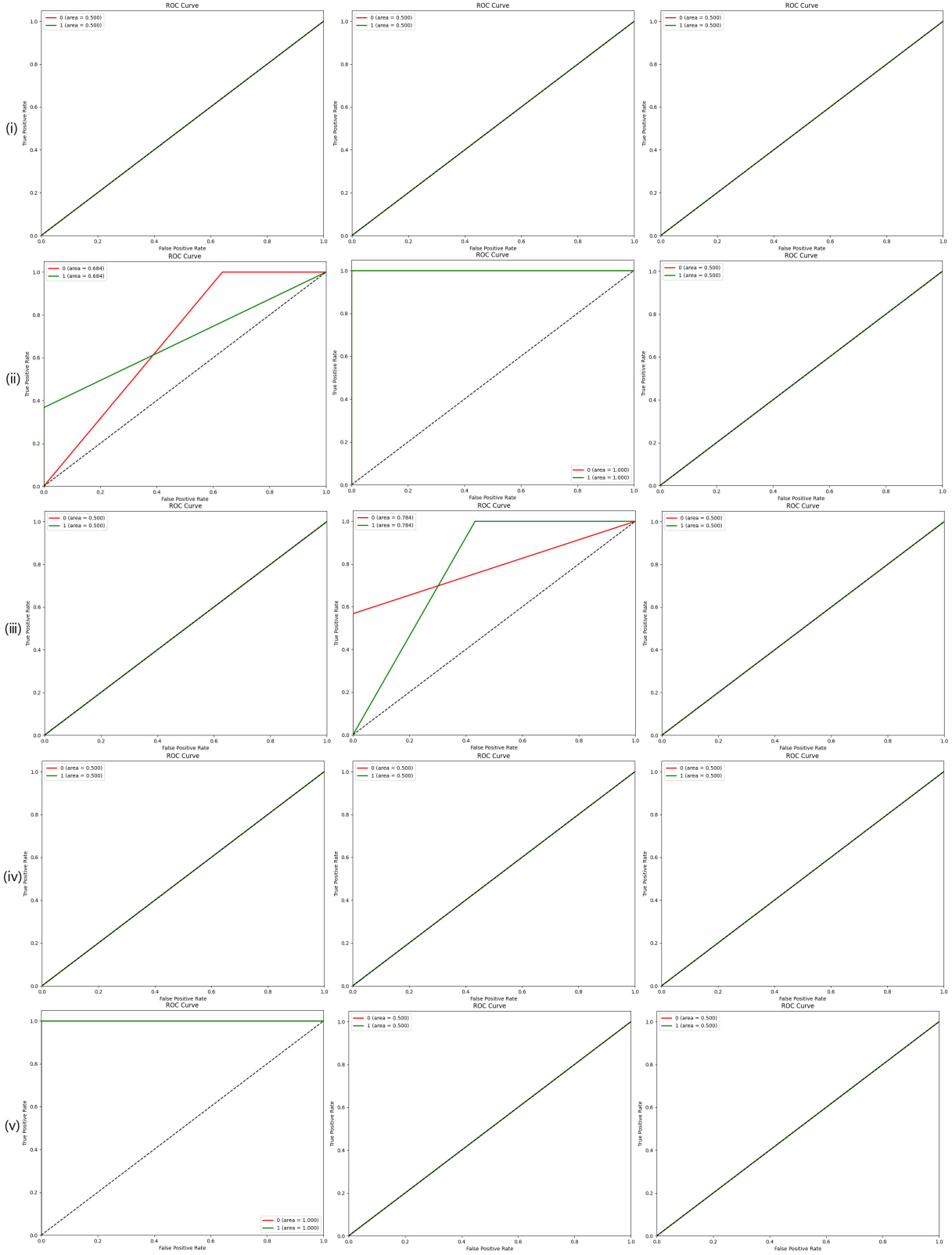

FIGURE 6. Visualization of ROC Curve results for comparative analysis of model convergence performance, (i)-(a) CNN using Global, (i)-(b) CNN using FedAVG, (i)-(c) CNN using IPA, (ii)-(a) CNN using Our proposal, (ii)-(b) ResNet50V2 using Global, (ii)-(c) ResNet50V2 using FedAVG, (iii)-(a) ResNet50V2 using IPA, (iii)-(b) ResNet50V2 using Our proposal, (iii)-(c) NASNetLarge using Global, (iv)-(a) NASNetLarge using FedAVG, (iv)-(b) NASNetLarge using IPA, (iv)-(c) NASNetLarge using Our proposal, (v)-(a) EfficientNetV2B0 using Global, (v)-(b) EfficientNetV2B0 using FedAVG, (v)-(c) EfficientNetV2B0 using IPA.

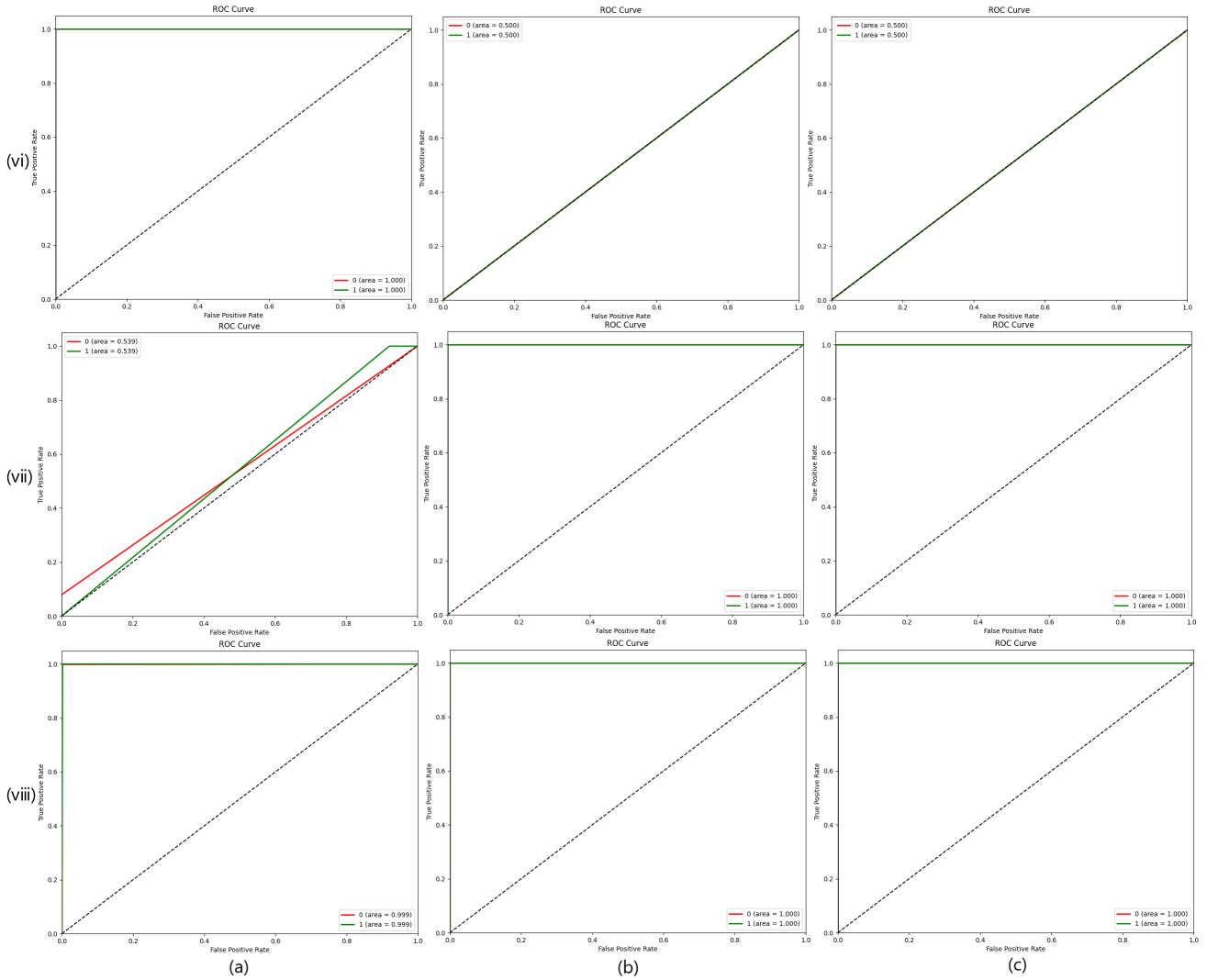

FIGURE 6. *(Continued.)* **Visualization of ROC Curve results for comparative analysis of model convergence performance, (vi)-(a) EfficientNetV2B0 using Our proposal, (vi)-(b) EfficientNetB0 using Global, (vi)-(c) EfficientNetB0 using FedAVG, (vii)-(a) EfficientNetB0 using IPA, (vii)-(b) EfficientNetB0 using Our proposal, (vii)-(c) Swin Transformer using Global, (viii)-(a) Swin Transformer using FedAVG, (viii)-(b) Swin Transformer using IPA, (viii)-(c) Swin Transformer using Our proposal.**

The federated learning methods used in the experiments were compared with the global, local, FedAVG, and the proposed method. The Global method uses a non-split training dataset. The Local method used a training dataset split into ten parts, and the results from the ten models were averaged. The FedAVG method [11], [12] inputs the training dataset, which is split into ten parts, into ten models. The weights from these ten models were then averaged to create a new model, which was then tested using the non-split test dataset.

The IPA method [23] also uses a training dataset that is split into ten parts. Among the ten models, two were used to train the two models. The weights of these two models were averaged, and the averaged model was used to create two new models. The new models were trained using two other datasets. This process was repeated five times and the final model was tested using the test dataset.

$$\text{Precision} = \frac{\text{true positives}}{\text{no. of predicted positives}} \tag{1}$$

$$\text{Recall} = \frac{\text{true positives}}{\text{no. of actual positives}} \tag{2}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Eq. 1 and 2 represent Precision and Recall, respectively. The precision indicates the proportion of actual neat writing among the predictions made by the model. This measures the accuracy of the model in predicting neat writing. Recall, on the other hand, refers to the proportion of actual neat writing that the model correctly identifies as neat writing. In other words, Recall reflects how well the model detects actual neat writing. Eq. 3 represents the F1 Score. The F1

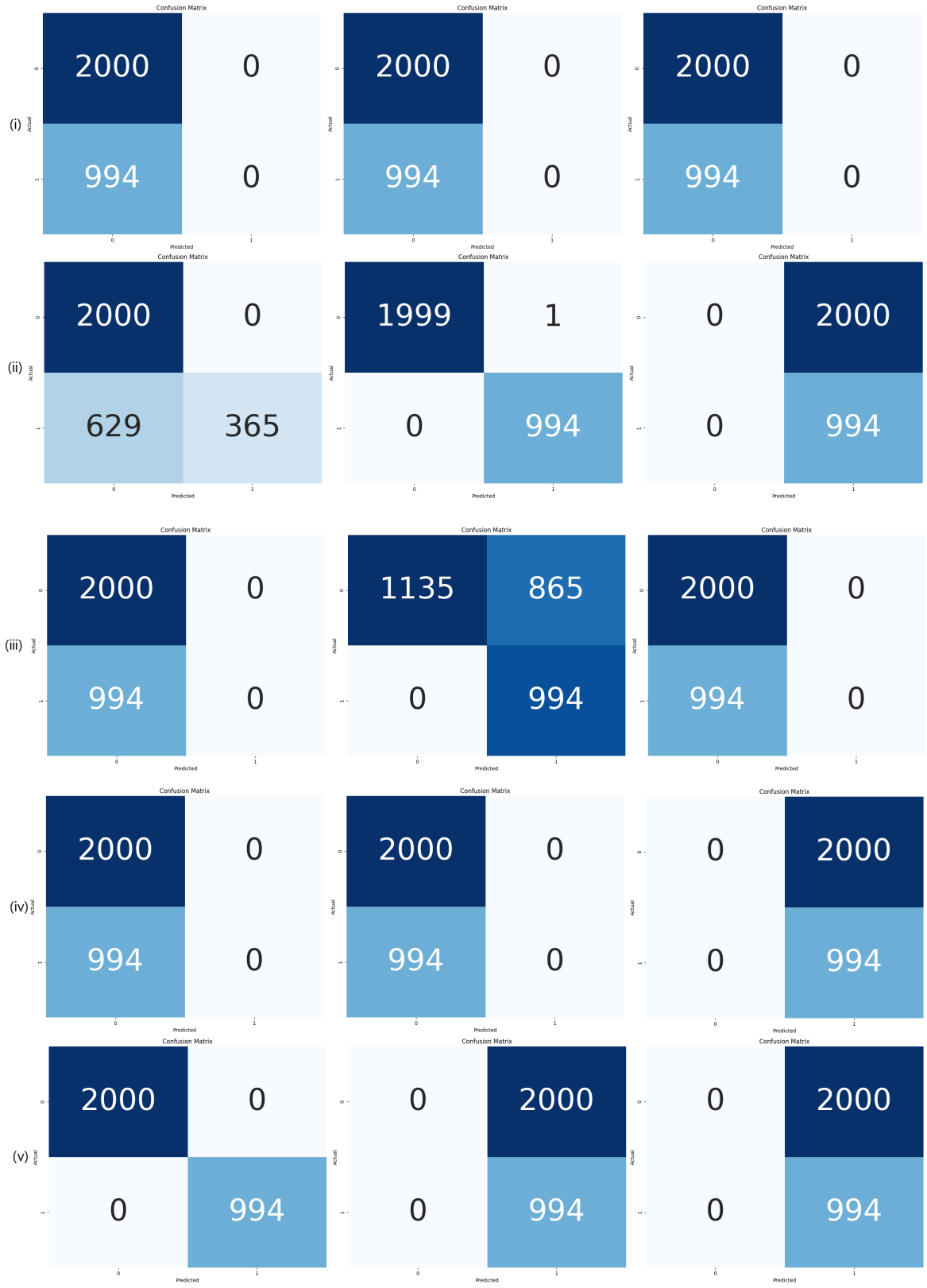

FIGURE 7. Visualization of Confusion matrix results for comparative analysis of model performance, (i)-(a) CNN using Global, (i)-(b) CNN using FedAVG, (i)-(c) CNN using IPA, (ii)-(a) CNN using Our proposal, (ii)-(b) ResNet50V2 using Global, (ii)-(c) ResNet50V2 using FedAVG, (iii)-(a) ResNet50V2 using IPA, (iii)-(b) ResNet50V2 using Our proposal, (iii)-(c) NASNetLarge using Global, (iv)-(a) NASNetLarge using FedAVG, (iv)-(b) NASNetLarge using IPA, (iv)-(c) NASNetLarge using Our proposal, (v)-(a) EfficientNetV2B0 using Global, (v)-(b) EfficientNetV2B0 using FedAVG, (v)-(c) EfficientNetV2B0 using IPA.

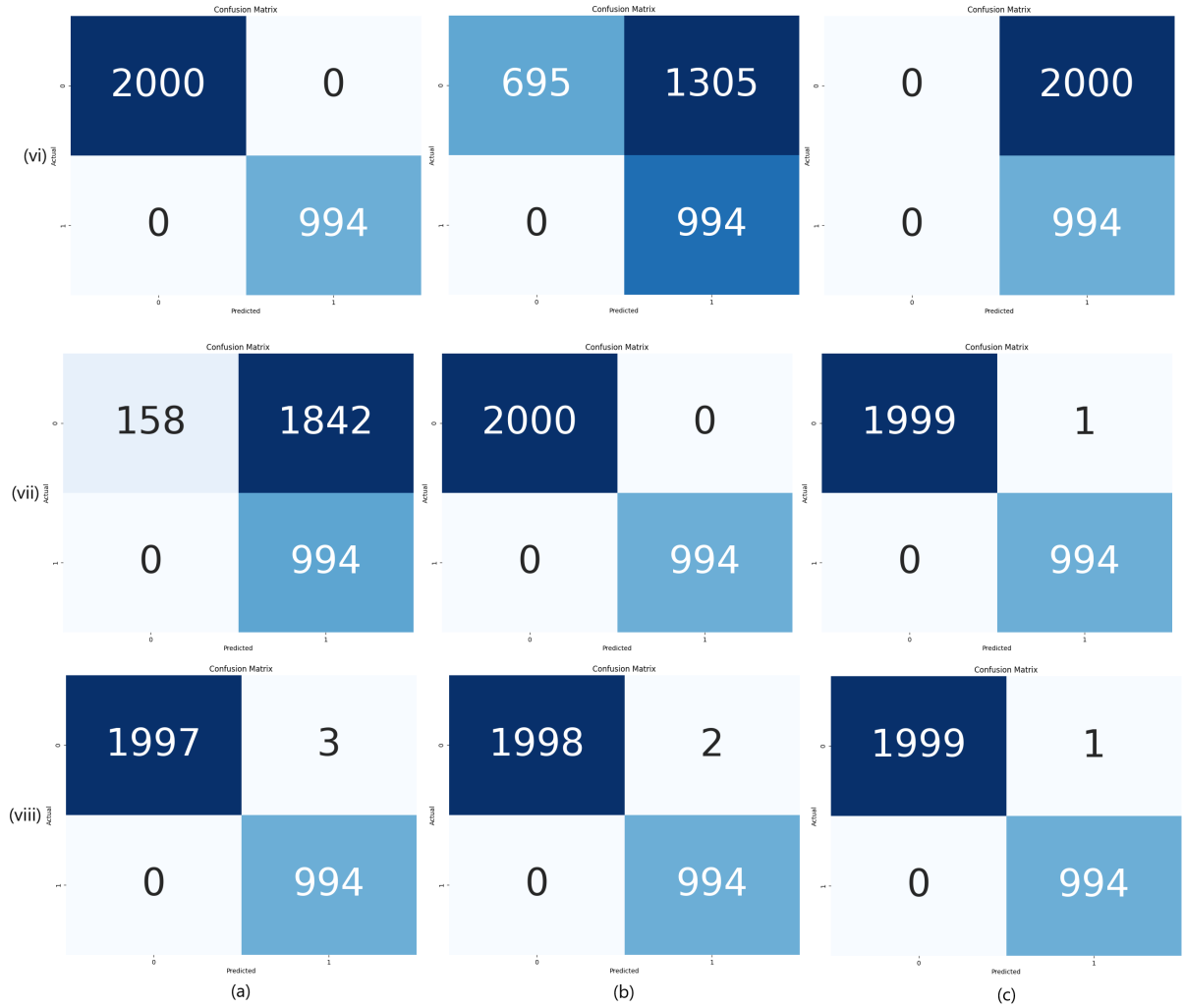

FIGURE 7. ***(Continued.)*** **Visualization of Confusion matrix results for comparative analysis of model performance, (vi)-(a) EfficientNetV2B0 using Our proposal, (vi)-(b) EfficientNetB0 using Global, (vi)-(c) EfficientNetB0 using FedAVG, (vii)-(a) EfficientNetB0 using IPA, (vii)-(b) EfficientNetB0 using Our proposal, (vii)-(c) Swin Transformer using Global, (viii)-(a) Swin Transformer using FedAVG, (viii)-(b) Swin Transformer using IPA, (viii)-(c) Swin Transformer using Our proposal.**

Score is a metric that evaluates the performance of model by simultaneously considering both Precision and Recall. It is calculated as the harmonic mean of Precision and Recall, and indicates how balanced the model's classification is.

The Confusion Matrix is used to assess the performance of a classification model by comparing the actual classes with those predicted by the model. This matrix displays the number of True Positives, True Negatives, False Positives, and False Negatives. The confusion matrix shows the False Negatives(FN) at the top right, True Negatives(TN) at the bottom right, False Positives(FP) at the top left, and True Positives(TP) at the bottom left.

A receiver operating characteristic (ROC) Curve was used to evaluate the performance of the binary classification model. This curve compares the False Positive Rate (FPR) and the True Positive Rate (TPR) to visualize the model's prediction performance. A larger area under the curve (AUC) indicated better classification performance of the model.

Local interpretable model-agnostic explanation (LIME) is a technique developed to explain the predictions of complex models. It simplifies the overall complexity of the model and uses local linear models to explain the individual predictions. LIME analyzes the model's predictions through small perturbations of the input data and visualizes important features based on this analysis. SHAP (SHapley Additive explanations) is a technique for quantitatively evaluating the extent to which each feature contributes to the model's predictions. Red indicates a positive contribution, and blue indicates a negative contribution.

V. EXPERIMENT RESULTS

Fig. 3 shows the visualization results of the data, where (a) is the t-SNE and (b) is the UMAP. In each figure, the colorbar on the right side represents the neatness levels, with purple (0.0) indicating neat writing and yellow (1.0) indicating messy writing. The t-SNE visualization effectively

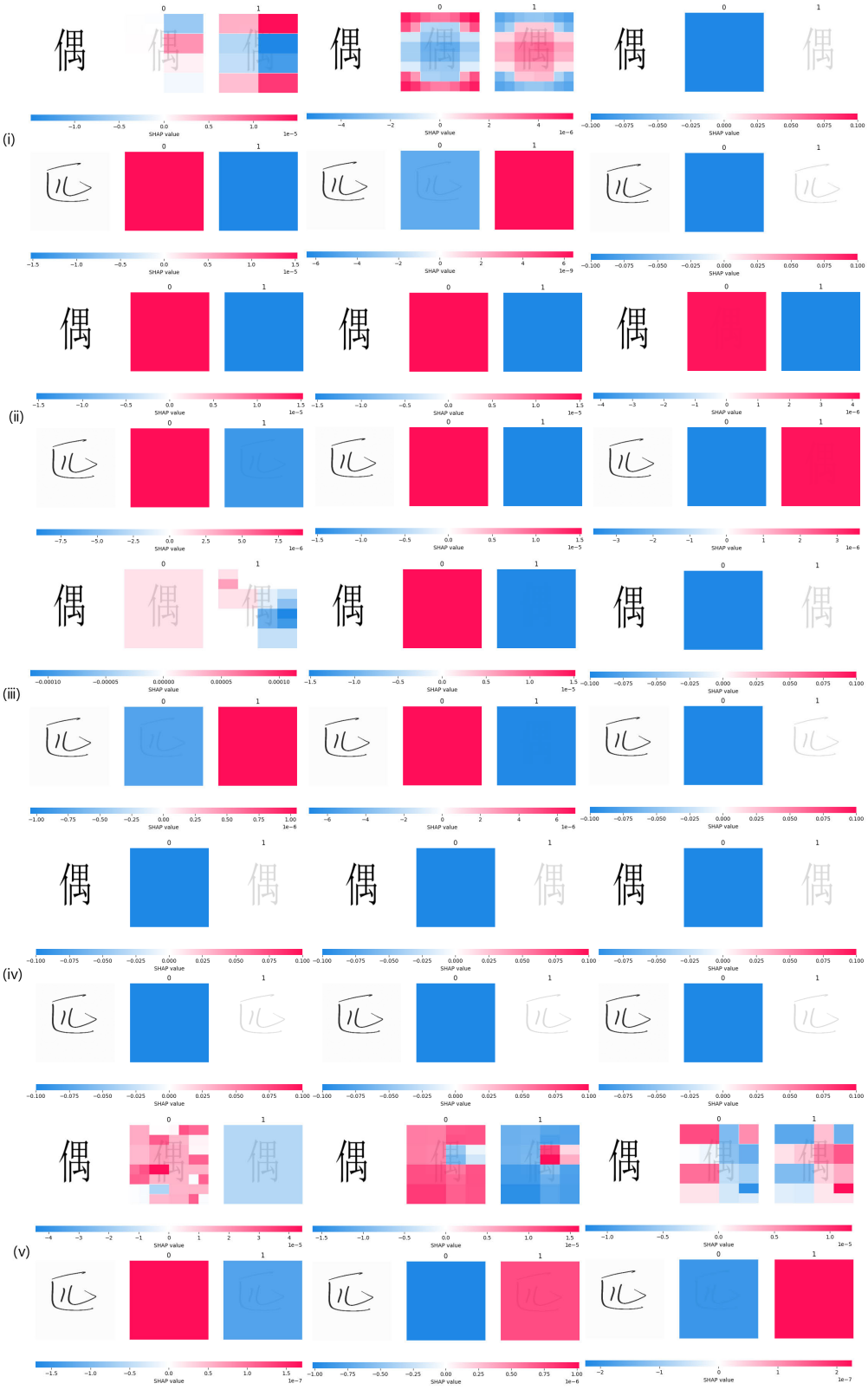

FIGURE 8. Visualization of SHAP results for comparative analysis of model performance, (i)-(a) CNN using Global, (i)-(b) CNN using FedAVG, (i)-(c) CNN using IPA, (ii)-(a) CNN using Our proposal, (ii)-(b) ResNet50V2 using Global, (ii)-(c) ResNet50V2 using FedAVG, (iii)-(a) ResNet50V2 using IPA, (iii)-(b) ResNet50V2 using Our proposal, (iii)-(c) NASNetLarge using Global, (iv)-(a) NASNetLarge using FedAVG, (iv)-(b) NASNetLarge using IPA, (iv)-(c) NASNetLarge using Our proposal, (v)-(a) EfficientNetV2B0 using Global, (v)-(b) EfficientNetV2B0 using FedAVG, (v)-(c) EfficientNetV2B0 using IPA.

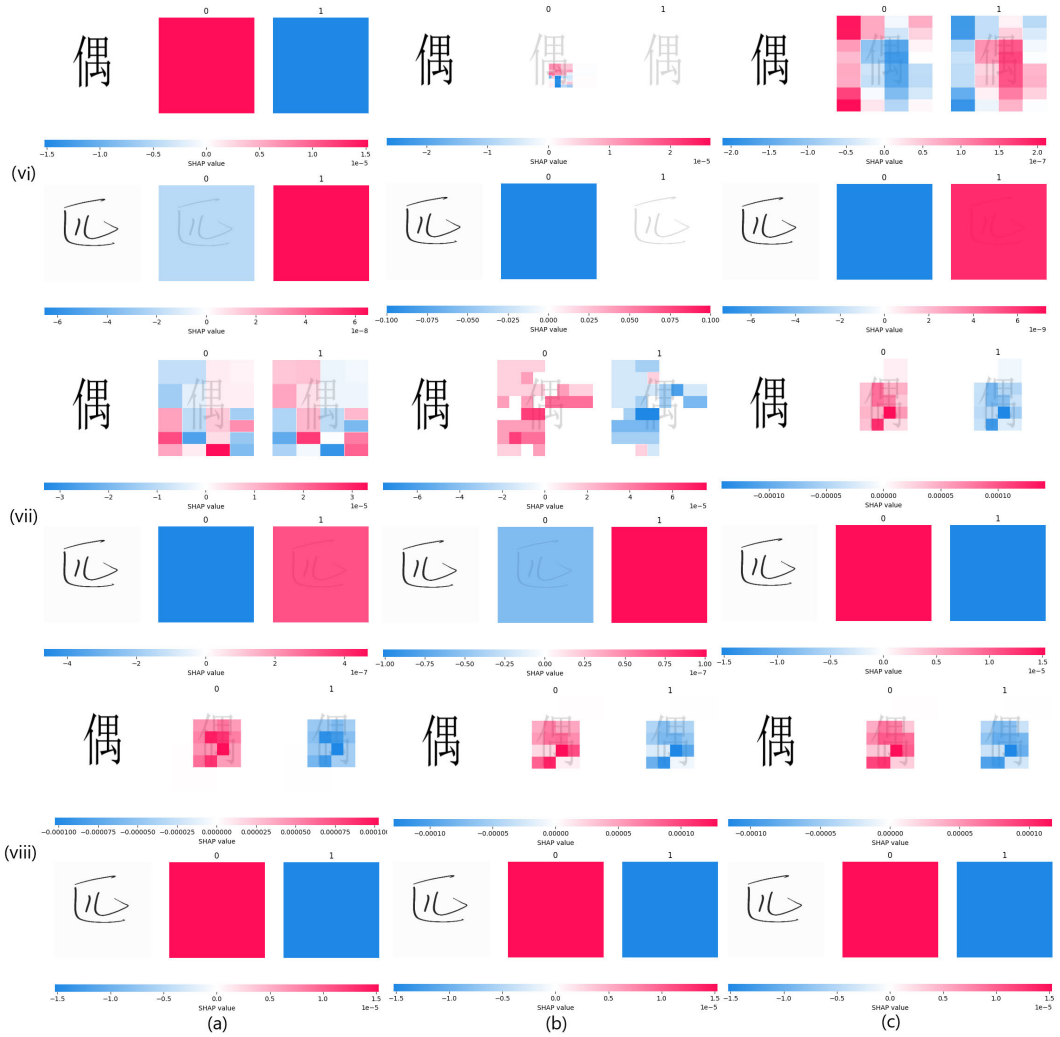

FIGURE 8. *(Continued.)* **Visualization of SHAP results for comparative analysis of model performance, (vi)-(a) EfficientNetV2B0 using Our proposal, (vi)-(b) EfficientNetB0 using Global, (vi)-(c) EfficientNetB0 using FedAVG, (vii)-(a) EfficientNetB0 using IPA, (vii)-(b) EfficientNetB0 using Our proposal, (vii)-(c) Swin Transformer using Global, (viii)-(a) Swin Transformer using FedAVG, (viii)-(b) Swin Transformer using IPA, (viii)-(c) Swin Transformer using Our proposal.**

displays class distributions with some overlapping regions. In contrast, UMAP reveals a more detailed clustering of the data and defines the boundaries between clusters more distinctly, providing a clearer overall structure.

The CCA graph in Fig. 4 shows the results of the analysis. The correlation coefficient, r is 0.99-, indicating an explanatory power of 0.980. Therefore, there was a very strong positive correlation. This implies that the relationship between the two classes was almost perfectly linear. Additionally, a p-value of 0 indicates that this correlation is highly statistically significant, implying that the probability of this result occurring by chance is extremely low. Observing the data distribution, the points formed an almost perfect diagonal line, suggesting that the characteristics of the two classes had similar patterns. The values on both the x-axis and y-axis range approximately between -3 and 3, indicating that the data have been standardized. Most data points are within ± 3 standard deviations of the mean. Finally, the red linear regression line almost perfectly matches the data points, again confirming that the relationship between the two classes is highly linear.

In Table 2, the t-statistic for Line Thickness is very high, with a p-value of 0, indicating a highly significant difference in line thickness between neat and messy writing. The t-statistic for curvature is close to 0, with a p-value greater than 0.05, suggesting no significant difference in curvature between the two writing styles. Both Spacing Consistency and Slope Consistency had very high t-statistic values and a p-value of 0, indicating significant differences in spacing and slope between neat and messy writing.

In Fig. 5, the data features are visualized using histograms. As seen in (a), neat writing images (10-30) have thicker lines than messy writing images (5-15), suggesting that messy writing was performed under inconsistent pressure. As shown

FIGURE 9. Visualizing LIME results for interpretable analysis of model performance, (i)-(a) CNN using Global, (i)-(b) CNN using FedAVG, (i)-(c) CNN using IPA, (ii)-(a) CNN using Our proposal, (ii)-(b) ResNet50V2 using Global, (ii)-(c) ResNet50V2 using FedAVG, (iii)-(a) ResNet50V2 using IPA, (iii)-(b) ResNet50V2 using Our proposal, (iii)-(c) NASNetLarge using Global, (iv)-(a) NASNetLarge using FedAVG, (iv)-(b) NASNetLarge using IPA, (iv)-(c) NASNetLarge using Our proposal, (v)-(a) EfficientNetV2B0 using Global, (v)-(b) EfficientNetV2B0 using FedAVG, (v)-(c) EfficientNetV2B0 using IPA.

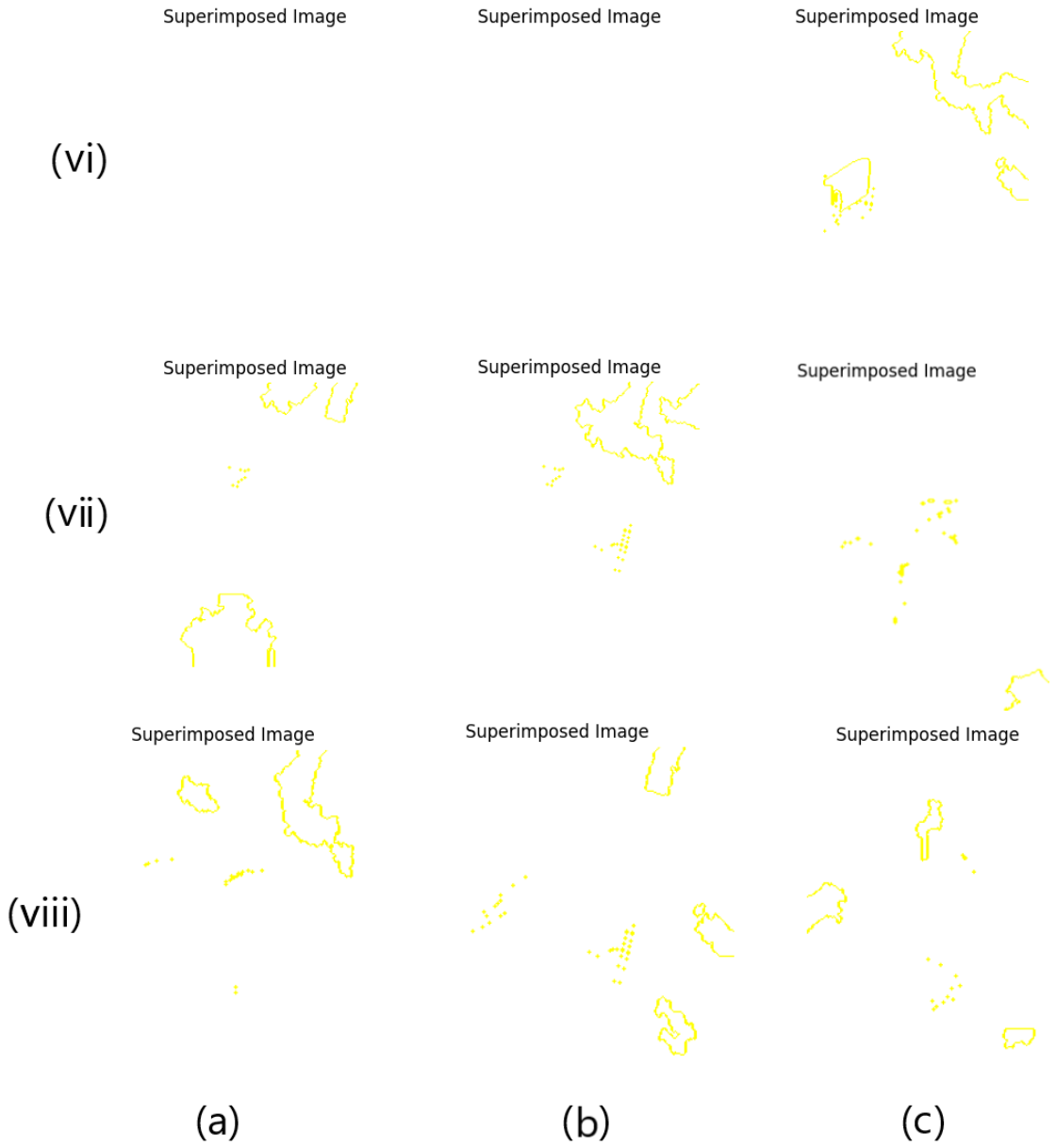

FIGURE 9. *(Continued.)* **Visualizing LIME results for interpretable analysis of model performance, (vi)-(a) EfficientNetV2B0 using Our proposal, (vi)-(b) EfficientNetB0 using Global, (vi)-(c) EfficientNetB0 using FedAVG, (vii)-(a) EfficientNetB0 using IPA, (vii)-(b) EfficientNetB0 using Our proposal, (vii)-(c) Swin Transformer using Global, (viii)-(a) Swin Transformer using FedAVG, (viii)-(b) Swin Transformer using IPA, (viii)-(c) Swin Transformer using Our proposal.**

in (b), both writing styles are distributed within a similar range (2-10). (c) shows spacing consistency, where neat writing is concentrated between 5,000 and 10,000, whereas messy writing is concentrated between 1,000 and 4,000, indicating that neat writing has more consistent spacing. In (d), neat writing is concentrated in a very narrow range (0.08-0.09), whereas messy writing is spread over a wider range (0.06-0.14), indicating greater variability in the angle for messy writing.

Overall, the line thickness and spacing exhibited significant differences between the neat and messy writing images, indicating that these are important distinguishing factors for the classification. The slope also shows a significant difference and can play an important role in distinguishing between neat and messy writing. In contrast, curvature did not show a significant difference between the two groups and thus did not have a major impact.

Table 3 presents the results of the six models and five training methods, with all performance values rounded to three decimal places. If the proposed method improves performance or is equal to the best, the performance is shown in bold. In terms of the AUC Score, the three models showed improved results when the proposed method was used. The CNN improved from 0.204 to 1.0, NASNetLarge

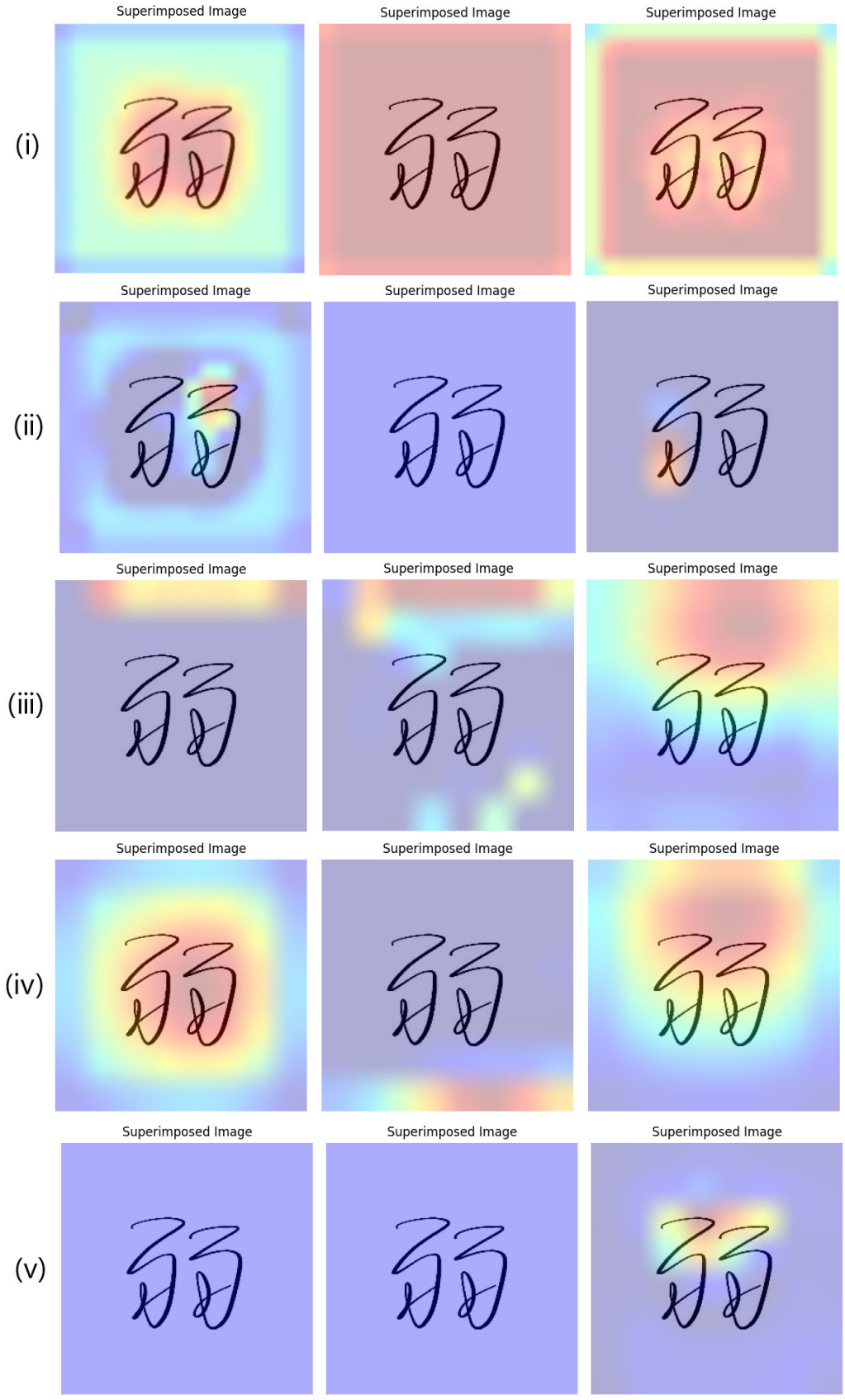

FIGURE 10. Visualizing Grad-gram results for explainable analysis of model performance, (i)-(a) CNN using Global, (v)-(a) EfficientNetV2B0 using Global, (v)-(b) EfficientNetV2B0 using FedAVG, (v)-(c) EfficientNetV2B0 using IPA.

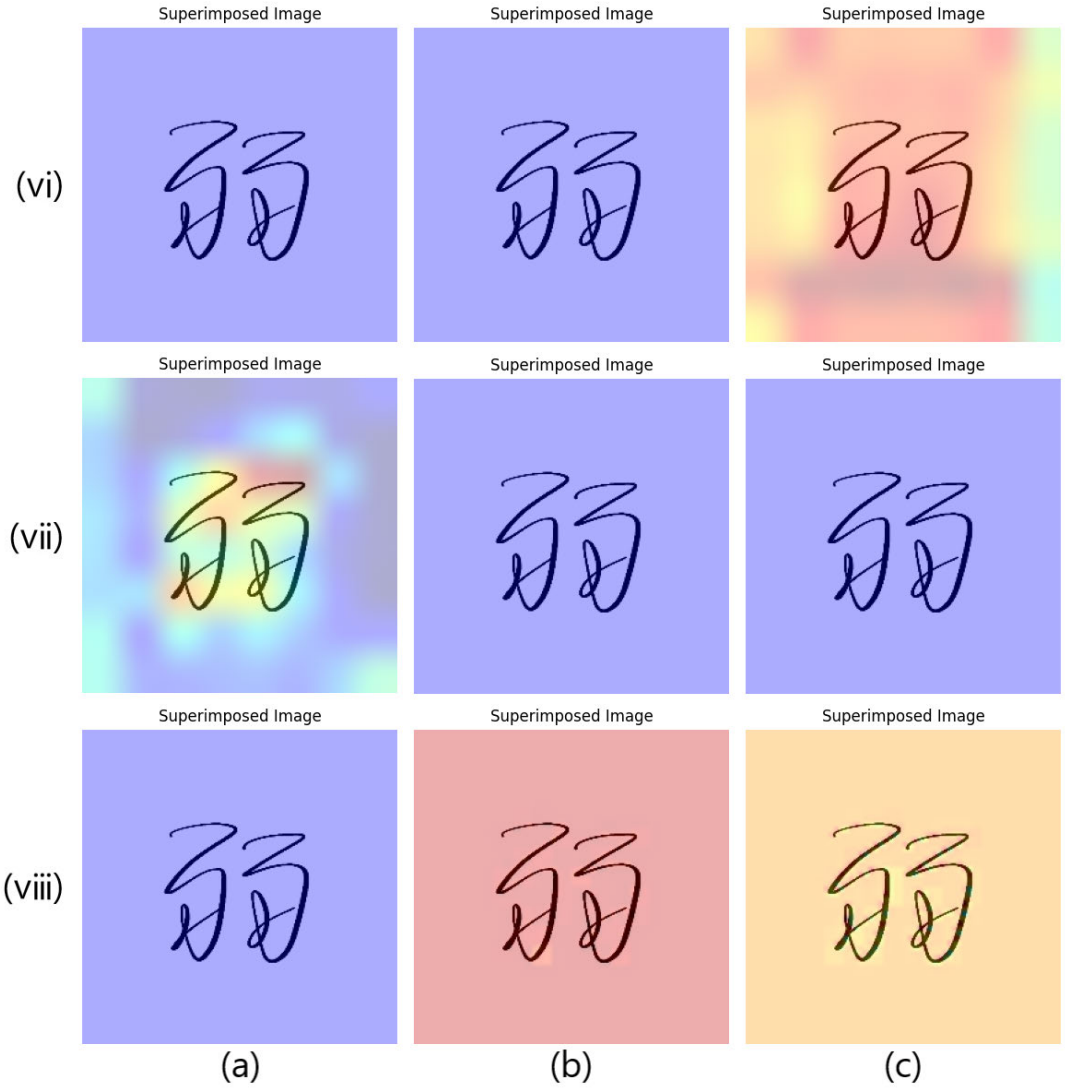

FIGURE 10. *(Continued.)* **Visualizing Grad-gram results for explainable analysis of model performance, (vi)-(a) EfficientNetV2B0 using Our proposal, (vi)-(b) EfficientNetB0 using Global, (vi)-(c) EfficientNetB0 using FedAVG, (vii)-(a) EfficientNetB0 using IPA, (vii)-(b) EfficientNetB0 using Our proposal, (vii)-(c) Swin Transformer using Global, (viii)-(a) Swin Transformer using FedAVG, (viii)-(b) Swin Transformer using IPA, (viii)-(c) Swin Transformer using Our proposal.**

increased from 0.229 to 0.5, and EfficientNetB0 increased from 0.414 to 1.0. For both the global and proposed methods, the other models maintained an AUC Score of 1.0 in both the Global and proposed methods. For the F1 Score, Recall, and Precision, both CNN and EfficientNetB0 exhibited higher values with the proposed method than with the global method. The CNN increased from 0.668 to 0.79, and EfficientNetB0 increased from 0.332 to 1.0. EfficientNetV2B0 maintained the same performance of 1.0 in both methods. In the test loss, both the CNN and EfficientNetB0 models exhibited significant improvement with the proposed method. The CNN decreased from 52.457 to 0.606, and EfficientNetB0 dropped from 7.36 to 0.0. EfficientNetV2B0 recorded a loss of 0.0 in both methods. Test accuracy showed the same results as the F1 Score, Recall, and Precision. The swin transformer maintained or increased the performance across all performance metrics.

The analysis in Table 3 indicates that the CNN and EfficientNetB0 models exhibit the most significant performance improvements with the proposed method. This suggests that the proposed method positively affects model performance. The proposed method can be considered an effective approach for maintaining or improving model performance.

Fig. 6 shows the results of the ROC Curve. The AUC Score is a key indicator of a model's classification performance, where a score of 0.5 indicates random guessing and a score of 1.0 indicates perfect classification. In Fig. 6 graphs (i)-(a), (i)-(b) and (i)-(c) show curves that exactly match the diagonal line, indicating poor model performance. The analysis

TABLE 4. Interpreting quantitative and qualitative analysis results in an integrated table.

Model	Federated Learning	Grad Cam	SHAP 0	SHAP 1	AUC Score	Count
CNN [7], [17]	Global	O	O	X	0.204	2
	FedAVG [11], [12]	X	X	X	0.237	0
	IPA [23]	X	X	X	0.500	0
	Our Proposal	O	X	X	1.000	1
ResNet50V2 [18]	Global	X	X	X	1.000	0
	FedAVG [11], [12]	X	X	X	0.500	0
	IPA [23]	X	X	X	0.999	0
	Our Proposal	X	X	X	1.000	0
NASNetLarge [19]	Global	X	X	X	0.229	0
	FedAVG [11], [12]	O	X	X	0.500	0
	IPA [23]	X	X	X	0.500	0
	Our Proposal	O	X	X	0.500	1
EfficientNetV2B0 [20]	Global	X	X	X	1.000	0
	FedAVG [11], [12]	X	O	X	0.994	1
	IPA [23]	O	X	X	0.715	1
	Our Proposal	X	X	X	1.000	0
EfficientNetB0 [21]	Global	X	X	X	0.414	0
	FedAVG [11], [12]	O	X	X	0.149	1
	IPA [23]	O	X	X	0.748	1
	Our Proposal	X	O	X	1.000	1
Swin Transformer [22]	Global	X	O	X	1.000	1
	FedAVG [11], [12]	X	O	X	1.000	1
	IPA [23]	X	O	X	1.000	1
	Our Proposal	X	O	X	1.000	1

showed that the performance of the proposed method was improved or maintained over that of the global method for CNN(ii-a), EfficientNetV2B0(vi-a), EfficientNetB0(vii-b), and swin transformer(viii-c).

Fig. 7 shows the results of the Confusion Matrix. As shown in Fig. 7 graph (ii)-(b), (v)-(a), (vi)-(d), (vii)-(b), (vii)-(c), (viii)-(a), (viii)-(b), and (viii)-(c), we can see that the darker the diagonal color, the better the model performs. However, as shown in graph (iii)-(c), (iv)-(a), (iv)-(b) and (iv)-(c), if the vertical colors are darker, the model's predictions are as accurate as those of the random guesses. Among the proposed methods (ii-, iii-b, iv-c, vi-a, vii-b and viii-c), there was only one graph with a dark vertical color; therefore our proposed method is the most effective.

Fig. 8 shows the SHAP results. The top part of the resulting images represents the SHAP values for the neat writing images, and the bottom part represents the SHAP values for the messy writing images. In Fig. 8 graph (vii)-(b), Fig. (vii)-(c), (viii)-(a), (viii)-(b) and (viii)-(c) the neat writing results show that the red and blue parts are precisely concentrated. However, in Fig. 8 graph (i)-(b), the neat writing results indicate that the model interprets the given image in the exact opposite manner. Fig. 8 graph (iii)-(c), (iv)-(a), (iv)-(b) and (iv)-(c) reveal that regardless of the learning method, the model struggles to recognize important features. For neat writing images, the features were recognized three times; however, for messy writing images, the features were not recognized.

Fig. 9 shows the LIME results. In Fig. 9 graph (iii)-(a), (vi)-(c), (vii)-(a), (vii)-(b), (vii)-(c), (viii)-(a), (viii)-(b) and (viii)-(c) the areas are highlighted in yellow. This indicates that the model accurately visualizes the important interpretable features. No significant results were found in most graphs.

Fig. 10 shows the Grad Cam results. Grad Cam is a technique that visually highlights the areas of an image on which the model focuses on making decisions. This technique allows us to understand the parts of the image that the model considers important. In Grad Cam, the areas marked in bright colors indicate parts that the model recognizes as significant, whereas dark areas signify parts that the model recognizes as less important. For example, in Fig. 10 graph (iv)-(a), the model identifies important features of the image. However, in ResNet50V2 (graph (ii)-(b), (ii)-(c), (ii)-(a) and (iii)-(b)) and swin transformer (graph (vii)-(c), (vii)-(a), (vii)-(b) and (vii)-(c)), the areas where the model focuses and where the real features are located did not match. This means that the model has difficulty recognizing important features, and this misalignment can lead to the degradation of the model's performance.

Table 4 provides a comprehensive summary of the Grad Cam, SHAP, and AUC score results for the six models

TABLE 5. Results of applying six models using existing and proposed methods to OCR recognition.

Model	Classification	Neat Writing	Messy Writing	Overall
None	Count	50	50	100
	Image=Letter	21(42%)	20(40%)	41(41%)
	Image ≠ Letter	12(24%)	15(30%)	27(27%)
	Not Recognised	17(34%)	15(30%)	32(32%)
CNN [7], [17]	Count	50	31	81
	Image=Letter	21(42.0%)	18(58.06%)	39(48.14%)(+7.14%)
	Image ≠ Letter	12(24.0%)	8(25.8%)	20(24.69%)(-2.31%)
	Not Recognised	17(34.0%)	5(16.12%)	22(27.16%)(-4.84%)
ResNet50V2 [18]	Count	24	0	24
	Image=Letter	7(29.16%)	0	7(29.16%)(-11.18%)
	Image ≠ Letter	6(25%)	0	6(25%)(-2.0%)
	Not Recognised	11(45.83%)	0	11(45.83%)(+13.83%)
NASNetLarge [19]	Count	0	0	0
	Image=Letter	0	0	0
	Image ≠ Letter	0	0	0
	Not Recognised	0	0	0
EfficientNetV2B0 [20]	Count	50	0	50
	Image=Letter	21(42.0%)	0	21(42.0%)(+1.0%)
	Image ≠ Letter	12(24.0%)	0	12(24.0%)(-3.0%)
	Not Recognised	17(34.0%)	0	17(34.0%)(+2.0%)
EfficientNetB0 [21]	Count	50	0	50
	Image=Letter	21(42.0%)	0	21(42.0%)(+1.0%)
	Image ≠ Letter	12(24.0%)	0	12(24.0%)(-3.0%)
	Not Recognised	17(34.0%)	0	17(34.0%)(+2.0%)
Swin Transformer [22]	Count	50	0	50
	Image=Letter	21(42.0%)	0	21(42.0%)(+1.0%)
	Image ≠ Letter	12(24.0%)	0	12(24.0%)(-3.0%)
	Not Recognised	17(34.0%)	0	17(34.0%)(+2.0%)

based on specific criteria. Grad Cam, LIME, and SHAP, Grad Cam, and SHAP were selected because of their visual interpretability. Local was excluded from the table, because there were no Grad Cam or SHAP results for this method. Grad Cam is marked with an O if the heatmap focuses on the characters and with an X otherwise. For SHAP, an O is assigned if red blocks appear on the characters corresponding to class 0 and blue blocks appear on the characters corresponding to class 1 when an image of class 0 is given; otherwise, X is assigned. The "Count" indicates the number of O marks.

For the CNN model, the AUC Score for our proposal was the highest at 1.0, but the count for Global was two, and for our proposal, it was one. ResNet50V2 had counts of 0, although the AUC Score for our proposal was 1.0. NASNetLarge had the same AUC Score of 0.5 for FedAVG, IPA, and our proposal, but the count was the highest at 1. EfficientNetV2B0 had the highest AUC Score of 1.0 for both Global and our proposal. EfficientNetB0 had a count of 1 for all methods except Global, with the highest AUC Score of 1.0 for our proposal. Swin transformers showed the same results for all methods.

As shown in Table 4, the proposed method exhibited an overall superior performance. To validate this, two types of data were input into the OCR system: the original data without any model processing and the data classified as neat writing by the model.

Table 5 summarizes the results when inputting data into the OCR system without using a model and using a model. 'None' includes 50 images of neat writing and 50 images of messy writing. The recognition rate, which indicates the percentage of recognized characters that matched actual characters, was 41%. The mismatch rate, which indicates the percentage of recognized characters that differ from the actual characters, was 27%. The non-recognition rate, where the OCR system failed to recognize any characters, was 32%. The models used for data classification were CNN, ResNet50V2, NASNetLarge, EfficientNetV2B0, EfficientNetB0 and swin transformer. Only data classified as neat writing by the model were input into the OCR system. Comparisons with 'None' are marked in red for increases and in blue for decreases. The recognition rates were as follows: CNN achieved 48.14%, ResNet50V2 achieved 29.16%, EfficientNetV2B0, EfficientNetB0, and Swin Transformer achieved 42%. NASNetLarge classifies all data as messy writing; therefore, no results can be obtained from the OCR system.

Using six models to classify the data and check the recognition rate with the OCR system, we found that four models showed an increase in the recognition rate compared to when no model was used. Overall, compared with Table 5, the recognition rate increased, whereas the mismatch and non-recognition rates decreased. This demonstrates that the proposed method improved the recognition rate of the OCR system.

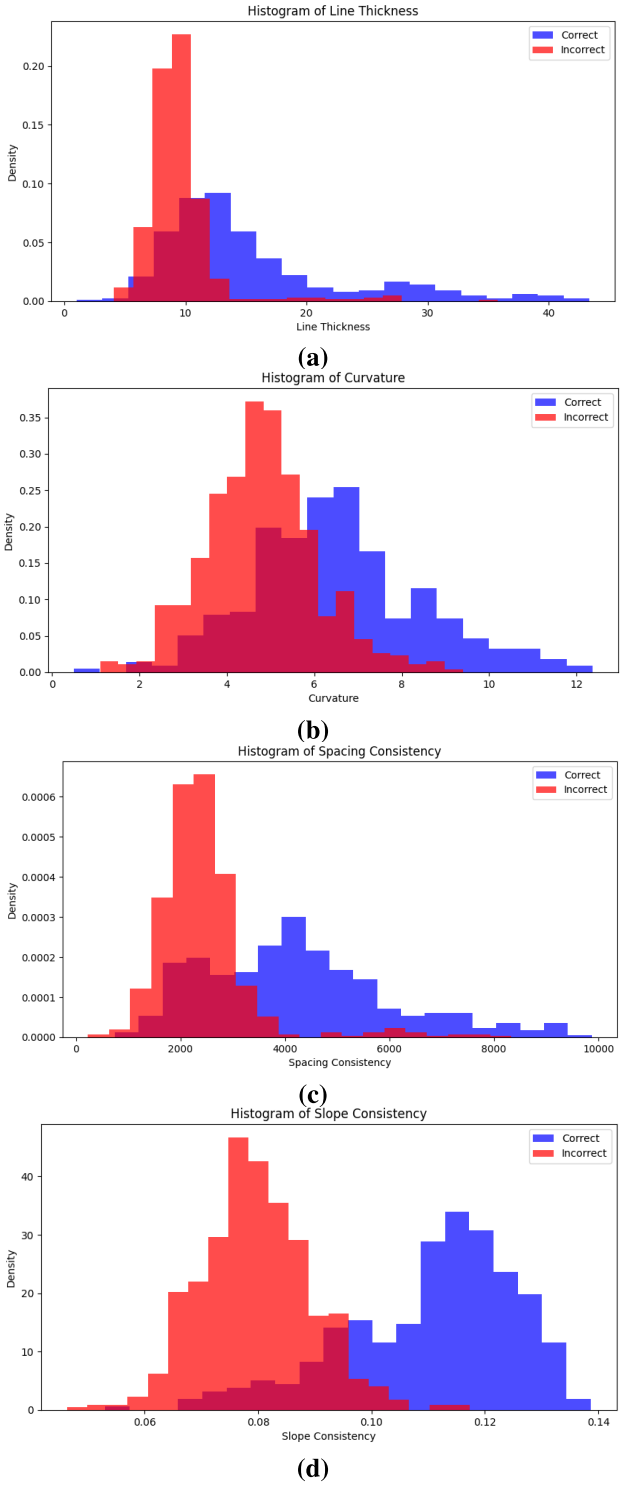

FIGURE 11. Histogram visualization analysis for properly classified messy writing vs. poorly classified messy writing. (a) Line Thickness, (b) Curvature, (c) Spacing Consistency, and (d) Slope Consistency.

Fig. 11 shows histograms for Line Thickness, Curvature, Spacing Consistency, and Slope Consistency of messy writing that was correctly classified versus messy writing that was incorrectly classified. In the case of incorrect classifications, the lines tended to be thinner, the curvature was lower, the spacing consistency between characters was reduced, and the slope consistency was lower.

VI. CONCLUSION

Chinese characters have established themselves as a crucial means of communication in East Asian culture. However, the accuracy of optical character recognition (OCR) systems deteriorates when messy written. To address the issue of recognizing complex Chinese characters and errors caused by messy writing, we propose a Chinese character messy writing classification program.

We analyzed the features of Chinese characters using various techniques and proposed several deep learning models using a new federated learning approach. In the data collection phase, we gathered and preprocessed the data for training. The models used for training included CNN, ResNet50V2, NASNetLarge, EfficientNetV2B0, EfficientNetB0 and Swin Transformer. We applied different federated learning methods global, local, FedAVG, IPA and our proposed method to evaluate the performance of each model. We then assessed the match rates between the images and the recognized characters in the OCR.

By analyzing t-SNE, UMAP, and CCA, we identified the feature differences between neat and messy writing data. Specifically, stroke thickness, spacing consistency, and slope consistency were found to be crucial distinguishing features between the two groups. Among the six deep learning models evaluated with the five learning methods, CNN and EfficientNetB0 showed performance improvements through federated learning, suggesting that our proposed method positively affects impact model training. Finally, we compared the match rates of the images and the recognized characters with and without using the model, showing a maximum increase of 7.14%.

When we analyzed misclassified messy writing, we found that it tended to have thin lines, low curvature, inconsistent spacing between characters, and low slant consistency.

Our approach not only classifies messy writing but can also be expanded into a handwriting correction system in future research. Such a system can provide real-time feedback by recognizing a user's handwriting and converting it into a more accurate form. This would be beneficial for both educational purposes and practical applications such as document digitization.

Moreover, a handwriting correction system could be useful in the existing document digitization processes. For example, when digitizing historical documents or personal records in messy writing, accurate OCR can be challenging. Correcting messy writing to achieve accurate digitization would facilitate the preservation and utilization of important documents.

Future studies should focus on the following aspects. First, we built a comprehensive dataset that included various handwriting styles and document types to improve the

accuracy of the handwriting corrections. Second, it enhances the real-time feedback capabilities to help users instantly correct their handwriting using improved interfaces. Third, a handwriting correction system for various platforms, such as mobile devices and tablets, is required to increase accessibility.

In addition, the development of personalized handwriting correction and learning programs using artificial intelligence should be explored. For example, users' handwriting habits and error patterns can be analyzed to provide individualized correction feedback and to assist in improving their handwriting. Such systems can be applied to fields such as education, business, law, and healthcare.

Finally, continuous evaluation and improvement of handwriting correction systems through active user feedback are necessary. This enhances the reliability and efficiency of the system by providing tailored solutions that meet diverse user needs.

Through these efforts, the Chinese character messy writing classification and correction system could evolve beyond mere technical tools and become a valuable asset in education and document management, bringing innovative changes to Chinese character education and handling.

REFERENCES

[1] X. He, B. Zhang, and Y. Long, "Increasing offline handwritten Chinese character recognition using separated pre-training models: A computer vision approach," *Electronics*, vol. 13, no. 15, p. 2893, Jul. 2024.

[2] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark," *Pattern Recognit.*, vol. 61, pp. 348–360, Jan. 2017.

[3] F. Min, S. Zhu, and Y. Wang, "Offline handwritten Chinese character recognition based on improved googlenet," in *Proc. 3rd Int. Conf. Artif. Intell. Pattern Recognit.*, New York, NY, USA, Jun. 2020, pp. 42–46.

[4] K. Gyohten, H. Ohki, and T. Takami, "A method to identify the cause of misrecognition for offline handwritten Japanese character recognition using deep learning," in *Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods*, Valletta, Malta, 2020, pp. 446–452.

[5] D. Vijaya, D. M. R. Panda, D. A. Babu, S. L. Yadav, and D. Nidhya, "Machine learning algorithm to detect hand written character recognition," *Migrat. Lett.*, vol. 20, no. S13, pp. 549–559, Dec. 2023.

[6] L. Xu, Y. Wang, X. Li, and M. Pan, "Recognition of handwritten Chinese characters based on concept learning," *IEEE Access*, vol. 7, pp. 102039–102053, 2019.

[7] Y. Zhuang, "A handwritten Chinese character recognition based on convolutional neural network and median filtering," *J. Phys., Conf. Ser.*, vol. 1820, no. 1, Mar. 2021, Art. no. 012162.

[8] Y. Zhao, X. Zhang, B. Fu, Z. Zhan, H. Sun, L. Li, and G. Zhang, "Evaluation and recognition of handwritten Chinese characters based on similarities," *Appl. Sci.*, vol. 12, no. 17, p. 8521, Aug. 2022.

[9] S. Bhowmik, S. Risat, and B. Sarkar, "DSANet: Dilated spatial attention network for the detection of text, non-text and touching components in unconstrained handwritten documents," *Neural Comput. Appl.*, vol. 36, no. 27, pp. 16959–16976, Sep. 2024.

[10] S. Bhowmik, "Utilization of relative context for text non-text region classification in offline documents using multi-scale dilated convolutional neural network," *Multimedia Tools Appl.*, vol. 83, no. 9, pp. 26751–26774, Sep. 2023.

[11] Y. Zhao, "Comparison of federated learning algorithms for image classification," in *Proc. 2nd Int. Conf. Data Anal., Comput. Artif. Intell. (ICDACAI)*, Zakopane, Poland, Oct. 2023, pp. 613–615.

[12] O. B. Tanmoy, M. A. Mamun, S. Hasan, and A. Anwar, "Enhancing federated learning with globally shared model: A modified FedAVG approach (GSM-FedAVG)," in *Proc. 6th Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Khulna, Bangladesh, Dec. 2023, pp. 1–6.

[13] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 618–626.

[14] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Jan. 2017, pp. 4768–4777.

[15] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," *Neurocomputing*, vol. 597, Sep. 2024, Art. no. 128019.

[16] S. Ji, Y. Tan, T. Saravirta, Z. Yang, Y. Liu, L. Vasankari, S. Pan, G. Long, and A. Walid, "Emerging trends in federated learning: From model fusion to federated x learning," *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 9, pp. 3769–3790, Sep. 2024.

[17] A. A. Nafea, S. A. Alameri, R. R. Majeed, M. A. Khalaf, and M. M. Al-Ani, "A short review on supervised machine learning and deep learning techniques in computer vision," *Babylonian J. Mach. Learn.*, vol. 2024, pp. 48–55, Feb. 2024.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The, Netherlands, Jan. 2016, pp. 630–645.

[19] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8697–8710.

[20] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," 2021, *arXiv:2104.00298*.

[21] S. H. Emon, M. A. H. Mridha, and M. Shovon, "Automated recognition of Rice grain diseases using deep learning," in *Proc. 11th Int. Conf. Electr. Comput. Eng. (ICECE)*, Dhaka, Bangladesh, Dec. 2020, pp. 230–233.

[22] Y. Zheng, Y. Chen, X. Wang, D. Qi, and Y. Yan, "Ancient Chinese character recognition with improved Swin-transformer and flexible data enhancement strategies," *Sensors*, vol. 24, no. 7, p. 2182, Mar. 2024.

[23] S. H. Choi, "AI-connect protocol: A new federated deep learning method for user-centric AI service," in *Proc. Korean Soc. Intell. Inf. Syst.*, Daegu, South Korea, 2022, pp. 69–70.

MIN-SUN KIM is currently pursuing the bachelor's degree in AI application with Hansung University, Seoul, South Korea.

CHANG-HO SON received the B.S. degree from Korea Military Academy (KMA), in 2002, the M.S. degree from North Carolina State University (NCSU), in 2006, and the Ph.D. degree from Seoul National University (SNU), in 2012. He is a Professor with the Department of System Engineering, Korea Army Academy at Youngcheon (KAAY).

SEOUNG-HO CHOI received the B.S. and M.S. degrees from Hansung University, South Korea, in 2018 and 2020, respectively. He joined the Faculty of Basic Liberal Art, College of Liberal Art, Hansung University, in 2021, where he is an Assistant Professor.