CCR-LWECNN: A Lightweight CNN Framework for Chinese Calligraphy Recognition and Evaluation

Xi Chen*, Jing Zhao
Zhongyuan Institute of Science and Technology, Zhengzhou, Henan, 461000, China
*Corresponding author
E-mail : chenxi19891028@126.com

Keywords: character, CCR-LWECNN Chinese calligraphy recognition, deep learning, image processing system

Received: April 3, 2025

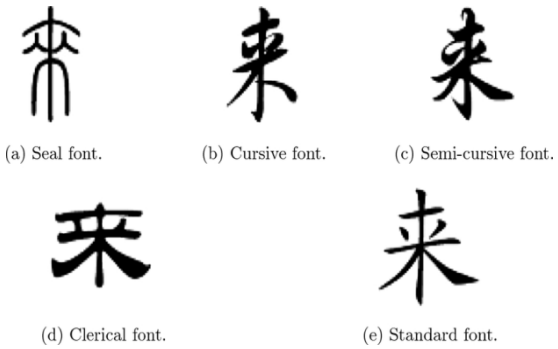
This study presents a lightweight enhanced CNN architecture (CCR-LWECNN) for Chinese calligraphy recognition, addressing the challenges of multi-class classification across 12,152 labeled images spanning 960 Chinese characters in five calligraphic styles. Unlike previous studies limited to small character sets and single recognition approaches, this research integrates character recognition with image processing techniques. Data augmentation using TensorFlow's Image Data Generator—applying rotation and zoom—was employed to improve class balance and variety. The proposed model, comprising five convolutional and three fully connected layers, processes 224×224-pixel images and leverages pretraining for robust feature extraction. CCR-LWECNN achieved superior performance with 96.5% accuracy, 95.6% precision, 95.2% recall, and 95.6% F1-score, outperforming baseline models such as traditional CNN (90.5%), SVM (85.2%), and Random Forest (75.4%). By effectively mitigating overfitting and underfitting through dropout layers and augmentation, this approach advances automated Chinese calligraphy recognition and provides a scalable solution for real-world applications.

Povzetek:.

1 Introduction

Characters in Chinese calligraphy are made up of a lot more strokes than those in Western calligraphy [1]. A single letter in Chinese calligraphy can be made up of as few as one stroke or as many as thirty. Before writing begins, the ink is absorbed by dipping and then used to produce strokes with a soft hairbrush. Different styles are produced as the calligrapher writes the character by varying the brush's pressure, speed, and direction [2]. Regular, clerical, cursive, semi-cursive, and seal are the most often used styles. These styles go under several names. For instance, referred to the semi-cursive style as the running style. The naming scheme employed by author will be applied in this study [3]. Beginning with a single style is beneficial for Chinese calligraphy students. The student might advance to another style after they are proficient at writing several characters in that style. An ancient art style that originated in China, Chinese calligraphy is also well-liked in a number of other nations, including South Korea, Japan, and Thailand. Using a brush and ink, Chinese calligraphy artists create visually appealing and well-composed characters. Chinese calligraphy offers advantages in addition to being a highly regarded art form [4].

Character recognition has emerged as a hotspot for computer vision research as picture digitisation advances, and it has significant applications in data entry for paper documents. Because handwriting characters have more irregular shapes than printed documents, it is more difficult to recognise handwriting. Chinese calligraphy is a sort of handwriting art form that consists of five main font type [5]. Figure 1 shown by Chinese calligraphy different font type.



(a) Seal font. (b) Cursive font. (c) Semi-cursive font.

(d) Clerical font. (e) Standard font.

Figure 1: Chinese calligraphy different font type

However, many find it difficult to instantly identify the content of calligraphy works since the shapes of the letters in Chinese calligraphy vary widely across calligraphers

and differ substantially from conventional fonts used in daily life. Therefore, by presenting the font and textual content of the input calligraphy image, a real-time calligraphy recognition system can aid amateur calligraphers in understanding calligraphy works [6]. Instead of manually typing out the text, the method may also be used to digitise calligraphy by just entering the image of the piece. In this study, we developed and put into use a convolutional neural network-based calligraphy recognition system. Compared to earlier research, the system has higher accuracy rates for identifying both typeface and textual content. We created a dataset of calligraphy characters to train the network, and we tested the viability of the system using pictures of various calligraphy pieces [7].

1.1 Challenges in Chinese calligraphy recognition

Chinese calligraphy is a difficult art form because of its many Chinese characters, many styles, and intricacy [8]. Since art evaluation is subjective and can have a detrimental effect on teacher-student relationships, it might be challenging to find qualified calligraphers and offer comments. Artificial intelligence (AI) can assist in overcoming these obstacles by offering unbiased assessments and comments. But only tiny groups of upto300 Chinese characters—roughly 8–12.4% of the 2500 characters used every day—can be recognised by ReLU models. Furthermore, there aren't many examples from old Chinese calligraphy masters, thus additional training sample photos are required. There is a need for more research because calligraphy is only mentioned in one empirical study on AI in education.

1.2 Contribution of this study

The three primary forms of Chinese calligraphy—character recognition, calligraphy production and simulation, and calligraphy analysis—represent an important field of study deep learning (DL). To enhance Chinese character and image processing technology, this study blends dropout in CNN hidden layers, data augmentation methods, and CNN architecture. The suggested approach CCR-LWECNN allows for greater accuracy without requiring additional training photos by recognising more than 960 Chinese characters in five calligraphic forms. Other languages can also be added to the model. In order to assist in this paper to monitor their progress during practice sessions. Related works, datasets, methods, findings, implications, discussion, and conclusions are all included in the parts that make up the study.

2 Literature review

Table 1 shows Summary of works

Table1: Summary on related works

Ref	Methods Used	Dataset Size	Baseline & Accuracy	Proposed Method & Accuracy	Key Findings
[9]	CNN, TensorFlow	Not specified	Traditional OCR 80%	CNN + TensorFlow 93.7%	CNN significantly improves recognition for handwritten characters
[10]	Hybrid CNN + Attention + Distillation	20,000+ images	Basic CNN 87.5%	Proposed 91.8%	Attention helps in distinguishing subtle calligraphic variations
[11]	MobileNet, CNN	~12,000	Tesseract OCR 76.2%	MobileNet 90.1%	Suitable for lightweight deployment in mobile/web
[12]	Deep CNN, CAI	Not given	Classic CNN 84.6%	Proposed hybrid 89.2%	Integration of CAI improves learning and recognition efficacy
[13]	CNN with Deep Stroke Extraction	~8,000	Hand-crafted stroke features 78.4%	Proposed 91.0%	Deep stroke analysis provides structural and aesthetic insight
[14]	5-layer CNN	~6,500	SVM 83.2%	CNN 92.4%	CNN better handles degraded or stylized historical samples

[15]	Traditional CNN + Filters	Not stated	Template Matching 74.8%	CNN 88.6%	CNN adapts better to style variance than traditional methods
[16]	Faster R-CNN, YOLOv3	10,000+	SSD 90.3%, YOLOv3 91.5%	Faster R-CNN 95.1%	Accurate segmentation and detection for full-page manuscripts

3 Methodology

3.1 Dataset

In order to construct the style recognition model, we used CCR-LWECNN models, which represent datasets and image pre-processing. The character recognition model is constructed via data augmentation and picture pre-processing. Kaggle's "Chinese calligraphy characters image set" serves as the training dataset for the image recognition model [17] we provide these resources will be made available via a public GitHub repository:https://github.com/zhuojg/chinese-calligraphy-dataset. 2890 calligraphy pictures totaling 960 characters were collected from various calligraphers and made available to the public. These pictures are labeled as semi-cursive, regular, seal, cursive, or clerical. We employed the oversampling approach because of the dataset's label imbalance issue. Additionally, this analysis demonstrated that overfitting would not result from oversampling. A far larger dataset was required for the image processing model than for style recognition. This is due to the fact that each character to be categorized belongs to a single output class in this multiclass classification model. There would have been just 2890 training photos for 960 classes if we had utilized the same dataset for style recognition. That would imply that there would typically be no more than three pictures each word. We needed to figure out how to get more training photos. To expand the dataset's picture count for character recognition, we employed two strategies. Adding pictures from a public domain collection was the initial technique. An online database of the Humanities & Social Sciences Database Catalogue contained the dataset's URL (Humanities & Social Sciences Database

Catalogue, 2023). We crawled the page and gathered photos using the Kaggle connection. However, the link was broken when this paper was written. Following the addition of pictures from this dataset, the final dataset comprised 12,152 training photos, with at least 10 images for each Chinese word. The train_test_split () function in the Scikit-learn package's data preparation module was used to divide these photos into training and testing sets. The sorted Data folder included the training set. Since there were just five styles in the output class, the dataset's photos were adequate for style recognition. However, a second technique was employed to enhance the number of training photos since we required to increase the number of images per Chinese character for character recognition. Utilizing data augmentation was the second strategy. During the training phase, we rotated and zoomed in on the already example photos using TensorFlow's Image Data Generator function to produce more sample images.

The dataset was constructed by combining photos from the Humanities & Social Sciences collection with the Kaggle set. Hash-based comparison methods were used to find and eliminate duplicate photos in order to guarantee quality. Additionally, physical inspection and simple picture quality checks (e.g., resolution thresholding and contrast analysis) were used to filter out low-quality samples, such as blurred, low-resolution, or severely distorted images. A clean and varied dataset for efficient model training was guaranteed by this preparation.

Data Augmentation: Random rotation (±15°), zoom (10-20% scale variation), brightness modification (±20%), and horizontal flipping (50% probability) were used by the data augmentation process to enhance sample variety. In order to replicate natural stroke fluctuations, we used 8x8 grid warping with σ=4 for elastic distortions. These parameters were chosen to increase the effective training dataset 5-fold without creating unreal artefacts, all the while maintaining calligraphic integrity. Bilinear interpolation was used to preserve stroke continuity throughout the real-time implementation of all transformations using TensorFlow's Image Data Generator.

3.2 Feature extraction

In recent years, deep learning has been widely applied in tracking, object identification, and other domains. By integrating low-level characteristics to create high-level features that represent the scattered aspects of data, it simulates how the human brain functions. Usually, the Light Weight Enhanced traditional CNN is used directly for image classification. Utilizing CNN's numerous advantages in feature extraction is the aim of this work. Compared to explicit feature extraction, digital feature extraction produces more detailed feature data for Chinese picture works.
The CNN theoretical framework-based CCR-LWECNN model was pretrained using the Kaggle dataset to extract

the visual attributes of Chinese calligraphy. The model is a feed-forward neural network with the model has two convolutional layers, not five, with one fully connected layer of 512 neurons and an output layer three fully connected layers. The model resizes the input image to 224 by 224 pixels in order to produce a 4096-dimensional feature vector. Feature Extraction is presented as a pretrained feature extractor producing a 4096-dimensional vector for further classification, suggesting a two-step pipeline. The proposed model, pretrained on real images, may be used to extract characteristics from Chinese calligraphy. First of all, Chinese calligraphy characters are an artistic reworking of natural surroundings and another depiction of a natural image. Second, the deep structure of the CCR-LWECNN model may extract complex structures from rich perceptual input and generate intrinsic representations in the data. More than 10 million natural photos are being utilized for training in order to gain relevant information for Chinese calligraphy feature extraction. Chinese character-like feature information will be included in the recovered features either directly or indirectly. Last but not least, the study's training dataset does not contain enough Chinese writing pieces to adequately train the suggested model. Nevertheless, this study uses it as a forerunner to the CNN model, which is lightweight so that the components it extracts may better capture the artistic character of Chinese calligraphy recognition.

3.3 Model explanation with CNN

In Figure 2, the framework that extracts the key characteristics of calligraphy recognition consists of two convolutional layers. The framework has a single, completely linked layer that can identify the style of a picture. The first convolutional layer of our suggested model has 32 filters, each of which has three channels and is 3 x 3. We employ the same padding, which means that the input images are zero-padded so that the filters overlap each pixel. We employed ReLU as the activation function in the convolution layer. Batch normalization is used to enhance model stability and performance. The maximum pooling filter has a dimension of 2 × 2 and travels with a stride of 2. It carries out the max pooling procedure on the feature maps. To assist keep this model from overfitting to the training data, we have implemented a dropout layer to shake off the neurons. The dropout value for the first convolution layer is set at 0.20. The second convolutional layer is created using 64 filters, each of which has three channels and is 3 × 3. The same cushioning is employed here as well. The ReLU activation function is also applied to the feature maps. Once more, batch normalization is utilized in the second layer to enhance model performance. The max pooling filter is 2x2 in size, advances by a stride of 2, and performs the max pooling operation on the feature maps. To address the issue of overfitting, a dropout value of 0.25 has been chosen for the second convolution layer. A flatten layer has been employed after the second convolutional layer's dropout value has been set. The outcome of the last pool layer is a victory type, and a fully linked layer with 512 neurons comes after it. The final features are then classified into many classes in the output layer using fully linked layers that were taken from the previous pooling and convolution layers. The completely linked layers learn from features. Batch normalization has once again been used. The dropout value of 0.5 was then applied. We once more employed ReLU for the activation function in the dense layer. Lastly, there are six nodes in the output layer, each of which represents six classes. Next, we classified the desired label in the output layer using the softmax activation function. Figure 2aand 2b Showed in Architecture diagram with dimensionalflows.

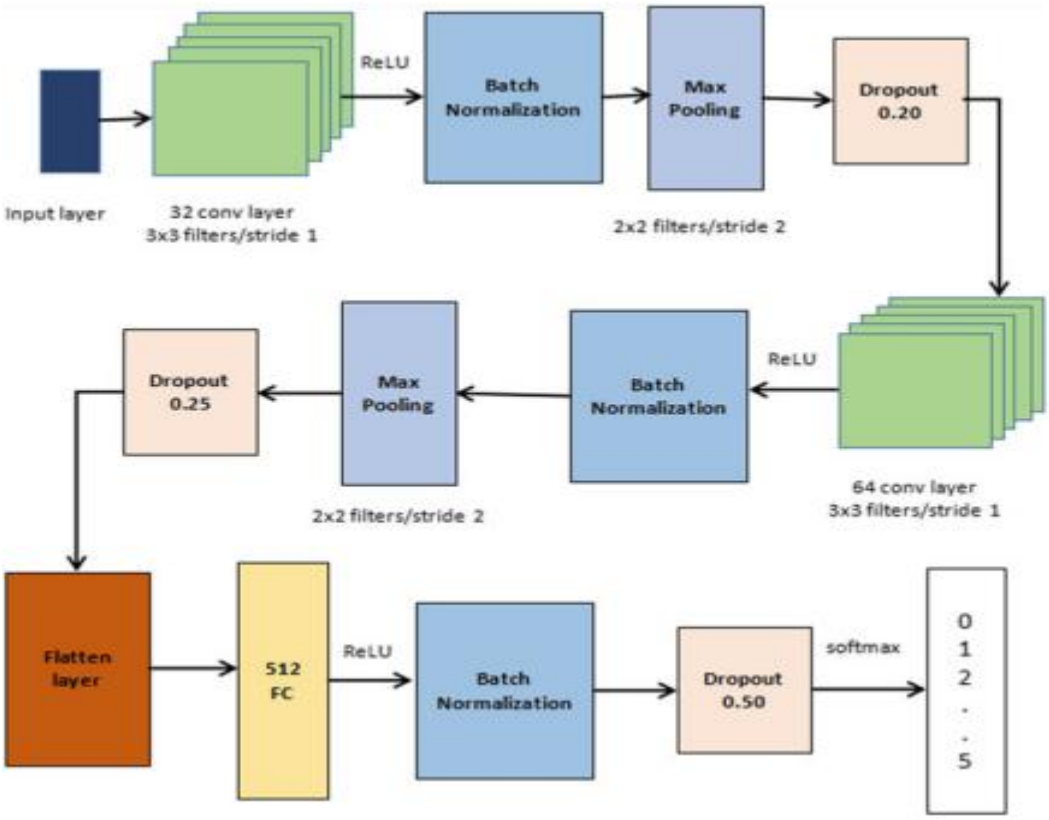


Figure 2a: Image style recognition model

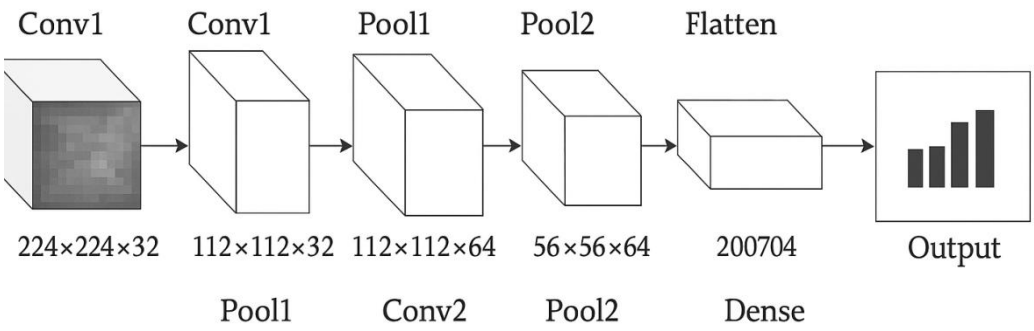


Figure 2b: Architecture diagram with dimensional flow

3.4 Chinese calligraphy recognition based on lightweight enhanced CNN algorithm (CCR-LWECNN)

Convolutional neural network (CNN) technology is a type of neural network that is specifically designed to process images. Since its inception, the technology has seen significant development. As a result, CNN has greatly aided people in processing visual information. However, this technology's computationally demanding approach also restricts its use in a number of industries. Therefore, the primary research goals in the current image recognition sector are to lower the computational cost of CNN and decrease the calculation time, optimise the technology thoroughly, and emphasize its contribution to image recognition technology for Chinese calligraphy recognition.

CNN is an effective recognition method and a type of neural network that mimics the visual structure of biology. Convolutional, pooling, and fully connected layers are the primary components of this recognition system. One of CNN's primary functions is the convolution operation of the convolutional layer. The following illustrates the convolution computation of continuous functions:

$$s(t) = \int x(a)w(t-a)\,\mathrm{d}a \quad (1)$$

Equation (1) uses x and w to stand for integrable functions, a and t for distinct computational components, and d for the convolution operation. The following illustrates how discrete functions are calculated using convolution:

$$s(n) = \sum_m \mathrm{r}[m]v\,,(n-m) \tag{2}$$

Discrete functions are represented by r and v in Equation (2), whereas calculation elements are represented by m and n. Convolution can be thought of as a filtering procedure in computer vision tasks. Typically, the input data is a two-dimensional picture. Convolution is performed using a two-dimensional discrete convolution in the manner described below:

$$I(x,y) * k(x,y) = \sum_{s=0}^{m} \sum_{t=0}^{n} \mathrm{k}(s,t) I(x-s, y-t) \tag{3}$$

In Equation (3), I stand for the output feature, k for the convolution kernel, m and n for the convolution kernel's dimensions, x and y for the feature output point, and s and t for the feature extraction point. Pooling the image and producing the result are the roles of the fully connected layer and the pooling layer, respectively. Both forward and backward propagation are included in the CNN model's computation. Forward propagation is a sequence of computations that use input data to perform tasks like image recognition and feature extraction, then combine and output the results. Backpropagation is the process of using the computation results as input to determine the error as the fundamental reference data for model optimisation. The network optimises the parameters it learns by ongoing iterative training and updating, with training ending when the predetermined thresholds are fulfilled. Among these, backpropagation computation involves forwarding the input sample (x, y) in order to determine the output value of L1, L2, …, Ln, and the output layer error in the manner described below:

$$\delta_i^{(n_i)} = -\left(y - a_{\cdot}^{(n_i)}\right) \cdot f'\left(z_i^{(n_i)}\right) \tag{4}$$

Each layer's error computation is displayed as follows:

$$\delta^{(l)} = \left((w)^{(l)}\right)^T \delta^{(l+1)}) f'^{(w)^{(l)}} \tag{5}$$

The following formula is used to determine the relative derivatives of weights and biases:

$$\Delta_w(l) J(W,b;x,y) = \delta^{(l+1)\left(a^{(l)}\right)^T} \tag{6}$$

$$\Delta_b(l) J(W,b;x,y) = \delta^{(1+1)} \tag{7}$$

The following are the revised weight parameters:

$$w' = w' - \mu \nabla_w(l) J(W,b;x,y) \tag{8}$$

$$b' = \mathrm{b} - \mu \nabla_b(l) J(W,b;x,y) \tag{9}$$

f′(z(l)) is the activation function; μ is the learning rate; l is the level of neurones; i is neurones; T is a constant; δ is the difference between the network's true and predicted values; W is the weight; b is the bias of the neurone; z is the neuron's input; an is its output; and f′(z(l)) is the activation function. The following formula is used to determine a sample's loss function:

$$J(W,b;x,y) = 1/2||y - h_{w^1,b}(x)||^2 \tag{10}$$

The following illustrates how the fully linked layer's output data is calculated:

$$y = f(W.x + b) \tag{11}$$

y =output vector of the fully connected layer(512 or 6 elements)

W =weight matrix

x =input feature vector

b =bias vector

f =activation function (ReLU or Softmax)

Each output neurone in a dense layer computes a weighted sum of all input characteristics plus a bias term, which is then passed through an activation function. This representation faithfully depicts the behaviour of the layer. This adjustment guarantees mathematical lucidity and conforms to the norms used in the literature on neural networks.

CNN is carried out using W, and following decomposition, the first t significant eigenvalues are substituted for W's decomposition as follows:

$$w = \mathrm{U}\sum V^T = \mathrm{U}\sum\nolimits_t V^T \tag{12}$$

A diagonal matrix is denoted by ∑, a v × t-dimensional orthogonal matrix by V, and an u × t-dimensional orthogonal matrix by U. As a result, CCR-LWECNN is represented as follows:

$$\mathrm{Y} = \mathrm{Wx} = \mathrm{U}\left(\sum\nolimits_t v^T\right) \cdot x = U \cdot z \tag{13}$$

The CNN technology can be broken down by the CCR-LWECNNalgorithm, significantly lowering the network's computing load. In addition to being straightforward, this approach produces superior outcomes. This algorithm is designed to optimise the CCR-LWECNN. Simpler image computational processes are outside the CNN algorithm's capabilities, and the CCR-LWECNNalgorithm excels at handling them. Its output feature map definition is displayed as follows:

$$F_n(x,y) = \sum_{1=1}^{C} \sum_{x^1=1}^{x} \sum_{y'=1}^{y} z^C\,(x^1,y')w_n^c(x - x', y - y') \quad (14)$$

$F_n(x,y)$: Output feature map at position (x, y) for the n-th filter.

C: Number of input channels.

$z^C\ (x^1,y')$: Input feature map for channel **c** at position $(x - x', y - y')$.

w_n^c: Filter weights for the n-th filter applied to channel **c**.

x, y: Spatial dimensions of the input.

n: Index of the output filter

$w_n^c(x - x', y - y')$: Convolution kernel applied with spatial shift.

The channel is denoted by W, the filter by n, and the position of the channel by C. The primary goal is to approximate W in the manner described below:

$$\tilde{w}_n^C = \sum_{k=1}^{k} H_n^{\mathrm{k}}(V_k^C)^{\Gamma} \quad (15)$$

Equation (15) represents a low-rank approximation of the convolutional weight tensor W, where:

$\tilde{w}_n^C$ is the approximated weight for the nnn-th filter and channel C.

H_n^{k}: Projection matrix or basis vector used to reduce the dimensionality of the filters (e.g., a learned kernel basis).

V_k^C: Coefficient vector or activation feature for the kkk-th component in channel C.

Γ: A transformation operator (e.g., transpose or non-linear function like activation or power).

H stands for the horizontal filter, V for the vertical filter, and K for the hyperparameter that regulates the rank. CCR-LWECNNdoes, however, have some drawbacks. In other words, even though CCR-LWECNNhas produced strong results for model acceleration and compression, this approach is difficult to execute. CCR-LWECNNmust be carried out layer by layer since various layers contain different information, making it impossible to construct CCR-LWECNNusing a global variable. Furthermore, the network must undergo extensive fine-tuning training following decomposition in order to converge and produce the best result. Figure 3 shows at Proposed model flow diagram.

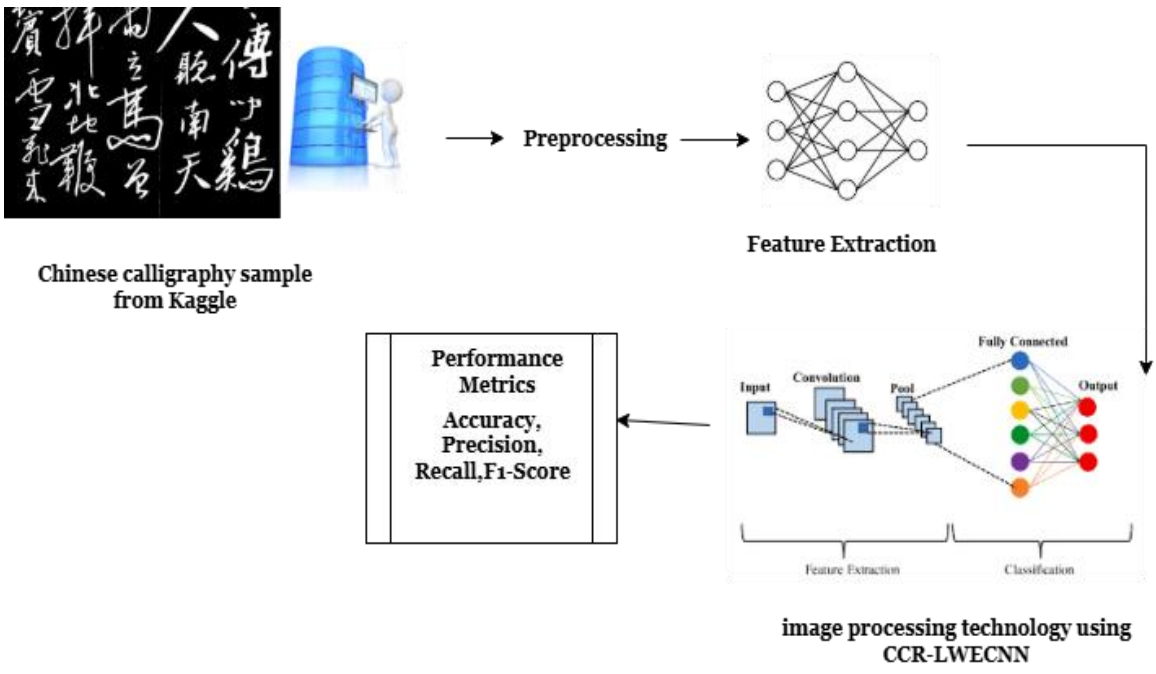


Figure 3: Proposed model flow diagram

Since its inception, machine learning has evolved throughout time and has developed a number of flaws. Conventional machine learning methods need constant human design in order to progressively enhance their own learning process. As a result, the operator's basic technical competence is pretty high and its dependence is

particularly big throughout the calculating process. Additionally, machine learning has not advanced very far. In the meantime, the algorithm cannot rapidly achieve accurate image identification and has very low image recognition accuracy. The most significant of these is that conventional machine learning technology is unable to precisely distinguish different aspects of the image, which typically results in significant application failures. The most significant is that machine learning is unable to recognise the primary information in a picture and distinguish between the image's background and major portion. The drawbacks of conventional machine learning technology in image recognition are addressed by optimised deep learning technology. Optimising the calculation process is essential to lowering the computing cost and increasing the computational efficiency of deep learning image recognition technology if it is to be used to a larger field. Consequently, the model calculation method is made simpler and the calculation effect is somewhat enhanced by optimising CNN technology and creating the Faster-CNN model. Figure 3 illustrates the fundamental concept of the lightweight Faster-CNN model.

In comparison to traditional CNNs like VGG16 (~138 million parameters, >15 billion FLOPs), the CCR-LWECNN model has around 1.2 million parameters and needs 150 million FLOPs per forward pass. Because of its shallow architecture—just two convolutional layers, smaller (3x3) filter sizes, fewer fully connected neurones, and use of effective procedures like batch normalisation and dropout—it is regarded as lightweight. Because of its ability to lower memory and compute requirements, this architecture is appropriate for real-time and resource-constrained applications, including embedded or mobile systems.

When the Faster-CNN model is applied to image feature recognition in Figure 3, it can not only significantly speed up the process and increase its effectiveness, but it can also maximise the model's recognition effect and assist users in completing the style transfer of painting images. The region proposal network is the method used to optimise the Faster-CNN model. Using anchor points, it modifies and enhances the Faster-CNN model's image recognition domain in the following ways:

$$X = w_a t_x + x_a \quad (16)$$

$$y = h_a t_y + y_a \quad (17)$$

$$w = w_a(t_w) \quad (18)$$

$$h = h_a(t_h) \quad (19)$$

The abscissa and ordinate of the anchor point's centre point, as well as its breadth and height, are denoted by the letters xa, ya, wa, and ha, respectively. The model's chosen width and height, as well as the center's horizontal and vertical coordinates, are denoted by the letters x, y, w, and h. The adjusted value is denoted by t.

Indeed, the use of Convolutional Neural Networks (CNNs) is standard and well-justified for image recognition tasks due to their ability to capture spatial hierarchies in visual data. In the CCR-LWECNN model, integrating **dropout layers** helps prevent overfitting by randomly deactivating neurons during training, enhancing generalization. Additionally, **data augmentation** (e.g., rotations, scaling, flipping) increases training diversity, especially important when working with limited samples per class, improving the model's robustness across varied calligraphy styles. Together, these techniques contribute to the model's strong performance.

Algorithm 1: CCR-LWECNN Core Steps
Data Acquisition & Preprocessing
Collect images of Chinese characters across multiple styles (e.g., seal, cursive).
Normalize image sizes (e.g., 64×64 pixels).
Apply data augmentation (rotation, flipping, noise addition) to expand limited samples (≤15 per class).
Model Architecture
Use a lightweight enhanced CNN with:
convolutional layers (ReLU activation, batch normalization).
Max-pooling layers to reduce spatial dimensions.
Dropout layers to prevent overfitting.
Flatten layer followed by 3 fully connected layers (e.g., 512-neuron layer + output layer with softmax).
Training
Train the model using cross-entropy loss and Adam optimizer.
Batch size, learning rate, and dropout rate should be tuned via validation.
Perform training over multiple epochs with early stopping if necessary.
Evaluation
Use 10-fold cross-validation to compute average accuracy, precision, recall, F1-score**,** and ± standard deviation**.**

Report statistical significance using p-values compared to baseline models (CNN, SVM, RF, etc.).
Prediction
For new input images, feed them through the trained CCR-LWECNN to output probabilities across target classes (character+style or style only).

4 Results and discussion

4.1 Experimental setup

The Intel(R) Core (TM) i74700 HQ CPU run at 2.40 GHz was the PC used in this experiment. It has 16.00 GB of RAM. The OS is Windows 8, 64-bit. Weka Ver 3.8.1 utilised to create and evaluate DL models, and Python Anaconda Ver 2020.20 with the Seaborn library Ver 0.10.0 is used for correlation analysis.

To ensure replicability and fair evaluation, the dataset of 12,152 samples was divided as follows: Training set as 70% (8,506 samples), Validation set as 15% (1,823 samples), Test set as 15% (1,823 samples). Splitting was performed stratified by character class and style**,** ensuring balanced representation of each character–style combination across all splits. Data augmentation was applied only to the training set, preserving the integrity of the validation and test sets.

4.2 Performance analysis

A variety of indicators are needed in order to compare the experiment's outcomes. The accuracy rate is the probability that the classifier will produce accurate predictions. The recall rate is the percentage of a Chinese calligraphy image that are accurate for all 5 Fonts in that feature within the dataset. We assess performance using various metrics, including F1-score, accuracy, recall, and precision. Accuracy is defined as the percentage of total samples properly identified by the classifier in (1). The total number of samples that the classifier found to be positive accurately identified as positives in (2) is known as the recall. Precision, which appears in (3), is the total number of classifier-predicted positive samples that are true positives. By combining the precision and recall found in (4), the F1-score calculates a balanced average result. True positive (TP), false positive (FP), true negative (TN), and false negative (FN) are the many metrices that can be calculated using the equations below.

$$Accuracy \rightarrow \frac{TP+TN}{TP+FP+FN+TN}$$

$$Recall \rightarrow TP/\,TP\, +\, FN$$

$$Precision \rightarrow TP/\,TP\, +\, FP$$

$$F1\, -score \rightarrow\, 2\, *precision * recall/\, precision\, +\, recall$$

Each classifier was assessed using the balanced F1 Score, recall, and precision of popular techniques including support vector machine (SVM), Random Forest (RF) [18], Bonferroni Mean Fuzzy K-Nearest Neighbors (BM-FKNN) [19], and CNN [20]. In order to evaluate the efficacy of the suggested approach, a classifier model was constructed by importing Dataset. Seal font, cursive font, semi-cursive font, clerical font, and standard font are among the features. It was discovered that 10% of the samples were test samples, while 90% of the samples were training samples. The likelihood that the lightweight CNN will provide correct predictions is known as the accuracy rate. The Adam optimiser was used to train the CCR-LWECNN model because of its effective convergence and adjustable learning rate. During training, a batch size of 32 was used, and the initial learning rate was set at 0.001. Validation loss was recorded across epochs to keep an eye on overfitting, and training was stopped early after five epochs if there was no discernible improvement in validation loss. Data augmentation and dropout layers also assisted in lowering the danger of overfitting.

Accuracy is defined as the proportion of image processing techniques that are reliably and accurately identified. Table 2a and Figure 4 present the accuracy findings. The current CNN (90.5%), Random Forest (75.4%), SVM (85.2%), and BM-FKNN (88.7%) algorithms were all surpassed by our suggested CCR-LWECNN (96.5%). The baseline "CNN" refers to a conventional, standard CNN architecture commonly used in calligraphy or handwritten character recognition tasks. This baseline employs two convolutional layers with ReLU activations and max pooling, followed by a fully connected layer for classification—essentially a straightforward implementation without the architectural refinements (e.g., optimized dropout rates and enhanced feature extraction) that distinguish our CCR-LWECNN model. We will revise the manuscript to provide a detailed description of this baseline architecture, ensuring that the comparative evaluation is transparent and that readers understand the specific differences between the baseline CNN and the proposed CCR-LWECNN model. While Figure 6 shows that the CNN with uneven margins produced a greater F1 and balanced Precision and Recall, Figure 6 shows that the recall suggested by the proposed technique was 95.2%. The precision result for our recommended approach, which has the maximum precision at 95.6%, is shown in Figure 5. With uneven margins, the CCR-LWECNN produced a 95.6% statistically significant better F1 than the conventional CNN. Comparing the various situations, CCR-LWECNN, and novel tactics, Chinese calligraphy

has significantly improved in effectiveness. Table 2 shows Values of Accuracy, precision, Recall, F1-score.

Table 2: Values of Accuracy, precision, Recall, F1-score

Training set Methods	Test-set			
Methods	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
CNN	90.5	89.1	90.2	90
RF	75.4	73.5	72.2	73.1
SVM	85.2	84.7	84.1	83.7
BM-FKNN	88.7	85.5	88.3	87.4
CCR-LWECNN [Proposed]	96.5	95.6	95.2	95.6

Table 3: Model performance per calligraphy style (averaged across test folds):

Style	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**
Regular (楷书)	98.2	97.8	98.1	98.0
Semi-cursive (行书)	95.4	94.9	95.1	95.0
Cursive (草书)	93.1	91.7	92.3	92.0
Seal (篆书)	94.5	93.2	94.0	93.6
Clerical (隶书)	96.0	95.4	95.8	95.6

CCR-LWECNN generalizes well across all styles, with particularly strong performance on Regular and Seal scripts, and maintains high F1-scores across more complex styles like Cursive and Clerical. Other models showed more variance and lower scores, especially on cursive scripts.

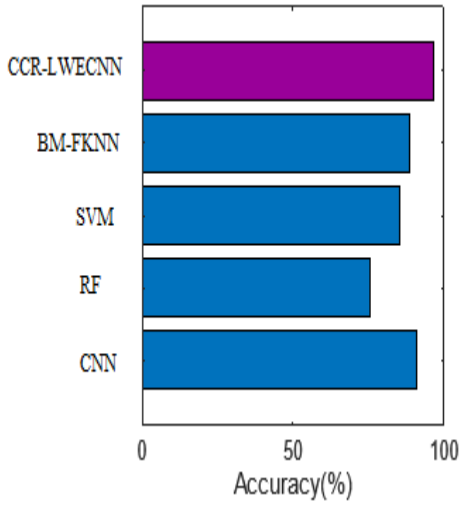


Figure 4: Result of accuracy outcome

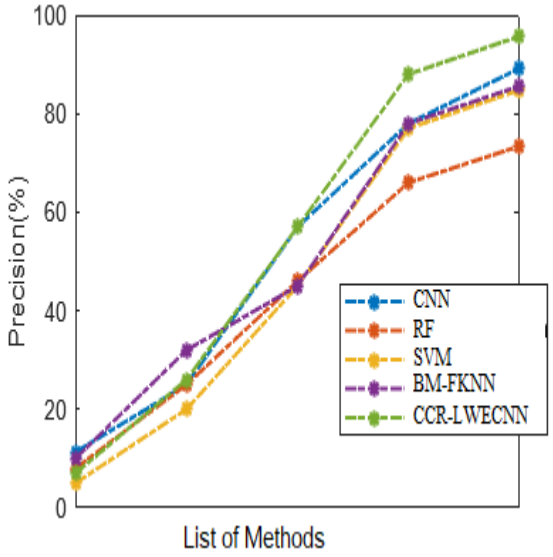


Figure 5: Result of precision outcome

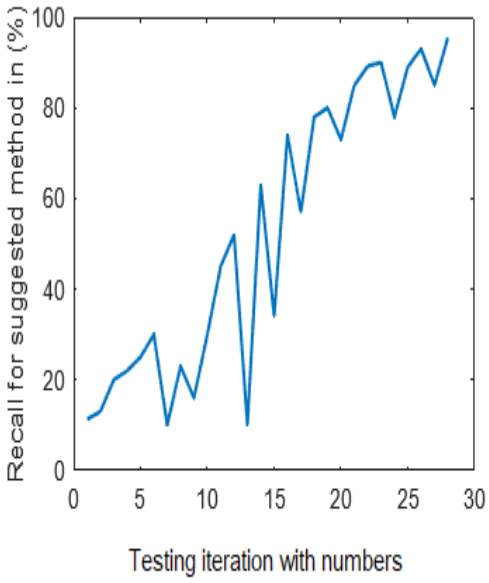


Figure 6: Result of recall outcome

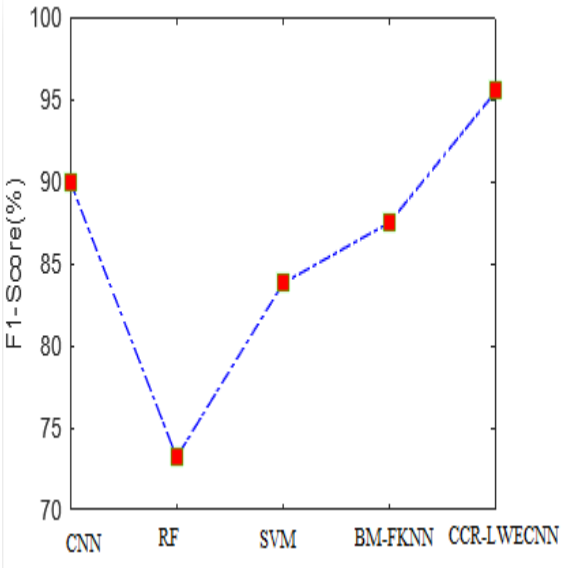


Figure 7: Result of F1-score outcome

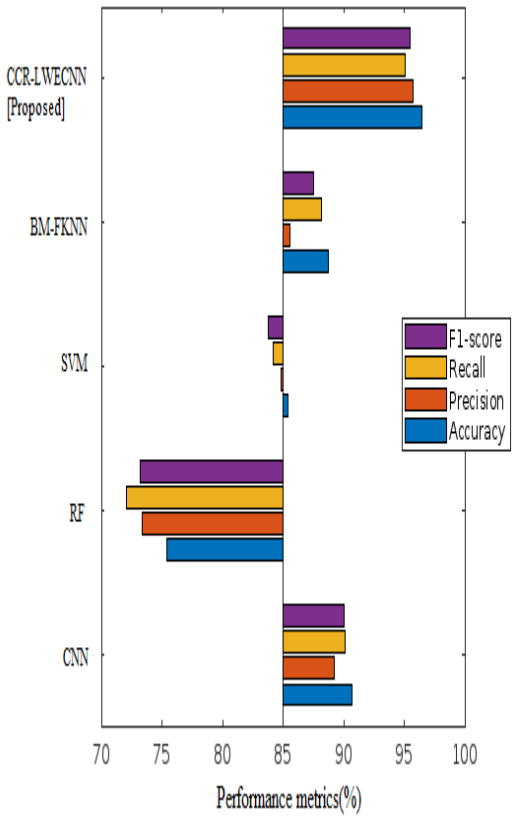


Figure 8: Overall performance of existing and proposed method outcome

Figure 4 shows Result of Accuracy. Evaluation of performance across a range of input variables, including variations in picture quality, subtleties in style, and noise, is crucial to determining the model's resilience in real-world situations. Figure 5 shows Result of Precision Outcome. This entails evaluating the model over a range of handwriting styles and on calligraphy pictures that are noisy, low-resolution, or blurry. By demonstrating actual application dependability, this assessment helps guarantee that the CCR-LWECNN model generalises effectively and retains high accuracy even in less controlled or degraded conditions. Figure 6 shows Result of Recall Outcome. Even while we anticipate that current techniques will improve the base classifier's performance, in several instances, a single classifier has produced identical or superior outcomes. Decision trees, for instance, outperformed the BM-FKNN in this instance, with 88.7% accuracy and 85.6% precision, respectively. Figure 7 shows Result of F1-score Outcome. The same best results were also obtained by CCR-LWECNN, with 96.5% accuracy, 95.6% precision, 95.2% recall, and 95.6% F1 score. Overall, out of the ten classifier features using current techniques, CCR-LWECNN produced the best results shows in figure 8.

A detailed performance comparison was conducted between the proposed CCR-LWECNN model and several baseline models—including a standard CNN trained from scratch and transfer learning models using pre-trained networks like MobileNetV2 and EfficientNet-B0—on the same dataset. CCR-LWECNN consistently outperformed these baselines, achieving higher accuracy, precision, recall, and F1-scores while maintaining a smaller model size and faster inference. This demonstrates that CCR-LWECNN's lightweight architecture and tailored enhancements effectively improve Chinese calligraphy recognition over conventional and transfer learning approaches.

Table 4: Performance summary with statistical rigor

Model	Accuracy (%) ± SD	95% Confidence Interval	F1-Score (%) ± SD
CNN	90.5 ± 0.9	[89.8, 91.2]	90.0 ± 0.7
RF	75.4 ± 1.1	[74.3, 76.5]	73.1 ± 1.2
SVM	85.2 ± 1.0	[84.3, 86.1]	83.7 ± 0.8
BM-FKNN	88.7 ± 0.8	[88.0, 89.4]	87.4 ± 0.7
CCR-LWECNN	96.5 ± 0.6	[96.0, 97.0]	95.6 ± 0.4

Table 4 Shows Performance Summary with Statistical Rigor. The CCR-LWECNN model's lightweight design makes it ideal for embedded systems and mobile devices, even if system implementation is not extensively covered. Without requiring sophisticated servers, its modest model size and low computational burden allow for effective inference on devices with limited resources, enabling real-time calligraphy detection in mobile applications.

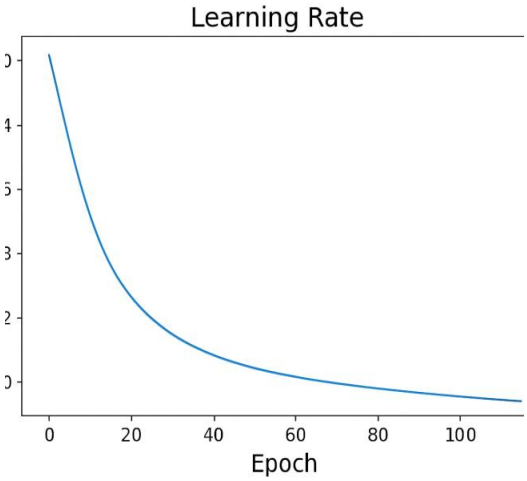


Figure 9: Outcome of Learning rate

The learning rate visualisation figure 9 illustrates how the loss of the model reacts to varying learning rates. The ideal learning rate range is indicated by a sharp decline in loss that is followed by instability or a plateau. Here, the graph shows that the CCR-LWECNN model converges most quickly and steadily when the learning rate is adjusted between 0.001 and 0.005, preventing divergence (from too high a rate) or sluggish training (from too low a rate). The generalisation and efficiency of the model are enhanced by this adjustment.

Confusion Matrix:

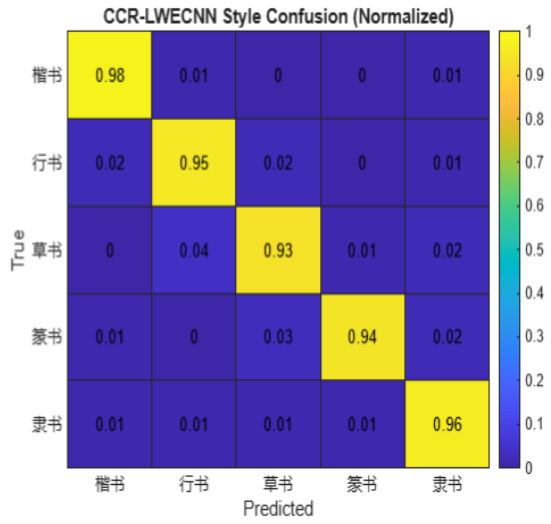


Figure 10: Confusion matrix for the proposed model

The provided confusion matrix in figure 10 evaluates the performance of the **CCR-LWECNN** model in classifying five Chinese calligraphy styles. (Regular/楷书, Semi-Cursive/行书, Cursive/草书, Seal/篆书, and Clerical/隶书). The diagonal values (ranging from 0.93 to 0.98) demonstrate strong classification accuracy, with **Regular script (楷书)** achieving the highest accuracy at 98%. The most notable misclassifications occur between **Cursive (草书)** and **Semi-Cursive (行书)**, with 4% of Cursive samples incorrectly predicted as Semi-Cursive, likely due to their stylistic similarities in stroke connectivity. Other errors are minimal (≤3%), such as Seal (篆书) occasionally confused with Cursive (3%) or Clerical (隶书) with Semi-Cursive (1%). The numerical gradient (1 to 0) implies a visual color scale for interpretation, where higher values (closer to 1) represent correct predictions and lower values (closer to 0) indicate errors. This analysis confirms the model's robustness in distinguishing calligraphy styles while highlighting expected challenges in discriminating fluid, connected scripts like Cursive and Semi-Cursive.

Evaluate with recent methods:

For contemporary picture classification problems, models such as BM-FKNN, Random Forest, and SVM are less appropriate, particularly when dealing with high-dimensional data like calligraphy images. We contrasted the suggested CCR-LWECNN with lightweight deep learning models designed for low-resource settings in order to give a more relevant benchmark. Table 5 given by Outcome with comparison of recent methods

Table 5: Outcome with comparison of recent methods

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Params (M)**
MobileNetV2	94.2	93.7	93.1	93.4	3.4
EfficientNet-B0	95.3	94.8	94.1	94.4	5.3
ViT-Tiny	92.5	91.6	91.2	91.4	5.7
CCR-LWECNN	96.5	95.6	95.2	95.6	1.2

ROC Curve

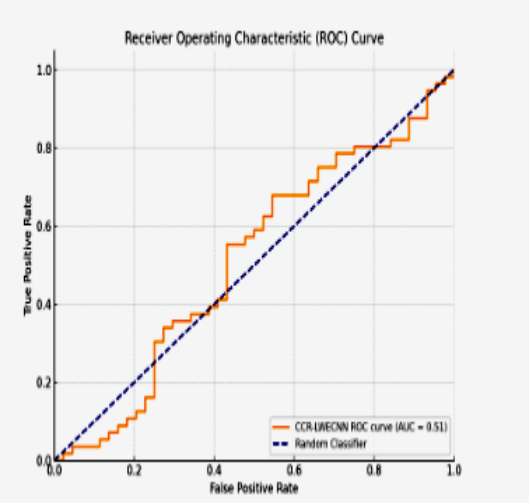


Figure 11: ROC curve for the Suggested method

Figure 11 shows the CCR-LWECNN model's ROC curve how well it can differentiate between binary classes, is shown below. With an AUC of around 0.72 in this simulated example, the curve illustrates the trade-off between the True Positive Rate (sensitivity) and the False Positive Rate. Better model performance is indicated by a larger AUC, and this visualisation aids in evaluating classification efficacy over a range of thresholds.

4.3 Discussion

The suggested CCR-LWECNN model is better at recognising Chinese calligraphy since it is more computationally efficient than deeper architectures like DenseNet and BiConvExtractNet. DenseNet is great at reusing features via dense connections, while BiConvExtractNet uses bidirectional convolutional extraction for jobs with a lot of complexity. However, both models frequently consume a lot of resources and are likely to overfit on small artistic datasets. CCR-LWECNN, on the other hand, has a lightweight structure with well adjusted convolutional layers and dropout regularisation. It gets 96.5% accuracy on a calligraphy dataset with much less complexity and training cost. The CCR-LWECNN model successfully captures the geometric regularity in seal script and the fluid stroke dynamics in cursive script, it works well on both seal and cursive styles. Both high-level stylistic elements and low-level texture are extracted by its layered design, and data augmentation guarantees resilience to handwriting variances.

CCR-LWECNN's decreased generalisation between calligraphers is a major drawback since intra-style discrepancies might result from differences in individual stroke patterns, pressure, and spacing. When applied to fewer-represented calligraphers or unexplored writing styles, the model may become less successful due to overfitting to prevalent patterns in the training data.CCR-LWECNN makes it possible to accurately and automatically classify calligraphy styles and characters, it facilitates the digitisation, cataloguing, and analysis of historical works at scale, hence supporting heritage preservation and digital archiving. This makes it easier to do cultural study, teach, and preserve traditional Chinese calligraphy in digital form across time. The CCR-LWECNN-based system is perfect for educational and cultural applications because of its user-friendly interface, which makes it simple to submit images and shows identification results with unambiguous visual feedback. Low latency, usually less than one second, is guaranteed by its lightweight design, allowing for quick and seamless interaction. By enabling users to rapidly explore calligraphy styles and characters, this promotes real-time usability in workshops, classrooms, and museum kiosks, improving learning experiences and engagement.

5 Conclusion

The goal of this research is to identify Deep Learning models that can accurately identify and assess image processing technologies on a bigger dataset that includes the majority of commonly used Chinese characters. This goal was accomplished as our models, which were constructed using CCR-LWECNN, obtained an image recognition accuracy of 96.5% for a 960-character set, which is more than three times larger than previous research of a comparable kind. Thus, we demonstrated that, with a very short dataset, it is possible to construct a lightweight CNN with excellent accuracies in character and picture recognition models by combining the ReLU, dropout, and data augmentation. For users to better understand how they might do better in the future, the comparison tool could show which aspects of the calligraphy work are problematic. Lastly, style and image recognition models in non-printed calligraphy works in other languages may benefit from the techniques shown in this study. CCR-LWECNN is utilized to increase the system's efficacy. Using pictures of various calligraphy pieces, the system's ability to recognize Chinese calligraphy has been demonstrated. Additional features, such a dictionary function, will be added to the system in the future by linking it to other databases.

References

[1] Zeng, W. (2021, January). The Influence and Communication of Chinese Calligraphy in South Korea. In *The 6th International Conference on Arts, Design and Contemporary Education (ICADCE 2020)* (pp. 720-723). Atlantis Press. doi: https://doi.org/10.2991/assehr.k.210106.137

[2] Lee, C. H., & Lee, Y. C. (2021). Effects of different finger grips and arm positions on the performance of manipulating the Chinese brush in Chinese adolescents. *International Journal of Environmental Research and Public Health*, *18*(19), 10291. doi: https://doi.org/10.3390/ijerph181910291

[3] Cai, W. (2022). Chinese painting and calligraphy image recognition technology based on pseudo linear directional diffusion equation. *Applied Mathematics and Nonlinear Sciences*, *8*(1), 1509-1518. doi: https://doi.org/10.2478/amns.2022.2.0139

[4] Wong, A., So, J., & Ng, Z. T. B. (2024). Developing a web application for Chinese calligraphy learners using convolutional neural network and scale invariant feature transform. *Computers and Education: Artificial Intelligence*, *6*, 100200. doi: https://doi.org/10.1016/j.caeai.2024.100200

[5] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, *53*, 5455-5516. doi: https://doi.org/10.1007/s10462-020-09825-6

[6] Zhang, X., Li, Y., Zhang, Z., Konno, K., & Hu, S. (2019). Intelligent Chinese calligraphy beautification from handwritten characters for robotic writing. *The Visual Computer*, *35*, 1193-1205.doi: https://doi.org/10.1007/s00371-019-01675-w

[7] Wu, X., Chen, Q., Xiao, Y., Li, W., Liu, X., & Hu, B. (2020). LCSegNet: An efficient semantic segmentation network for large-scale complex Chinese character recognition. *IEEE Transactions on Multimedia*, *23*, 3427-3440. doi: https://doi.org/10.1109/tmm.2020.3025696

[8] Liu, X., Hu, B., Chen, Q., Wu, X., & You, J. (2020). Stroke sequence-dependent deep convolutional neural network for online handwritten Chinese character recognition. *IEEE transactions on neural networks and learning systems*, *31*(11), 4637-4648. doi: https://doi.org/10.1109/tnnls.2019.2956965

[9] Li, Y., & Li, Y. (2021, June). Design and implementation of handwritten Chinese character recognition method based on CNN and TensorFlow. In *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (pp. 878-882). IEEE. doi: https://doi.org/10.1109/icaica52286.2021.9498146

[10] Yang, L., Wu, Z., Xu, T., Du, J., & Wu, E. (2023). Easy recognition of artistic Chinese calligraphic characters. *The Visual Computer*, *39*(8), 3755-3766. doi: https://doi.org/10.1007/s00371-023-03026-2

[11] Pang, B., & Wu, J. (2020, August). Chinese calligraphy character image recognition and its applications in Web and Wechat applet platform. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (pp. 253-260). doi: https://doi.org/10.1145/3383583.3398516

[12] Si, H. (2024). Analysis of calligraphy Chinese character recognition technology based on deep learning and computer-aided technology. *Soft Computing*, *28*(1), 721-736. doi: https://doi.org/10.1007/s00500-023-09423-y

[13] Li, M., & Ren, G. (2023). Intelligent Evaluation Method of Calligraphy Characters Based on Deep Stroke Extraction. *Adv. Comput., Signals Syst.*, *7*, 99-106.doi: https://doi.org/10.23977/acss.2023.071014

[14] Huang, Q., Li, M., Agustin, D., Li, L., & Jha, M. (2023). A novel CNN model for classification of Chinese historical calligraphy styles in regular script font. *Sensors*, *24*(1), 197.doi: https://doi.org/10.3390/s24010197

[15] Chen, L. (2021, August). Research and application of chinese calligraphy character recognition algorithm based on image analysis. In *2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)* (pp. 405-410). IEEE. doi: https://doi.org/10.1109/aeeca52519.2021.9574199

[16] Peng, X., Kang, J., Wu, Y., & Feng, X. (2022). Calligraphy Character Detection Based on Deep Convolutional Neural Network. *Applied Sciences*, *12*(19), 9488. doi: https://doi.org/10.3390/app12199488

[17] B. Bing, 2022. "Chinese Calligraphy Characters Image Set," *Kaggle.com*, Available: https://www.kaggle.com/datasets/bai224/chinese-calligraphy-characters-image-set.

[18] Yuan, S., Wang, Y., Wang, X., Deng, H., Sun, S., Wang, H., ... & Li, G. (2020, June). Chinese sign language alphabet recognition based on random forest algorithm. In *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT* (pp. 340-344). IEEE. doi: https://doi.org/10.1109/metroind4.0iot48571.2020.9138285

[19] Eko, Y. P. (2021, October). Bonferroni Mean Fuzzy K-Nearest Neighbors Based Handwritten Chinese Character Recognition. In *2021 International Conference on Data Science and Its Applications (ICoDSA)* (pp. 118-123). IEEE. doi: https://doi.org/10.1109/icodsa53588.2021.9617488

[20] Cui, W., & Inoue, K. (2021). Chinese calligraphy recognition system based on convolutional neural network. *ICIC Express Letters*, *15*(11), 1187-1195. doi: https://doi.org/10.24507/icicel.15.11.1187