Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.37598

Deep Learning-Based Calligraphic Aesthetic Assessment Guided by Gestalt Psychological Features

Hongtao Yan

Shanghai World Foreign Language Academy, Shanghai, China
yanjason952@gmail.com

Abstract. Calligraphic aesthetic assessment requires the information about local stroke appearance and the spatial organization of the complete character. This paper develops a supervised regression method that combines twelve Gestalt-inspired geometric descriptors with a ResNet50 image encoder. The descriptors quantify the visual balance, ink distribution, reflected structure, projection rhythm, and distance-transform smoothness. Two feature-wise linear modulation (FiLM) modules convert these measurements into channel-wise scales and shifts for intermediate convolutional features. A direct-concatenation control supplies the same descriptors to the regression head. This design allows explicit geometric information to interact with the learned visual representations during score prediction. In the ResNet50 experiment, intermediate FiLM achieves PCC 0.7067 and MAE 1.3554, compared with 0.6958 and 1.3793 for image-only regression. In whole-work assessment, combining local window-score statistics with global layout descriptors achieves PCC 0.6323. These results support descriptor-guided regression at the character level and the combination of local appearance with global composition for whole-work assessment.

Keywords: Chinese calligraphy, aesthetic assessment, Gestalt descriptors, ResNet50, FiLM

1. Introduction

Calligraphic aesthetic assessment concerns the shape of strokes and the relations among them. Centroid placement, open space, opposing components, and directional rhythm contribute to the appearance of a complete character. A computational model can connect these visible properties to annotated scores and provide repeatable feedback for calligraphy analysis and learning.

Existing methods use centers, projections, skeletons, convolutional features, and component graphs to assess handwritten forms [1-4]. Geometric descriptors expose structural information in a compact form, while convolutional encoders learn local and hierarchical image representations. Combining the two is particularly relevant when the available aesthetic annotations are limited.

Gestalt psychology provides a basis for describing balance, figure–ground organization, symmetry, and continuity [5, 6]. Computational aesthetics has related measurable aspects of these concepts to human ratings in several visual tasks [7-9]. This paper translates these ideas into twelve descriptors and uses them as conditioning inputs to a convolutional network. The resulting model learns how the descriptors relate to ratings for the characters and scripts represented in the data.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.37598

The proposed system extracts geometric measurements and image features in parallel. FiLM uses the standardized measurements to modulate intermediate ResNet50 channels before score regression [10]. This architecture supplies explicit information about the complete glyph while retaining the learned representation of strokes and texture. The same local-to-global idea can be extended to whole works by aggregating window-level scores and adding composition descriptors.

1. This paper defines twelve computable descriptors of calligraphic geometry and integrates them with CNN features for supervised aesthetic-score regression.
2. This paper designs a two-stage FiLM conditioning pathway in which the descriptors generate channel-wise scales and shifts, together with direct-concatenation and grouped-generator controls for evaluating the fusion design.
3. This paper evaluates the approach through multiscript quantitative comparisons and Grad-CAM visualizations. Intermediate FiLM improves on the ResNet50 image-only baseline across the four reported metrics. The whole-work experiment also shows gains from combining local window-score statistics with global layout descriptors.

2. Related work

This paper reviews computational calligraphy assessment and Gestalt-based visual description to motivate using explicit geometric features as CNN conditioning inputs.

2.1. Computational assessment of Chinese calligraphy

Early computer-assisted calligraphy assessment often compared a learner's writing with a reference character. Han, Chou, and Wu segmented characters, extracted center, size, and projection features, and combined them through fuzzy inference to generate scores and corrective feedback [1]. The visible criteria were useful for instructional feedback, but the target was reference-based grading rather than continuous aesthetic regression across script categories. Sun combined skeleton and corner similarity with Mi-grid partitions, Hu moments, and Pearson correlation to describe local strokes and global structure [2]. Its feedback likewise depended mainly on similarity to a standard form.

Later studies introduced learned representations. Hong investigated glyph shape and full-angle projections using VGG16 feature maps. The final scoring system combined physical-center, convex-hull-center, and convolutional feature vectors, comparing them with scored samples of the same character through cosine similarity and converting the similarities into score estimates [3]. Prediction therefore remained tied to same-character comparisons. Qi represented radicals and their spatial relations as a graph and combined graph convolution with matrix completion to reconstruct standard glyph features for layout evaluation [4]. The graph made component structure explicit and required a radical-based representation.

These studies establish the utility of centers, projections, skeletons, and component relations for glyph assessment. Their emphasis on reference matching or predefined structure leaves room to study geometric descriptors as conditioning signals within an aesthetic-regression network. This paper therefore supplies a fixed twelve-dimensional Gestalt vector to intermediate CNN channels and compare this placement with fusion at the regression head.

DOI: 10.54254/2755-2721/2026.37598

2.2. Gestalt theory and computational visual aesthetics

Gestalt theory describes how visual elements form perceptual wholes. Arnheim connected balance and visual weight to artistic composition [5], while Palmer reviewed grouping and perceptual organization [6]. These accounts motivate measurements of ink placement, regional correspondence, and continuity in calligraphic characters.

Bauerly and Liu quantified symmetry, balance, and compositional elements in geometric and interface layouts [7]. Lai et al. extended visual-weight models to text-overlaid images and found an association between averaged balance and aesthetic ratings, whereas symmetry did not show a proportional relationship with aesthetic ratings in their experiments [8]. Their findings motivate measuring several complementary cues and learning their task-specific relation to scores.

Brachmann and Redies reviewed handcrafted, statistical, and learned representations in computational aesthetics [9]. Following this combination of interpretable measurements and learned representations, the model uses geometric descriptors to condition the CNN feature hierarchy. The comparison between intermediate FiLM and late fusion tests where these descriptors contribute to prediction.

3. Methodology

The system combines a visual representation of strokes with explicit measurements of the spatial organization of a character. As shown in Figure 1, it contains an image encoder, a Gestalt descriptor branch, and a score regressor. Preprocessing supplies the grayscale image and its ink mask to the two branches. The descriptor vector generates FiLM parameters that adapt intermediate image features, and pooled features are mapped to an aesthetic score. This paper first describes data construction and then define each computational component.

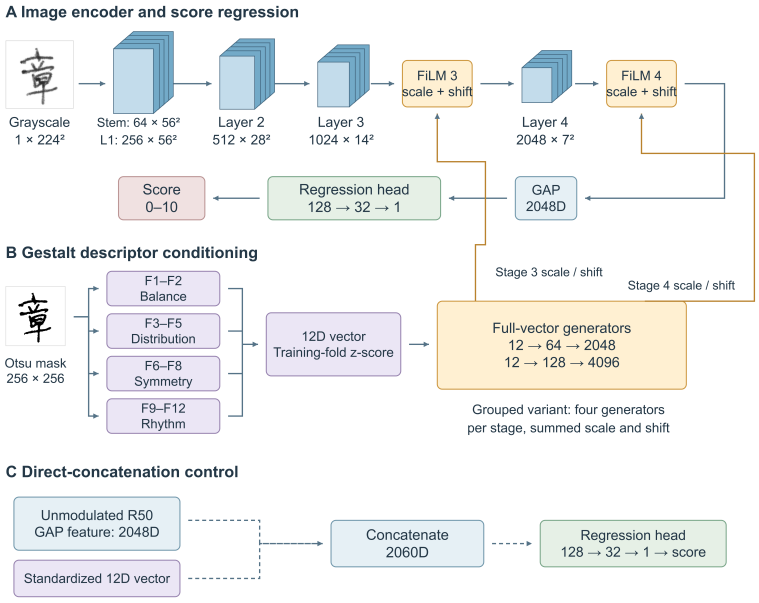

Figure 1. Gestalt-guided ResNet50 architecture and direct-concatenation control. Feature-map labels give channels × spatial dimensions (picture credit: original)

DOI: 10.54254/2755-2721/2026.37598

3.1. Dataset collection and preparation

The dataset contains 897 labeled character images from three script groups: 272 Kaishu, 415 Lishu, and 210 Xingshu. The Kaishu and Lishu images are retained from the multiscript collection associated with CASIA-HWDB [11]. The Xingshu images were selected and annotated from Data Set #2 in a separate scoring session. Images are stored as 256 × 256 grayscale glyphs, and the annotation record contains the score, character identity, script label, and source path. The 58 separately selected full-score reference characters are kept outside this evaluation collection.

Preparation checks image readability, matches every score to its image, and records a pixel hash for each target. All 897 target hashes are unique. For descriptor extraction, Otsu's threshold selection method [12], applied with inverse binarization, converts dark ink to a foreground value of one and the white background to zero. For the image encoder, the grayscale image is resized to 224 × 224 and normalized with mean and standard deviation 0.5. Empty foreground masks are rejected. A fixed descriptor order and saved image identifiers preserve correspondence throughout training and evaluation.

The evaluation uses three sample-level folds, stratified jointly by script and score band, with split seed 42. Training and test image indices and hashes are disjoint in every fold. Different images of the same character can occur across folds. Writer identifiers are unavailable, and the later Xingshu annotations retain their own acquisition-session identity. Each training seed uses the same folds and preprocessing.

3.2. Input representation and network overview

Let X be the preprocessed grayscale image and v the twelve-dimensional descriptor extracted from its original-resolution ink mask. The model predicts a scalar aesthetic score from these two inputs:

$$y_{pred} = f_\theta(X, v), \quad v \in \mathbb{R}^{12}, \quad y \in [0, 10]. \tag{1}$$

Here y_{pred} is the predicted score, y is the human-assigned score, f_θ is the regression network, and θ denotes all learned weights and biases. X is the one-channel image tensor, v is a column vector of twelve real-valued descriptors, $\mathbb{R}$ denotes the real numbers, and [0, 10] is the output score interval.

The ResNet50 encoder [13] produces feature maps with 64 channels after the stem and 256, 512, 1024, and 2048 channels after its four residual stages. Their spatial sizes are 56 × 56, 56 × 56, 28 × 28, 14 × 14, and 7 × 7. FiLM operates after stages 3 and 4. Global average pooling then produces a 2048-dimensional vector for a regression head with 128 and 32 hidden units. ReLU follows each hidden linear layer, dropout with probability 0.3 follows the first ReLU, and a sigmoid scaled by 10 produces the final score.

The pretrained first convolution is converted to one input channel by summing its three input-channel kernels. Linearity makes this operation preserve the convolution response to the same normalized grayscale image replicated across three input channels:

$$K_{gray} = \sum_{c=1}^{3} K_c, \quad \sum_{c=1}^{3} K_c * X = \left(\sum_{c=1}^{3} K_c\right) * X = K_{gray} * X. \tag{2}$$

K_c denotes the pretrained kernel for RGB input channel c, with c = 1, 2, 3. K_{gray} is the adapted grayscale kernel, X is the same normalized grayscale image in each replicated channel, * denotes convolution, and Σ denotes summation over the three input channels. The number of output channels remains unchanged.

DOI: 10.54254/2755-2721/2026.37598

3.3. Gestalt feature extraction

The twelve descriptors summarize visual balance, the allocation of ink and blank space, correspondence between opposing regions, and directional rhythm. These properties give the model explicit information about the organization of a complete character. Their relation to aesthetic quality is learned from ratings, allowing the same geometric value to have different implications across glyph structures and scripts.

Let I(x, y) be the binary ink mask at pixel coordinates (x, y), with the origin at the upper-left corner. Coordinates increase rightward and downward. W and H are the canvas width and height, respectively. Ink area and raw image moments are defined by:

$$A = \sum_{x,y} I(x,y), \quad m_{pq} = \sum_{x,y} x^p y^q \, I(x,y). \tag{3}$$

A is the number of foreground pixels. m_{pq} is the raw moment of nonnegative integer orders p and q; the horizontal and vertical coordinates raised to these powers weight the binary mask. The sum extends over the complete W × H canvas. The zero-order moment m_{00} equals A.

$$x_c = \frac{m_{10}}{m_{00}}, \quad y_c = \frac{m_{01}}{m_{00}}, \quad x_0 = W/2, \quad y_0 = H/2. \tag{4}$$

x_c and y_c are the horizontal and vertical ink-centroid coordinates. m_{10} and m_{01} are the first-order horizontal and vertical moments; m_{00} is the ink area. x_0 and y_0 are the implementation-defined canvas-center coordinates, and W and H are measured in pixels.

Normalized centroid displacement. The position of the ink centroid describes the overall balance of the written form. Measuring its displacement from the canvas center quantifies how far the character is offset while accommodating deliberately asymmetric structures.

$$F_1 = \frac{\left(\sqrt{(x_c - x_0)^2 + (y_c - y_0)^2}\right)}{\frac{\sqrt{(W^2 + H^2)}}{2}} \tag{5}$$

F_1 is the normalized displacement. x_c and y_c are the ink-centroid coordinates, x_0 and y_0 are the canvas-center coordinates, and W and H are the canvas dimensions. The denominator is half the canvas diagonal.

Centroid displacement direction. The direction of a centroid offset describes whether the character leans rightward, leftward, upward, or downward. Together with the displacement magnitude, it represents the posture of the complete glyph.

$$F_2 = \frac{atan2(y_c - y_0, x_c - x_0)}{\pi} \tag{6}$$

F_2 is the displacement angle divided by π. atan2 takes the vertical offset $y_c - y_0$ first and the horizontal offset $x_c - x_0$ second, retaining the quadrant. π is the circle constant. Coordinates increase downward, so 0.5 points down and −0.5 points up. The value lies in [−1, 1]; the implementation returns zero for zero displacement.

Convex hull fill rate. The proportion of ink inside the convex envelope captures the openness of a character. It distinguishes densely filled forms from forms with substantial internal blank space, supporting analysis of ink distribution and enclosure.

DOI: 10.54254/2755-2721/2026.37598

$$F_3 = \frac{A}{A_h} \tag{7}$$

F_3 is the fill rate. A is the ink-pixel area and A_h is the polygon area of the convex hull enclosing the external ink contours. A degenerate zero-area hull is assigned a fill rate of zero.

Contour compactness. Compactness relates the amount of ink to the length of its boundary. It provides a geometric measure of how concentrated or boundary-rich the written form is, complementing the convex-hull description.

$$F_4 = \frac{4\pi A}{P^2} \tag{8}$$

F_4 is compactness, A is ink area, P is the total closed arc length of external contours, and π is the circle constant. Interior-hole boundaries are excluded. Zero perimeter is handled by returning zero.

Spatial distribution entropy. The spatial distribution of ink describes how the character uses the surrounding blank space. Entropy across a regular grid measures whether ink is concentrated in a few regions or distributed more evenly across the canvas.

$$F_5 = -\frac{1}{ln(16)} \sum_{i:p_i>0} p_i ln(p_i) \tag{9}$$

F_5 is normalized entropy. The canvas is divided into 16 equal cells; p_i is the ink mass in cell i divided by total ink area A. The sum includes cells with p_i > 0. ln is the natural logarithm, and ln(16) is the entropy of equal mass in all cells.

Left right reflection similarity. Correspondence between left and right components contributes to perceived structural balance. Reflection similarity quantifies this relation while allowing the regression model to learn its relevance for each type of glyph.

$$F_6 \;=\; SSIM(J,\; R_h(J)). \tag{10}$$

F_6 is left–right similarity. $J = 255I$ is the foreground mask on an 8-bit intensity scale, R_h reverses its column order, and SSIM is the structural similarity index [14], applied here to the mask and its reflection and averaged as defined below.

Top bottom reflection similarity. The relation between upper and lower components describes another aspect of structural organization. This measurement complements left–right similarity by capturing correspondence along the vertical direction.

$$F_7 \;=\; SSIM(J,\; R_v(J)). \tag{11}$$

F_7 is top–bottom similarity. J is the 8-bit foreground image, R_v reverses its row order, and SSIM is the same local similarity operator used for F_6.

Quadrant balance. The distribution of ink among four canvas regions describes regional visual weight. A balanced allocation can support stability, while differences between quadrants represent the distinctive organization of individual characters.

$$F_8 = max(0,1 - \frac{Var(q_1,q_2,q_3,q_4)}{(Mean(q_1,q_2,q_3,q_4))^2}) \tag{12}$$

F_8 is quadrant balance. q_1, q_2, q_3, and q_4 count ink in the upper-left, upper-right, lower-left, and lower-right quadrants. Mean is their arithmetic average and Var is their population variance with

DOI: 10.54254/2755-2721/2026.37598

divisor 4. The square is applied after averaging. max clips negative values to zero.

Column projection tension. Variation in the column projection summarizes the alternation between dense vertical bands and open gaps. This descriptor captures horizontal changes in stroke distribution and provides a measure of directional rhythm.

$$F_9 = ln(1 + \frac{Var(p_x)}{(Mean(p_x))^2}) \tag{13}$$

F_9 is the logarithmically compressed relative variance of the column projection p_x. Mean and Var are computed across all W columns using population variance. ln is the natural logarithm. The projection itself is defined by the following sum.

$$p_x(x) \;=\; \sum_y I(x,y). \tag{14}$$

p_x (x) is the ink count in column x. The sum runs over row coordinate y, and I(x, y) is the binary foreground mask.

Row projection tension. Variation in the row projection describes upper–lower banding and the alternation of ink-rich and open horizontal regions. It complements the column descriptor in representing the two-dimensional rhythm of a glyph.

$$F_{10} = ln(1 + \frac{Var(p_y)}{(Mean(p_y))^2}) \tag{15}$$

F_{10} is the logarithmically compressed relative variance of the row projection p_y. Mean and Var are computed across all H rows using population variance, and ln is the natural logarithm. The row projection is defined below.

$$p_y(y) \;=\; \sum_x I(x,y). \tag{16}$$

p_y (y) is the ink count in row y. The sum runs over column coordinate x, and I(x, y) is the binary foreground mask.

Scaled projection peak count. Prominent peaks in the two projections summarize repeated concentrations of ink. Their combined count describes the frequency of dense bands and complements the magnitude-based rhythm descriptors.

$$F_{11} = \frac{N_{(peak)}(p_x)+N_{(peak)}(p_y)}{20} \tag{17}$$

F_{11} is the scaled peak count. N_{peak} counts retained maxima in each projection after a five-point moving average. A peak is retained when its height above the larger of the left-side and right-side minima reaches 8% of the smoothed projection maximum. p_x and p_y are the column and row projections. The constant 20 scales the combined count.

Distance transform smoothness. A distance transform measures the interior width of ink regions. Variation in its local second differences provides a compact description of stroke-form regularity, linking the geometric branch to the smoothness and continuity of the written form.

$$F_{12} \;=\; \frac{1}{1 + Std_{I(x,y)=1}(|\Delta D(x,y)|)} \tag{18}$$

DOI: 10.54254/2755-2721/2026.37598

F_{12} is the smoothness descriptor. D(x, y) is the OpenCV L2 distance transform with mask size 5, and ΔD is its discrete Laplacian with kernel size 1. Absolute values are taken before Std, the population standard deviation over foreground pixels $I(x, y) = 1$. The reciprocal maps lower variability to a larger descriptor value.

The reflection descriptors use the spatial mean of local structural similarity following Wang et al. [14]. Local statistics are computed with an 11 × 11 Gaussian window of standard deviation 1.5 pixels, using OpenCV default border handling:

$$SSIM(U,V) = Mean_{x,y}\left(\frac{(2\mu_U\mu_V + C_1)(2\sigma_{UV} + C_2)}{(\mu_U{}^2 + \mu_V{}^2 + C_1)(\sigma_U{}^2 + \sigma_V{}^2 + C_2)}\right). \tag{19}$$

U and V denote the two 8-bit images being compared. μ_U and μ_V are their local means, σ_U^2 and σ_V^2 their local variances, and σ_{UV} their local covariance. C_1 and C_2 stabilize the denominator. $Mean_{x,y}$ averages the local similarity values over image positions x and y.

$$C_1 = (0.01 \times 255)^2, \quad C_2 = (0.03 \times 255)^2, \quad v = [F_1, \ldots, F_{12}]^T. \tag{20}$$

C_1 and C_2 use intensity range 255 and factors 0.01 and 0.03, respectively. v is the complete descriptor column vector in the numerical order F_1 through F_{12}, and superscript T denotes transpose. The ellipsis represents the intermediate descriptor entries.

Each descriptor dimension is standardized using its mean and population standard deviation fitted on the current training subset. Subtracting the mean and dividing by the fitted standard deviation produces the conditioning vector z. A standard deviation below 0.000001 is replaced by one. The same fitted transformation is applied to validation and test images.

3.4. Film conditioning and fusion controls

Following the FiLM formulation [10], each selected residual stage uses a two-layer multilayer perceptron to map our standardized descriptor to a scale increment and an additive shift. The increment is added to one to form the channel scale:

$$[\Delta\gamma^l;\ \beta^l] = g_l(z), \quad \gamma^l = 1 + \Delta\gamma^l. \tag{21}$$

l identifies residual stage 3 or 4. g_l is its condition generator, z is the standardized twelve-dimensional descriptor, $\Delta\gamma^l$ is the scale-increment vector, β^l is the shift vector, and γ^l is the resulting scale vector. The bracketed semicolon stacks the two output vectors. The scalar one is broadcast to every scale channel.

$$Fmod_{c,h,w}{}^l = \gamma_c{}^l F_{c,h,w}{}^l + \beta_c{}^l. \tag{22}$$

$F^l_{c,h,w}$ is the input activation at channel c and spatial position (h, w) of stage l, and $Fmod$ is its modulated output. γ_c^l and β_c^l are the scale and shift for channel c. The same pair is used across all spatial positions in that channel.

For ResNet50, the stage-3 generator has dimensions 12 → 64 → 2048, producing 1024 scales and 1024 shifts. The stage-4 generator has dimensions 12 → 128 → 4096 for 2048 channels. ReLU separates the two linear layers. The output-layer weights and biases are initialized to zero, so the

DOI: 10.54254/2755-2721/2026.37598

initial scales equal one and the shifts equal zero. Training learns how the descriptor vector adjusts the channel responses.

The direct-concatenation control supplies the pooled image vector and the same standardized descriptors to the first regression layer. Its implementation separates their linear contributions:

$$W_{cat}\,[h; z] \;+\; b \;=\; W_h h \;+\; W_z z \;+\; b. \tag{23}$$

h is the 2048-dimensional pooled visual vector and z is the standardized 12D vector. $[h;\ z]$ denotes their concatenation, W_{cat} is the first-layer weight matrix, W_h and W_z are its visual and descriptor blocks, and b is the bias. The two expressions compute the same linear transformation.

The descriptor block W_z is zero-initialized to preserve the shared initial visual prediction. The image-only control uses the visual block alone. All configurations share the visual head dimensions, the initial visual weights, and the score-output transformation.

The grouped-generator control uses four descriptor subsets: centroid measures $F_1 - F_2$, distribution $F_3 - F_5$, symmetry and regional balance $F_6 - F_8$, and projection and smoothness $F_9 - F_{12}$. Each group has its own generator. Their output increments and shifts are added at each stage:

$$\left[\Delta\gamma^{l};\ \beta^{l}\right] \;=\; \textstyle\sum_{k=1}^{4} g_{l,k}\,(z_k). \tag{24}$$

k indexes the four descriptor groups. z_k is the standardized subvector for group k, and $g_{l,k}$ is its generator at residual stage l. Summation combines four output vectors to obtain $\Delta\gamma^{l}$ and β^{l} for that stage. Each group has hidden width 16 at stage 3 and 32 at stage 4.

The group generators use the same zero-output initialization. The full-vector and grouped variants thus begin with an unchanged visual pathway and differ in how descriptor information enters the condition generators.

3.5. Output learning and training procedure

Training minimizes the Huber loss [15]. For sample i, the residual r_i is its predicted score $y_{pred,i}$ minus its human-assigned score y_i. The threshold δ is set to one score unit:

$$\mathscr{L}_{\delta}\,(r_i) = \begin{cases} \frac{1}{2}\,r_i^{2}, & |r_i| \le \delta, \\ \delta\left(|r_i| - \frac{1}{2}\,\delta\right), & |r_i| > \delta. \end{cases} \tag{25}$$

$L_{\delta}(r_i)$ is the per-sample Huber loss, r_i is the prediction residual, δ is the transition threshold, and $|\ r_i\ |$ is the absolute residual. The quadratic branch applies when $|\ r_i\ | \le \delta$ and the linear branch when $|\ r_i\ | > \delta$.

$$\mathscr{L} \;=\; \tfrac{1}{B}\,\textstyle\sum_{i=1}^{B} L_{\delta}(r_i). \tag{26}$$

$\mathscr{L}$ is the minibatch training objective, B is the minibatch size, i indexes its samples, and $L_{\delta}(r_i)$ is the corresponding per-sample loss. The sum averages the losses over the minibatch.

The quadratic branch gives smooth gradients around accurate predictions, and the linear branch limits the gradient magnitude for larger residuals. AdamW [16] updates parameters using adaptive gradient moments and decoupled weight decay. The backbone learning rate is 0.00001, while the

DOI: 10.54254/2755-2721/2026.37598

regression and FiLM layers use 0.0003. Weight decay is 0.0001, batch size is 32, and the training budget is at most 20 epochs.

Epoch selection and final evaluation are separated. In each outer fold, a fixed inner validation subset selects the epoch with highest PCC (earliest epoch on a tie). A fresh model is then initialized and trained on the complete outer training subset for that many epochs. Only this refitted model is evaluated on the outer test fold. Normalization statistics are refitted on the outer training subset for this step. Python, NumPy, PyTorch, and data-loader random states are controlled; matched configurations use the same batch ordering and common visual initialization.

4. Experiments and analysis

4.1. ResNet50 comparison protocol

The ResNet50 experiment compares image-only regression with intermediate FiLM. Section III also defines direct-concatenation and grouped-FiLM controls for examining the fusion design. The 897-image collection and shared outer folds are defined in Section III.A. A 20% inner validation subset is drawn from each outer training set using stratified sampling with random state 4200 plus the fold index. Inner validation selects the training duration, after which the model is reinitialized and fitted on the complete outer training set. Training seeds 11, 23, and 37 are shared across configurations.

This paper reports mean absolute error (MAE), mean squared error (MSE), Pearson correlation (PCC), and Spearman rank correlation (SRCC). MAE and MSE measure prediction errors in score units and squared score units. PCC measures linear association, and SRCC measures rank association with average ranks for tied values. Table 1 summarizes the image-only and intermediate-FiLM results, and Figure 2 visualizes the same four metrics.

Table 1. ResNet50 image-only and intermediate-FiLM results

Configuration	MAE ↓	MSE ↓	PCC ↑	SRCC ↑
Image only	1.3793	2.9593	0.6958	0.6721
Intermediate FiLM	1.3554	2.9211	0.7067	0.6835

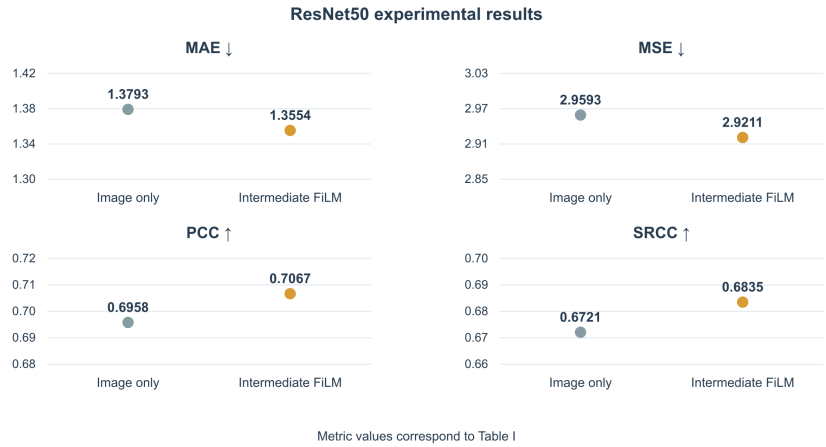

Figure 2. ResNet50 image-only and intermediate-FiLM results corresponding to Table 1 (picture credit: original)

DOI: 10.54254/2755-2721/2026.37598

Both configurations use the ResNet50 image encoder. The image-only baseline predicts scores from visual features, while intermediate FiLM conditions residual-stage features on the twelve Gestalt descriptors. This comparison evaluates the contribution of descriptor-based intermediate conditioning.

Intermediate FiLM reduces MAE from 1.3793 to 1.3554 and MSE from 2.9593 to 2.9211. PCC increases from 0.6958 to 0.7067 and SRCC from 0.6721 to 0.6835. The lower errors and higher correlations indicate a modest, consistent improvement across the four reported metrics.

4.2. Baseline and backbone comparisons

This paper additionally compares four locally implemented baselines and five BFF backbone variants. PureCNN_R18 uses ResNet-18 [13], while SE_ResNet18 combines that backbone with squeeze-and-excitation channel recalibration [17]. Trad26MLP and GCN represent handcrafted-feature and graph-based baselines. The backbone variants use ResNet-18, ResNet-34, ResNet-50, and ResNet-101 [13], and VGG16 [18]. Table 2 retains the recorded model identifiers; these citations identify the underlying architectures.

Table 2. Locally implemented baselines and backbone variants

Model	Type	MAE ↓	MSE ↓	PCC ↑	SRCC ↑
PureCNN_R18	Local baseline	1.466	3.247	0.664	0.639
Trad26MLP	Local baseline	1.942	5.781	0.067	0.124
GCN	Local baseline	1.925	5.571	0.274	0.255
SE_ResNet18	Local baseline	1.524	3.514	0.653	0.629
BFF_ResNet-18	BFF variant	1.450	3.195	0.674	0.656
BFF_ResNet-34	BFF variant	1.367	2.905	0.701	0.682
BFF_ResNet-50	BFF variant	1.415	3.017	0.695	0.675
BFF_ResNet-101	BFF variant	1.440	3.121	0.686	0.656
BFF_VGG-16	BFF variant	1.439	3.285	0.687	0.674

The BFF backbone variants in Table 2 are distinct model configurations from the intermediate-FiLM model in Table 1. Their results characterize the locally implemented baselines and the effect of backbone choice within BFF.

Among the four locally implemented baselines, PureCNN_R18 achieves the lowest MAE (1.466) and the highest PCC (0.664). Among the BFF backbone variants, ResNet34 achieves MAE 1.367 and PCC 0.701, while ResNet50 achieves MAE 1.415 and PCC 0.695. Within the BFF comparison, a deeper backbone does not consistently improve performance. Table 1 separately evaluates intermediate FiLM against an image-only ResNet50 baseline.

4.3. Multiscript visualization

The ResNet50 visualization set contains 299 out-of-fold candidates: 74 Kaishu, 168 Lishu, and 57 Xingshu. Candidate identities were fixed before training. Figure 3 presents three examples with four columns: the original glyph, CNN positive CAM, FiLM positive CAM, and FiLM negative CAM.

DOI: 10.54254/2755-2721/2026.37598

The original Grad-CAM formulation weights feature maps by spatially averaged gradients and applies ReLU to the weighted sum [19]. Figure 3 displays positive and negative response panels separately for score regression. The positive and negative labels denote score-increasing and score-decreasing responses. The heatmaps describe model sensitivity, with colors normalized within each image. Responses extending into blank regions motivate further examination of how local strokes and surrounding space jointly affect regression.

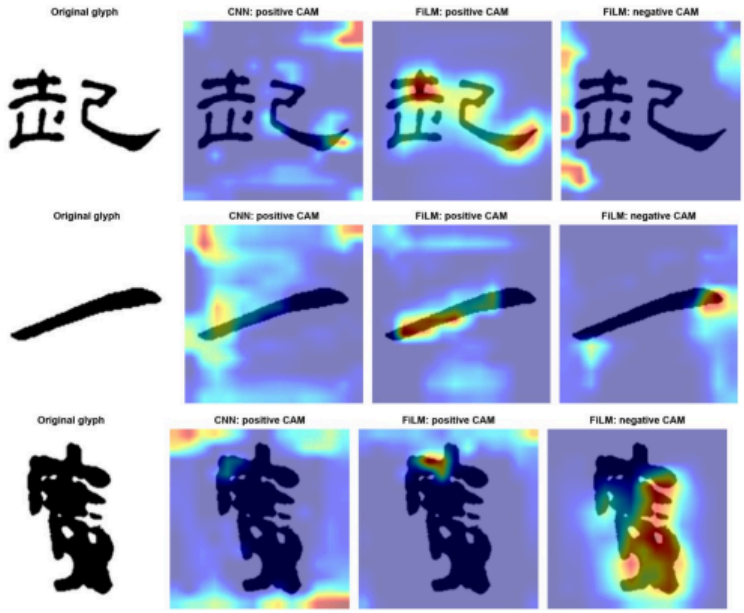

Figure 3. Directional CAM examples. Columns show the original glyph, CNN positive CAM, FiLM positive CAM, and FiLM negative CAM (picture credit: original)

4.4. Extension to whole calligraphic works

Whole-work assessment (Table 3) uses a hierarchical approach that combines local visual scoring with global composition measurements. A character-level image model scores local windows, after which score statistics are combined with geometric measurements of page layout. This preserves local stroke information while introducing the placement of characters and blank space at the composition level.

This paper evaluates this approach on 122 whole-work images. The image-only character model is frozen and applied to 128 × 128 and 256 × 256 sliding windows. The mean and standard deviation of window scores at each scale produce four local descriptors. Six additional layout descriptors measure blank-space fraction, horizontal and vertical centroid offsets, left–right symmetry difference, and horizontal and vertical projection variance. Standardized descriptors are combined with RidgeCV and evaluated by leave-one-out cross-validation.

Table 3. Whole-work assessment with local window-score statistics and layout descriptors

Whole-work representation	MAE ↓	PCC ↑	SRCC ↑
Window-score statistics	1.6540	0.6154	0.5928
Window statistics + layout descriptors	1.6311	0.6323	0.6226

DOI: 10.54254/2755-2721/2026.37598

Adding the layout descriptors reduces MAE from 1.6540 to 1.6311, increases PCC from 0.6154 to 0.6323, and increases SRCC from 0.5928 to 0.6226. These improvements support combining local visual scores with composition-level descriptors for whole-work assessment. The whole-work comparison evaluates the contribution of layout information to window-score statistics.

4.5. Future work

Future work will expand and balance the three script groups, record writer and work identities, and collect ratings from multiple annotators. These additions will support score calibration and evaluation on independent sources. Further studies will assess variability across additional training seeds, extend fusion comparisons and prior-method reproductions, and examine the contribution of different local encoders to whole-work assessment. Descriptor sensitivity to thresholding, resolution, and angular boundaries will also be examined.

5. Conclusion

This paper develops a calligraphic aesthetic-regression model that combines twelve Gestalt-inspired geometric descriptors with a ResNet50 image encoder. The descriptors summarize visual balance, ink distribution, structural correspondence, and directional rhythm. Two FiLM modules translate these measurements into intermediate channel scales and shifts. In the ResNet50 comparison, intermediate FiLM reduces MAE from 1.3793 to 1.3554 and increases PCC from 0.6958 to 0.7067, with corresponding improvements in MSE and SRCC. Local-baseline and BFF backbone comparisons further characterize alternative model configurations. In whole-work assessment, adding layout descriptors to local window-score statistics reduces MAE from 1.6540 to 1.6311 and increases PCC from 0.6154 to 0.6323. The results support integrating explicit geometric information with learned visual representations and combining local scoring with global composition measurements. Larger and more balanced collections, multiple annotators, and evaluation on independent sources will help establish how well these findings generalize across scripts and whole works. A limitation of this study is the relatively small and unbalanced dataset, together with the lack of writer identities and multiple-annotator ratings, which may restrict the generalizability of the results. Future work will expand the dataset across more scripts and independent sources, incorporate multiple annotators, and further examine the robustness of the proposed framework in whole-work assessment.

References

[1] Han, C.-C., Chou, C.-H., & Wu, C.-S. (2008). An interactive grading and learning system for Chinese calligraphy. *Machine Vision and Applications, 19*(1), 43-55. https://doi.org/10.1007/s00138-007-0076-0

[2] Sun, J. (2020). *Research and implementation of an intelligent evaluation method of Chinese calligraphy*. Master's thesis, Hubei University of Technology. (in Chinese)

[3] Hong, W. (2021). *Aesthetic scoring of Chinese character glyphs based on deep learning*. Master's thesis, Kunming University of Science and Technology. (in Chinese; title translated)

[4] Qi, Z. (2022). *A handwritten glyph evaluation method based on graph convolutional network and radical graph*. Master's thesis, Central China Normal University. (in Chinese)

[5] Arnheim, R. (2004). *Art and visual perception: A psychology of the creative eye*. 2nd Edition, University of California Press. Berkeley, CA.

[6] Palmer, S. E. (1999). *Vision science: Photons to phenomenology*. MIT Press. Cambridge, MA.

[7] Bauerly, M., & Liu, Y. (2006). Computational modeling and experimental investigation of effects of compositional elements on interface and design aesthetics. *International Journal of Human-Computer Studies, 64*(8), 670-682. https://doi.org/10.1016/j.ijhcs.2006.01.002

DOI: 10.54254/2755-2721/2026.37598

[8] Lai, C.-Y., Chen, P.-H., Shih, S.-W., Liu, Y., & Hong, J.-S. (2010). Computational models and experimental investigations of effects of balance and symmetry on the aesthetics of text-overlaid images. *International Journal of Human-Computer Studies, 68*(1-2), 41-56. https://doi.org/10.1016/j.ijhcs.2009.08.008

[9] Brachmann, A., & Redies, C. (2017). Computational and experimental approaches to visual aesthetics. *Frontiers in Computational Neuroscience, 11*, Article 102. https://doi.org/10.3389/fncom.2017.00102

[10] Perez, E., Strub, F., de Vries, H., Dumoulin, V., & Courville, A. (2018). FiLM: Visual reasoning with a general conditioning layer. *Proceedings of the AAAI Conference on Artificial Intelligence, 32*(1), 3942-3951. https://doi.org/10.1609/aaai.v32i1.11671

[11] Liu, C.-L., Yin, F., Wang, D.-H., & Wang, Q.-F. (2011). CASIA online and offline Chinese handwriting databases. *Proceedings of the 11th International Conference on Document Analysis and Recognition*, 37-41. https://doi.org/10.1109/ICDAR.2011.17

[12] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics, 9*(1), 62-66. https://doi.org/10.1109/TSMC.1979.4310076

[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[14] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*(4), 600-612. https://doi.org/10.1109/TIP.2003.819861

[15] Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics, 35*(1), 73-101. https://doi.org/10.1214/aoms/1177703732

[16] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

[17] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141.

[18] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *Proceedings of the International Conference on Learning Representations*. https://arxiv.org/abs/1409.1556

[19] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision*, 618–626. https://doi.org/10.1109/ICCV.2017.74