ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240317001402
Original scientific paper

TransUNet Image 3D Reconstruction with Hyperparameter Optimization

Xiaofang WANG*, Zhihao LUO, Mingrui GOU, Kerui MAO, Liang ZHOU

Abstract: Ancient architecture is characterized by its complexity and exquisite structure, but most existing images are in 2D format. This study proposes a TransUNet-based 3D reconstruction method with hyperparameter optimization for depth prediction to enhance the effectiveness, accuracy, and efficiency of reconstructing ancient buildings from 2D images. The method employs Restricted Boltzmann Machine (RBM) for depth prediction and an optimized ant colony algorithm for network parameter optimization. Experiments demonstrate that the proposed method achieves an average *F*1 score of 96.8% in reconstructing ancient buildings, outperforming other algorithms in terms of processing time and efficiency. The results validate the superiority of the proposed algorithm in processing images of ancient architecture, improving measurement accuracy and reducing execution time. This study contributes to the digitization and preservation of cultural heritage.

Keywords: hyperparameter optimization; neural networks; TransUNet; 3D Reconstruction

1 INTRODUCTION

Reconstructing ancient architecture from 2D images is a highly challenging task. It not only involves capturing complex structures but also ensuring high-quality reconstruction results [1]. Ancient buildings hold significant historical and cultural value, and documenting their structure and details is crucial for the preservation and transmission of cultural heritage. However, due to historical reasons, many ancient buildings are only recorded in 2D images. Transforming these 2D images into 3D reconstructions has become an important research topic in cultural heritage preservation, attracting considerable attention [2]. Traditional image processing methods, such as stereo matching [3] and multi-view geometry reconstruction [4], face limitations including high equipment requirements, low accuracy, and long execution times, which hinder their wide application [5].

With the continuous advancement of artificial intelligence technologies, deep learning has shown great promise and potential in the field of ancient architecture reconstruction. Image 3D reconstruction techniques using networks such as CNN [6], DNN [7], AlexNet [8], and GoogleNet [9] have emerged. Among these, CNNs have fewer parameters, reducing computational complexity and making them suitable for large-scale data processing. DNNs, with their deeper layers, can learn complex feature representations, benefiting the reconstruction of structures and details. AlexNet introduces *ReLU* activation functions, Dropout, and local response normalization, significantly improving the speed and accuracy of reconstruction training. GoogleNet employs parallel convolution and pooling operations, effectively enhancing computational efficiency. While each of these methods has its advantages, certain technical issues still require further optimization.

Di et al. [10] introduced a compact network based on SqueezeNet, which achieves performance comparable to AlexNet in 3D image reconstruction but with 50 times fewer parameters. This opens new pathways for 3D modeling but presents challenges related to limited size and computational intensity. Shi et al. [11] proposed a 3D reconstruction method that integrates high-precision SLAM with deep learning, utilizing camera pose estimation and keyframe information for point cloud reconstruction. While this method ensures reconstruction quality through global and local optimization, it is highly dependent on the shooting angle.

Furthermore, Zhou et al. [12] developed a method for 3D image reconstruction from multi-view radar sequences. This approach connects 3D target structures with observed images using explicit expressions of radar and optical imaging geometry and employs multi-view stereo technology to extract contour information from radar image sequences. Despite enhancing reproducibility by fusing radar and optical image features, this method is limited by its requirement for radar images, restricting its applicability. Ma et al. [13] presented a rapid 3D mesh construction method in the IoT context using oblique images. This method employs real-time adaptive octree spatial partitioning and graph-cut methods, significantly improving model reconstruction efficiency. However, the network model is sensitive to normal vectors, and its complex structure poses additional challenges.

Further research by Yang et al. [14] proposed a multi-view 3D reconstruction algorithm based on Long-Range Grouping Attention (LGA). This algorithm ensures feature richness by grouping all views for focused attention, following a divide-and-conquer principle. Nevertheless, its performance heavily relies on the quality and quantity of the views, leading to high computational complexity. Zou et al. [15] introduced a 3D Recurrent Reconstruction Neural Network (3D-R2N2) that learns the mapping from images to underlying 3D shapes using large amounts of synthetic data. This network can reconstruct 3D models from single or multiple viewpoints, offering good flexibility and robustness. However, it requires extensive computational resources and synthetic data for training, limiting its generalization capability.

To overcome the limitations of existing methods and improve the effectiveness of 3D image reconstruction, we propose a TransUNet-based method with hyperparameter optimization and multi-task learning strategies. This method addresses current shortcomings in detail capture, computational efficiency, and generalization capability. It combines joint bilateral filtering and texture-structure-aware interpolation for image preprocessing, uses a Restricted Boltzmann Machine (RBM) for depth prediction, and optimizes network parameters with an enhanced ant colony algorithm to achieve efficient and accurate 3D reconstruction.

2 RELATED WORKS

2.1 TransUNet

Introduced in 2021, TransUNet [16] has been proposed for medical 3D image segmentation. This network architecture excels in handling details and extracting global contextual features effectively. Leveraging the detail-processing strengths of TransUNet, this study adapts it for image 3D reconstruction to enhance the quality of image reconstruction. As a hybrid CNN-Transformer neural network architecture, TransUNet utilizes the high-resolution spatial information from CNN features [17] and the global context characteristics encoded by the Transformer [18] to construct an encoder for image feature extraction. Following the design principles of a U-Net architecture [19], it integrates features from the Transformer's self-attention mechanism through upsampling and skip connections, merging features of varying resolutions to accomplish image reconstruction. The architecture is illustrated in Fig. 1.

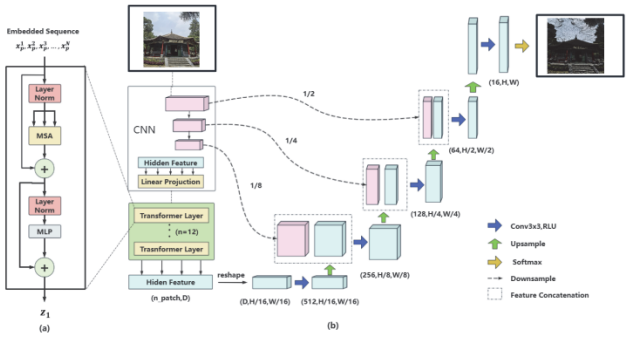


Figure 1 TransUNet network architecture

As illustrated in Fig. 1, panel (a) depicts the Transformer architecture, which is composed of a sequence embedding layer, layer normalization (Layer Norm), Multi-head Self Attention (MSA) mechanism [20], another layer of normalization, and a Multilayer Perceptron (MLP) [21]. Panel (b) showcases the encoder-decoder architecture of TransUNet. The encoder begins by utilizing a CNN to extract hidden features, which are then mapped to sequence space through linear projection for processing by Transformer layers. The decoder consists of multiple convolutional layers and upsampling operations. In the figure, H and W represent the height and width of the image, respectively, while 1/2, 1/4, and 1/8 indicate the scaling ratios of the feature maps. Arrows signify different operations: solid arrows pointing upwards indicate upsampling, solid arrows pointing downwards denote downsampling, dashed arrows represent feature concatenation, and rectangular blocks signify convolutional layers performing various operations.

The TransUNet network processes input images $F_C^1, F_C^2, ..., F_C^n$ through a CNN encoder for feature extraction. This involves convolution operations and activation functions, as described in Eq. (1).

$$F_e = ReLU\left(Conv\left(B\right)\right) \tag{1}$$

In this context, B represents the input image, while F_e denotes the features encoded by the encoder.

The feature maps outputted by the encoder are progressively downsampled to obtain features at different scales, with each downsampling step including a max pooling operation followed by a convolution operation, as outlined in Eq. (2).

$$\begin{cases} F_{ds} = MaxPool\left(F_e\right) \\ F_{ds} = ReLU\left(conv_{stride=2}\left(F_e\right)\right) \end{cases} \tag{2}$$

Here, F_{ds} signifies the features after downsampling, with a stride of 2 indicating the step size for the convolution operation.

Subsequently, the multi-scale feature maps are projected onto a serialized module through a linear layer to meet the input requirements of the Transformer, as demonstrated in Eq. (3).

$$F_{transformed} = F_{ds}W + be \tag{3}$$

Here, W represents the weight matrix of the linear mapping, be is the bias term, and $F_{transformed}$ denotes the feature sequence after mapping, serving as the input to the Transformer layer.

The Transformer layer enhances the serialized features $F_{transformed}$ through feature augmentation, based on a Multi-head Self-Attention mechanism (MSA) and feedforward neural networks. MSA, as a form of self-attention mechanism, is utilized within the Transformer layer to enhance the representational capability of the input features. It maps the input features into three sets of vectors: queries (Q), keys (K), and values (V), to capture the relational information among the input sequence, as illustrated in Eq. (4).

$$\begin{cases} Q = F_{transformed}\partial_Q \\ K = F_{transformed}\partial_k \\ V = F_{transformed}\partial_v \end{cases} \tag{4}$$

Here, ∂_Q, ∂_k and ∂_v are learnable weight matrices. Utilizing the Q, K, and V values for attention computation yields single-head attention, as shown in Eq. (5).

$$head_j = Attention\left(Q\partial_{jQ}, K\partial_{jk}, V\partial_{jv}\right) \tag{5}$$

Here, $head_j$ represents the attention from the j th head, with $Attention\left(Q, K, V\right) = soft\,max\left(\frac{QK^T}{d_k}\right)V$, where d_k is the dimension of the keys. The result of the multi-head attention is obtained by concatenating the outputs from all heads and then projecting them through a linear layer, as depicted in Eq. (6).

$$MHA\left(Q, K, V\right) = Concat\left(head_1, ..., head_h\right)\partial° \tag{6}$$

After processing by the *MHA*, the data is further processed by a feedforward neural network, which typically includes two linear transformations and an activation function, as shown in Eq. (7).

$$FFNN\left(MHA\right)=\max\left(0,\ \mathrm{MHA}\partial 1+\varnothing_1\right)\partial 2+\varnothing_2 \tag{7}$$

Each self-attention and feedforward network sub-layer includes a residual connection and normalization combination. After processing through multiple Transformer layers, a reshape operation is used to convert the data back into feature maps, as demonstrated in Eq. (8).

$$F=reshape\left(F_{out},\ \left(batch_{size},H,W,C\right)\right) \tag{8}$$

Here, $batch_{size}$ refers to the number of images processed in each batch, F_{out} represents the serialized features processed by the Transformer layer, and H, W and C respectively stand for height, width, and the number of channels.

The decoder reconstructs three-dimensional features from the feature maps F through upsampling and convolutional layers, as outlined in Eq. (9).

$$F_{3D}=ConvTranspose\left(F\right) \tag{9}$$

2.2 Ant Colony Algorithm

The Ant Colony Algorithm [22] is inspired by the Traveling Salesman Problem (TSP), leveraging the natural behavior of ants leaving pheromones along their paths to guide other ants. This process highlights that paths with higher concentrations of pheromones are more optimal.

The Ant Colony Algorithm involves four main steps: parameter initialization, pheromone update strategy, path selection strategy, and convergence criteria.

Step 1: Parameter Initialization. This step establishes the initial position of the ant colony. A weight matrix is defined, and ants are randomly placed within this matrix. The colony then searches for the optimal values within the matrix, driven by the path selection and pheromone update strategies.

Step 2: Path Selection Strategy. During their search for food, ants select paths based on the concentration of pheromones. The more ants that travel a path, the higher the pheromone concentration, and thus, the greater the likelihood that an ant will choose this path, as illustrated in Eq. (10).

$$P_{ab}=\begin{cases}\dfrac{\tau_{ab}x\cdot\eta_{ab}y}{\sum_{\varepsilon\in HQ}\tau_{ab}x\cdot\eta_{ab}y} & \varepsilon\in HQ\\ 0 & \text{other}\end{cases} \tag{10}$$

Here, P_{ab} represents the probability of choosing path ab, and $\tau_{ab}x$ denotes the pheromone concentration on the path ab.

Step 3: Pheromone Update Strategy. The concentration of pheromones decreases over time, aiming to eliminate suboptimal paths, as shown in Eq. (11).

$$\begin{cases}\tau_{ab}\left(i\right)=\rho\cdot\tau_{ab}\left(ti\right)+\Delta\tau_{ab}\\ \Delta\tau_{ab}=\sum_{k=1}^{N}\Delta\tau_{ab}^{k}\end{cases} \tag{11}$$

Here, ρ represents the evaporation coefficient.

Step 4: Convergence Criteria. During the optimization process by the ant colony, as the pheromone concentration on superior paths increases and that on inferior paths decreases, the ants will gradually converge towards a single path. This process continues until convergence criteria are met, with the path traversed by the ants representing the optimal solution.

3 RESEARCH METHOD

The study introduces a TransUNet image 3D reconstruction method optimized through hyperparameter tuning, encompassing image preprocessing, depth prediction, and 3D reconstruction of images. The process is outlined in Fig. 2.

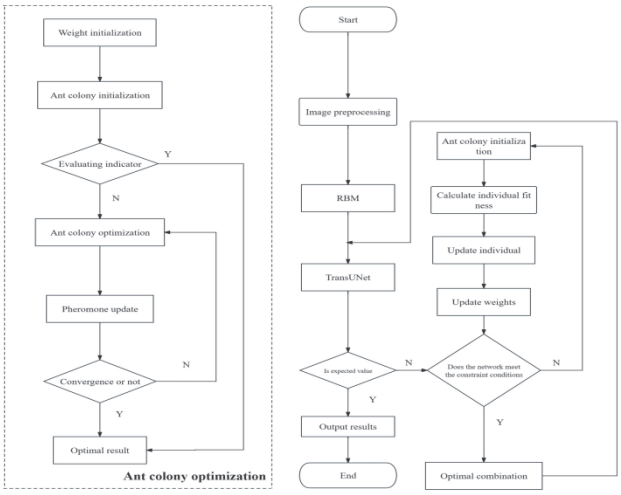


Figure 2 Algorithm flowchart

As illustrated in Fig. 1, the preprocessing phase employs joint bilateral filtering and texture and structure-aware interpolation to perform image denoising and fill in gaps, respectively. Depth prediction of the images leverages the concept of transfer learning, utilizing a trained Restricted Boltzmann Machine model for two-dimensional images. The 3D reconstruction is based on an encoder-decoder architecture, with the encoder applying CNN convolutional principles and Transformer encoding for global context feature extraction. The decoder, designed with a U-Net architecture, merges features through multi-layer upsampling and skip connections, finalizing the 3D reconstruction of images with a softmax function. The entire model training process utilizes an ant colony optimization algorithm for network parameter optimization and weight initialization.

3.1 Image Preprocessing

Due to external factors such as the photography equipment and lighting, images are prone to noise and gaps, which can directly impact the effectiveness of feature extraction and the quality of image reconstruction. To

mitigate these issues, the study employs joint bilateral filtering for noise reduction, followed by texture and structure-aware interpolation to fill in image gaps.

Joint bilateral filtering [23] is a linear filtering technique based on Gaussian filtering [24]. It processes images by replacing the value of a pixel with a weighted average of pixel values from its vicinity, as demonstrated in Eq. (12).

$$GF\left(xr\right)=\frac{1}{kS_p}\sum_{i=1}^{M}G_s\left(\left\|xr-xr_i\right\|\right)\cdot I_{(xr_i)} \tag{12}$$

In the formula, $GF(xr)$ denotes the filtered pixel value of the ancient architecture image, xr represents the current pixel, xr_i indicates the i th pixel neighboring xr), and M is the number of pixels neighboring xr. $I_{(xr_i)}$ is the grayscale value of the i th neighboring pixel, kS_p is the normalization coefficient, i.e., the standard deviation of the spatial Gaussian kernel. $\sum_{i=1}^{M}G_s\left(\left\|xr-xr_i\right\|\right)$ represents the spatial domain Gaussian function for the ancient architecture image, calculated as shown in Eq. (13).

$$G_s\left(\left\|xr-xr_i\right\|\right)=e^{\left(\frac{-\left[(xr_1-xr_2)^2+(yr_1-yr_2)^2\right]}{2\sigma_S^2}\right)} \tag{13}$$

In the equation, kS_p is used to ensure pixel weights, with kS_p in [0, 1], and its calculation is shown in Eq. (14).

$$kS_p=\sum_{i=1}^{M}G_s\left(\left\|xr-xr_i\right\|\right) \tag{14}$$

Single Gaussian filtering only considers spatial neighborhoods, which results in suboptimal edge handling. To effectively preserve the clarity of edges in ancient architecture while also suppressing noise, this study employs dual Gaussian filters as the filtering kernel, with the calculation shown in Eq. (15).

$$BF\left[I\right]_p=GF\left[xr\right]\cdot G_{re}\left(\left|I_{xr}-I_{xri}\right|\right) \tag{15}$$

In the equation, $BF[I]_p$ represents the pixel value of the ancient architecture image after filtering, and $G_{re}\left(\left|I_{xr}-I_{xri}\right|\right)$ denotes the Gaussian function in the pixel domain, with the calculation shown in Eq. (16).

$$G_{re}\left(\left|I_{xr}-I_{xri}\right|\right)=e^{\left(-\frac{\left|I_{xr}-I_{xri}\right|^2}{2\delta_{re}^2}\right)} \tag{16}$$

In the equation, δ_{re} is the standard deviation of the Gaussian function in the pixel domain of the ancient architecture image.

To preserve the rich texture and structural information of ancient architecture images, thereby avoiding overly smooth and detail-lacking reconstructed surfaces, this study proposes a texture and structure-aware interpolation method based on the eight-neighborhood pixel interpolation technique. This method estimates values using the information, texture, and structural features of the gap pixel dr and its surrounding pixels, with the algorithm processed as follows:

Step 1: Neighborhood Definition. For a pixel dr to be filled, its eight-neighborhood $NU(dr)$ is defined as the eight surrounding pixels.

Step 2: Weight Calculation. Weights based on image gradients and local variance are introduced to enhance the influence of texture and structure. The calculation is shown in Eq. (17) and Eq. (18).

$$w_s\left(q\right)=\exp\left(-\frac{\left\|\nabla I\left(q\right)-\nabla I\left(dr\right)\right\|^2}{2\sigma_s^2}\right) \tag{17}$$

$$w_r\left(q\right)=\exp\left(-\frac{\left\|L\left(q\right)-L\left(dr\right)\right\|^2}{2\sigma_r^2}\right) \tag{18}$$

Here, the image gradient weight is used to assess texture similarity, while the local variance weight evaluates structural similarity. Specifically, $\nabla I\left(q\right)$ and $\nabla I\left(dr\right)$ represent the gradient vectors of pixels q and dr, respectively, and $L(q)$ and $L(dr)$ denote the local variances. The parameters σ_s and σ_r control the distribution of weights, influencing the breadth of the weight distribution, with optimal values selected through cross-validation.

Step 3: Composite Weight Calculation. The final weights are calculated by combining the image gradient weights and local variance weights, as shown in Eq. (19).

$$w_r\left(q\right)=\propto w_s\left(q\right)-\left(1-\propto\right)w_r\left(q\right) \tag{19}$$

Here, $\propto$ is a parameter used to balance the contributions of image gradient weights and local variance weights, determining their respective contributions to the final outcome.

Step 4: Pixel Value Estimation. The value of the gap pixel dr is estimated using a weighted average method, as illustrated in Eq. (20).

$$I\left(dr\right)=\frac{\sum q\in NU\left(dr\right)w\left(q\right)\cdot I\left(q\right)}{\sum q\in NU\left(dr\right)w\left(q\right)\cdot I\left(q\right)} \tag{20}$$

Step 5: Repeat the aforementioned steps until there are no gaps left in the image.

It is important to note that while this method is suitable for filling small gaps, it has limited effectiveness in handling large gaps and severely degraded images. In such cases, the affected data images should be discarded to ensure reconstruction quality.

3.2 Image Depth Prediction

The quality of depth information extraction directly influences the results of 3D image reconstruction. Our study employs transfer learning techniques and uses the

Restricted Boltzmann Machine (RBM) [25] for depth prediction. The value of using RBM lies in its ability to automatically extract high-level features from input data, making it suitable for capturing complex structures and details. Additionally, the computational resources and labeled samples required for RBM are significantly lower compared to CNNs and monocular depth estimation methods. RBM's hidden nodes effectively capture non-linear relationships within images, enhancing depth prediction accuracy, which is crucial for 3D reconstruction of intricate ancient architecture images. RBM leverages forward and backward propagation to learn and understand the spatial relationships of various architectural components from the input images.

3.2.1 Construction of Depth Prediction Network

The process begins by inputting 2D images into the Restricted Boltzmann Machine (RBM). The visible layer nodes are initialized to match the dimensions of the input images, while the hidden layer nodes are set to one-fourth of the visible layer nodes. The weight matrix *We*, visible layer biases *A*, and hidden layer biases φ are initialized using random number generation based on a linear congruential generator, as calculated in Eq. (21).

$$X_{ni+1} = \left(AX_{ni} + \varphi\right) \bmod c \tag{21}$$

where X_{ni} is the current random number, X_{ni+1} is the next random number, and c is a modulus typically set to a large prime number.

3.2.2 Forward and Backward Propagation

The input image data is fed into the visible layer, and the hidden layer values are computed via forward propagation , as shown in Eq. (22).

$$D\left(si\right) = D\left(\sigma i\right) \cdot We + A \tag{22}$$

The hidden layer values are then back-propagated to update the visible layer, adjusting the weight matrix and hidden biases , as calculated in Eq. (23).

$$D\left(si\right) = D\left(\sigma i\right) \cdot We + \varphi \tag{23}$$

Here, $D(Si)$ and $D(\sigma i)$ represent the values of the hidden and visible nodes, respectively. These steps are repeated until the network converges or reaches the set number of iterations.

3.2.3 Training the Restricted Boltzmann Machine

The preprocessed ancient architectural image dataset is fed into the RBM for training. The learning process involves maximizing the likelihood function to evaluate the difference between the generated data and the real data, as demonstrated in Eq. (24).

$$Pk\left(\beta\right) = \sum\nolimits_{h} Pk\left(\beta, se\right) \tag{24}$$

Training the RBM involves fitting the training samples to maximize the likelihood function over the training set, as shown in Eq. (25).

$$\operatorname{argmax} L\left(\theta\right) = \operatorname{argmax} \sum\nolimits_{ne=1}^{me} \log Pk\left(\beta^{(ne)};\, \theta\right) \tag{25}$$

Gradient ascent [26] is used to solve for the partial derivatives of each parameter, as illustrated in Eq. (26).

$$\begin{aligned} &\frac{\partial \log P\left(\beta^{(ne)};\, \theta\right)}{\partial \theta} = - < \frac{\partial E\left(\beta^{(ne)};\, \omega;\, \theta\right)}{\partial \theta} > \\ &pk\left(\omega \middle| ve^{(ne)}\right) + < \frac{\partial E\left(\beta;\, \omega;\, \theta\right)}{\partial \theta} > pk\left(\beta,\, \sigma\right) \end{aligned} \tag{26}$$

Here, the gradient θ reflects the difference between the model distribution and the actual data distribution, adjusting the parameters to reduce this difference and improve the model's generation quality and accuracy.

3.3 Image 3D Reconstruction

To enhance the accuracy and robustness of 3D reconstruction from 2D images, our study utilizes an ant colony optimized TransUNet network. The depth images, containing spatial relationships predicted by the RBM, are input into the TransUNet. This network employs a hybrid CNN-Transformer encoder and a cascade upsampler decoder for image reconstruction, with network parameters optimized using an ant colony algorithm.

3.3.1 Weight Initialization

In the 3D reconstruction process using TransUNet, backpropagation plays a critical role. Backpropagation is an algorithm that adjusts network parameters by calculating the gradient of the loss function, enabling the model to learn mappings from 2D images to 3D structures more effectively. It utilizes gradient descent [27] to update the weights, but the process can lead to rapid convergence and local optima issues due to equal gradient values. To address these problems, we use an ant colony algorithm for weight initialization, improving the training performance.

Traditional ant colony algorithms initialize the weight matrix by randomly distributing ants, which can affect the optimization performance due to its randomness. Instead, we initialize the ant positions using a Gaussian distribution,as shown in Eq. (27).

$$\left(yg\right) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{\left(yg-\mu\right)^2}{2\sigma^2}\right) \tag{27}$$

Afterwards, ants select their paths based on pheromone concentration. However, traditional update strategies, which are based on a greedy approach, tend to lead the ant colony towards local optima. Therefore, this study proposes using the average of the maximum and minimum pheromone concentrations as an alternative to solely relying on the highest and lowest pheromone concentrations, as demonstrated in Eq. (28).

$$R_{\max} = R_{\min} = \frac{\left(R_{\max} + R_{\min}\right)}{2} \tag{28}$$

Here, $R_{\max}$ and $R_{\min}$ respectively represent the maximum and minimum pheromone concentrations. During this process, the location with $R_{\min}$ receives an increment of one to its "minimum pheromone amount" (i.e., the number of times this point has the lowest pheromone concentration during the current pathfinding process). When this value reaches a predetermined limit, the path is deemed the worst and is directly discarded, with the pheromone concentration on this path set to −1.

To prevent ants from redundant searching, the study employs two different strategies for two scenarios involving ant colonies.

(1) If two ants, belonging to different boundary groups, face each other and their next step would lead them to the same pixel, they choose to halt their next movement.

(2) If an ant encounters the path previously traversed by another ant, it also stops its next movement.

To avoid the inefficiency caused by repeated searches, a taboo list is established for the entire ant search process, recording the paths traversed by ants. If a path has already been taken by an ant, it is discarded, and a new selection is made, continuing until all paths have been explored. The optimization results of the ant colony then serve as the initial values for the network.

3.3.2 TransUNet 3D Reconstruction

The encoder initially utilizes a Convolutional Neural Network (CNN) to extract feature maps from the input images of ancient architecture. The CNN module consists of three convolutional layers, each with a 3x3 kernel and a *ReLU* activation function. The extracted feature maps are then serialized and fed into the Transformer. The Transformer architecture uses a multi-head attention mechanism for deep information extraction and feature enhancement. Our network comprises 12 Transformer layers, each featuring eight heads for attention calculation and a feature embedding dimension of 512, ensuring high representation capacity for complex features.

Post-processing involves decoding the serialized features through a cascade upsampler, consisting of four consecutive convolutional and upsampling layers. Each upsampling layer includes a 2× upsampling operation, a 3 × 3 convolutional layer, and a *ReLU* activation function. After upsampling, the results are skip-connected with the corresponding CNN features from the encoder to aggregate features at different resolutions. Finally, a 3 × 3 convolutional layer and a *ReLU* layer further refine the features, and a Softmax layer converts the feature maps into probability distributions across different categories, thus predicting the 3D reconstruction of the input images.

3.3.3 Hyperparameter Optimization

During the model training process, to address the impact of manually set network parameters on reconstruction performance, we apply the enhanced ant colony optimization algorithm proposed in Section 3.3.1 for adaptive hyperparameter optimization [28, 29]. This enhancement addresses issues like slow convergence and local optima in traditional ant colony algorithms. Hyperparameter Optimization Process:

Ant Colony Initialization: Initialize the ant colony using a Gaussian distribution, with each ant representing a set of hyperparameters.

Calculate Fitness: Assess the fitness value of each ant to evaluate the effectiveness of its hyperparameter combination.

Update Positions: Based on fitness results, ants select their movement direction in the hyperparameter space, guided by pheromone concentrations. Paths with higher fitness have higher pheromone concentrations.

Update Pheromones: After evaluating all hyperparameter combinations, update the pheromone concentrations. Use the mean of the maximum and minimum pheromone values to avoid local optima.

Constraint Evaluation: Determine if the hyperparameter combination meets network constraints. If it does, output the results. If not, reinitialize the ant colony and adjust the search strategy.

The hyperparameter combinations include both fixed and adaptive sets. The fixed parameters consist of the initial iteration count, activation function, convolution kernel size, and convolution stride; adaptive candidate parameters include the learning rate, learning decay rate, batch size, the number of nodes in the fully connected layer, and the dropout rate in the fully connected layer. The candidate and fixed values for these parameters are as shown in Tab. 1.

Table 1 Hyperparameter selection range

Parameter Adaptive Candidate Values		Parameter Fixed Values	
Hyperparameters	Value Range	Hyperparameters	Value
Learning Rate	{0.1, 0.01, 0.001, 0.0001}	Convolution Kernel Size	3 × 3
Learning Decay Rate	{1e−6, 1e−5, 1e−4}	Activation Function	*ReLU*
Batch Size	{16, 32, 64, 128}	Initial Iteration Count	200
Number of Nodes in the Fully Connected Layer	{64, 128, 256, 512, 1024}	Convolution Stride	(2, 2)
Dropout in the Fully Connected Layer	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6}		

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

The experiments were conducted on a Dell Precision T7960 workstation, which boasts significant specifications including an Intel Xeon W7-3465X 28-core processor at 2.5 GHz, equipped with 1TB of solid-state drive (SSD) storage and 16 TB of hard disk drive (HDD) storage solutions, along with four high-end NVIDIA RTX 6000 Ada graphics cards, running on a Linux operating system. The datasets utilized include: Buildings in Off-Nadir Aerial Images dataset, Chinese Academy of Sciences 3D Reconstruction dataset, and a custom-built Wuhou Shrine dataset. The model was trained on a curated selection from the Buildings in Off-Nadir Aerial Images and WHU Building Dataset, which are open-source, and validated on the Chinese Academy of Sciences 3D Reconstruction

dataset and the custom-built Wuhou Shrine dataset, totaling 116,060 architectural images. This includes 69,700 images in the training set, 23,200 images in the test set, and 23,160 images in the validation set.

4.2 Hyperparameter Selection

For the TransUNet, 30 rounds of hyperparameter optimization were set, with the network's loss value continuously decreasing during model training, ultimately approaching convergence. Experimental validation showed that when the network converged, the loss value of TransUNet approached 0.1. The study selected a loss value of 0.1 for iterative training, utilizing ant colony algorithms to calculate the model's hyperparameters, resulting in the optimal hyperparameter combination, as shown in Tab. 2.

Table 2 TransUNet network hyperparameter configuration

Hyperparameters	Value
Learning Rate	0.001
Learning Rate Decay	1e−6
Batch Size	64
Number of Nodes in the Fully Connected Layer	1024
Dropout in the Fully Connected Layer	0.3

From the table above, it is evident that the optimal results for training ancient architecture 3D reconstruction are achieved when the network's hyperparameters learning rate, learning rate decay, batch size, number of nodes in the fully connected layer, and dropout in the fully connected layer are set to 0.001, 1e−6, 64, 1024, and 0.3, respectively.

4.3 Experimental Results and Analysis

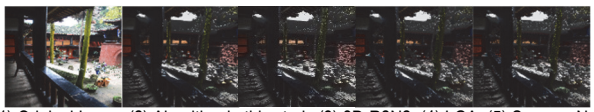
To validate the effectiveness of the algorithm, the study conducted reconstruction training on SqueezeNet [10], LGA [14], and 3D-R2N2 [15] under the same experimental conditions. The Shangqing Palace and Wuhou Shrine were selected for verification, with results presented in Fig. 3 to Fig. 5.

(1) Original Image (2) Algorithm in this study (3) 3D-R2N2 (4) LGA (5) SqueezeNet

Figure 3 Comparison results for Wuhou Shrine

(1) Original Image (2) Algorithm in this study (3) 3D-R2N2 (4) LGA (5) SqueezeNet

Figure 4 Comparison results for Shangqing Palace Di Fu De



(1) Original Image (2) Algorithm in this study (3) 3D-R2N2 (4) LGA (5) SqueezeNet

Figure 5 Comparison results for Shangqing Palace Langfang

The results illustrate the 3D reconstruction performance of four algorithms on images from Wuhou Shrine, Shangqing Palace Defoct, and Shangqing Palace Corridor. The original images are shown in (1), while (2) to (5) display the reconstructions using our proposed method, 3D-R2N2, LGA, and SqueezeNet, respectively. The results indicate that SqueezeNet's reconstructed images lack fine details. This is due to its simpler network structure, which, although advantageous in processing time, struggles to capture subtle feature relationships, leading to poorer reconstruction quality. SqueezeNet is thus more suitable for scenarios with high computational resource constraints and lower detail requirements. LGA outperforms SqueezeNet by capturing long-range dependencies through a long-range grouping attention mechanism, enabling the model to focus on critical features. However, LGA still falls short in detail processing. Consequently, LGA is better suited for applications that need attention to global features but also require a high level of detail. 3D-R2N2 surpasses both SqueezeNet and LGA in detail reconstruction, even reconstructing text in images. This is because 3D-R2N2 uses a novel recurrent neural network that can reconstruct images even without texture. While it excels in detail processing, 3D-R2N2 demands more computational resources and processing time, making it suitable for high-precision, detail-rich tasks requiring significant computational power.

Our proposed algorithm demonstrates significant advantages in detail processing and reconstruction quality, surpassing other methods. The key features include: Firstly, Texture and Structure-Aware Interpolation enhances structural and textural features, aiding in the capture of fine details and complex structures. Secondly, Self-Attention Mechanism and Skip Connections capture long-distance dependencies and focus on critical features, preserving detailed information and improving reconstruction performance. Finally, RBM for Depth Prediction automatically extracts high-level features from 2D images, capturing complex structures and details that traditional methods struggle to identify, thereby enhancing the understanding and reconstruction quality of intricate structures.

4.3.1 Algorithm Performance Analysis

To validate the effectiveness of the algorithm, the study inputs 5000 images from Shangqing Palace and Wuhou Shrine into SqueezeNet [10], LGA[14], 3D-R2N2 [15], and the algorithm proposed in this research for reconstruction. The comparison and analysis are conducted based on recall rate, IoU, *F*1 score, and error rate, as shown in Tab. 3.

Table 3 Quantitative Comparison Experiment Result for 2D Images Across Different Algorithmic Models

Algorithms	Recall Rate	IoU	*F*1 Score	Error Rate
SqueezeNet	0.918	0.935	0.916	0.049
LGA	0.941	0.940	0.938	0.048
3D-R2N2	0.956	0.946	0.942	0.044
Algorithms in this study	0.975	0.952	0.968	0.040

Recall measures the proportion of correctly reconstructed 3D points. From Tab. 3, SqueezeNet has the lowest recall at 0.918, followed by LGA, while our proposed method has the highest recall, demonstrating superior performance in capturing complex structures and fully reconstructing ancient architectural details. This effectiveness is due to RBM's ability to capture deep features in 2D images, allowing the model to

comprehensively recognize and reconstruct image details. Additionally, the self-attention mechanism in training helps the model capture long-distance dependencies, improving recognition and reconstruction capabilities for complex structures.

IoU assesses the overlap between predicted results and ground truth. Our method's IoU is slightly higher than others, indicating better reconstruction of object models, geometric shapes, and error filtering or correction, enhancing spatial accuracy, completeness, and robustness. This success is attributed to texture and structure-aware interpolation, which excels in filling missing information and improving geometric accuracy. The optimized ant colony algorithm resolves local optima and gradient descent errors by optimizing hyperparameters and network parameters, ensuring efficient data filtering and correction during reconstruction.

*F*1 score, the harmonic mean of precision and recall, evaluates the overall performance of the reconstruction model. Our method achieves the highest *F*1 score, with improvements of 5.6%, 3.2%, and 2.7% over SqueezeNet, LGA, and 3D-R2N2, respectively. This is due to the U-Net encoder-decoder architecture and skip connections, which effectively extract and fuse features at different scales, enhancing detail recognition and reconstruction. The Transformer layer captures global context information, improving detail restoration and overall reconstruction, ensuring higher precision and recall.

Error rate measures the proportion of incorrect predictions. Our method achieves the lowest error rate, with an average improvement of 9.1% over 3D-R2N2, demonstrating an advantage in avoiding erroneous reconstructions and providing results closer to reality. This is due to the joint bilateral filter effectively reducing image noise while preserving edge information, providing clear and accurate input for subsequent depth prediction and reconstruction. RBM quickly extracts deep features, accelerating the depth information acquisition process and reducing the computational burden in noise handling, thereby lowering the error rate.

4.3.2 Comparison of Algorithm Running Time and Efficiency

The study compares SqueezeNet [10], LGA [14], 3D-R2N2 [15], and the algorithm proposed in this research by conducting 3D reconstruction on 1000 two-dimensional images of ancient buildings within the same training, testing, and validation sets. It observes the time taken for reconstruction, the number of image frames processed per second, and the peak speed per second, with results presented in Tab. 5.

Table 4 Experimental results comparing 2D to 3D running time and efficiency of different algorithm models

Algorithms	TotalTime / s	FPS	FLOPS
SqueezeNet	56.37	65	53.81
LGA	63.89	58	54.38
3D-R2N2	60.93	60	58.97
Algorithms in this study	55.18	70	63.15

Based on the results, the proposed algorithm demonstrates the fastest processing speed among the four tested methods, with a conversion time of 55.18 seconds. This efficiency is due to the optimized ant colony algorithm, which reduces unnecessary computational overhead and enhances training and inference efficiency. Additionally, the RBM and U-Net architectures quickly extract deep features, significantly reducing reconstruction time.

The algorithm also achieves a 7.08% higher FLOPS compared to the best-performing 3D-R2N2, indicating superior computational efficiency and resource utilization. This improvement stems from the joint bilateral filter's ability to reduce noise while preserving edge information, providing clear input for depth prediction and 3D reconstruction, thus minimizing computational load. The combination of U-Net and Transformer effectively extracts and utilizes global contextual features, optimizing the feature extraction and reconstruction process.

Furthermore, the proposed method exhibits the best FPS, outperforming SqueezeNet, LGA, and 3D-R2N2 by 17.35%, 16.12%, and 7.1%, respectively. This advantage is attributed to the pre-trained RBM model used for depth prediction, which rapidly extracts deep features, optimizing overall computational time and efficiency. The combination of CNN and Transformer encoders effectively extracts global contextual features, while the U-Net-based decoder achieves feature fusion through multi-layer upsampling and skip connections, enhancing feature extraction and reconstruction efficiency.

4.4 Ablation Experiment

The study introduces a 3D reconstruction technique that leverages the strengths of Transformer, UNet [19], and RBM [23]. To assess the performance of the algorithm, an ablation study was conducted, where 1,000 images from Shangqing Palace and Wuhou Shrine were reconstructed using four different algorithms. The outcomes were analyzed using seven metrics for performance and efficiency evaluation, as presented in Tab. 5.

Table 5 Comparison of Ablation Experiments for 2D to 3D Reconstruction

Algorithms	Recall Rate	IoU	*F*1 Score	Error rate	Total Time / s	FPS	FLOPS
RBM	0.898	0.841	0.887	0.068	102.18	36.12	30.15
Transformer	0.912	0.905	0.918	0.056	65.65	48.11	41.23
UNet	0.922	0.919	0.924	0.051	60.28	54.15	50.63
Algorithms in this a study	0.975	0.952	0.968	0.040	55.18	70	63.15

Based on the table above, it is evident that our research algorithm outperforms three basic algorithms in performance evaluation metrics such as Recall Rate, IoU, *F*1 Score, and Error rate, as well as efficiency evaluation metrics including Total Time, FPS, and FLOPS. This superiority stems from the structural integration of RBM, Transformer, and U-shaped concepts, leveraging the advantages of each to enhance reconstruction performance, accuracy, and robustness. Additionally, employing ant colony optimization algorithm for optimizing network parameters further enhances the model's performance. By intelligently searching for optimal network parameter configurations, training time is reduced and overall model efficiency is improved. Our research algorithm demonstrates superior performance in 3D reconstruction tasks, particularly in enhancing reconstruction accuracy,

reducing errors, and accelerating processing speed. In summary, the research algorithm exhibits strong robustness.

5 CONCLUSION

This study introduces a novel TransUNet-based 3D reconstruction method with hyperparameter optimization for converting 2D images of ancient architecture into 3D representations. The proposed method combines joint bilateral filtering, texture-structure-aware interpolation, depth prediction using Restricted Boltzmann Machines, and an optimized ant colony algorithm for network parameter optimization. Experimental results demonstrate the superiority of the proposed method, achieving an *F*1 score of 96.8% in reconstructing ancient buildings and outperforming other state-of-the-art algorithms in terms of runtime and reconstruction efficiency. Despite its promising performance, the proposed method has some limitations, such as reduced reconstruction quality under low-light conditions and challenges in handling diverse image datasets. Future research should focus on integrating more advanced techniques for detail processing and robustness, exploring the algorithm's applicability across other domains, and utilizing various data types for 3D reconstruction. In conclusion, this study contributes to the digitization and preservation of cultural heritage by providing an efficient and accurate method for reconstructing ancient architecture from 2D images. The proposed TransUNet-based approach with hyperparameter optimization has the potential to significantly enhance the 3D documentation of historical sites and artifacts.

6 REFERENCES

[1] Arvanitis, N. (2021). Mapping ancient rome and teaching 3D technologies: Paul Bigot's bronze model at the institut d'art et d'archéologie in Paris. *Teaching Classics in the Digital Age.* https://doi.org/10.38072/2703-0784/p28

[2] Jun, Y., Shaohua, W., Jiayuan, L., & Qingwu, H. (2014). Research on fine management and visualization of ancient architectures based on integration of 2D and 3D GIS technology. *IOP Conference Series: Earth and Environmental Science, 17*, 012168. https://doi.org/10.1088/1755-1315/17/1/012168

[3] Scharstein, D., Szeliski, R., & Zabih, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*. https://doi.org/10.1109/smbv.2001.988771

[4] Szeliski, R. (2011). Computer vision algorithms and applications. *Springer London*. https://doi.org/10.1007/978-1-84882-935-0

[5] Furukawa, Y. & Ponce, J. (2010). Accurate, dense, and robust Multiview Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(8), 1362-1376. https://doi.org/10.1109/tpami.2009.161

[6] Zaidi, S. S., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing, 126*, 103514. https://doi.org/10.1016/j.dsp.2022.103514

[7] Wang, N., Hu, X., Zhu, F., & Tang, J. (2020). Single-view 3D Reconstruction Algorithm Based on View-aware. *Journal of Electronics & Information Technology, 42*(12), 3053-3060.

[8] Shi, Z., Meng, Z., Xing, Y., Ma, Y., & Wattenhofer, R. (2021). 3D-RETR: End-to-end single and multi-view 3D reconstruction with Transformers. *arXiv.org*.

[9] Vergauwen, M., & Van Gool, L. (2006). Web-based 3D Reconstruction Service. *Machine Vision and Applications, 17*(6), 411-426. https://doi.org/10.1007/s00138-006-0027-1

[10] Di, X. & Yu, P. (2017). 3D reconstruction of simple objects from a single view Silhouette image. *arXiv.org.*

[11] Shi C., Chen J., & Luo R. (2022). Three-dimensional Monocular Image Reconstruction of Indoor Scene Based on Neural Network. *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 445-449. https://doi.org/10.1109/ICCASIT55263.2022.9986840

[12] Zhou, Y., Zhang, L., Xing, C., Xie, P., & Cao, Y. (2019). Target three-dimensional reconstruction from the multi-view radar image sequence. *IEEE Access, 7*, 36722-36735. https://doi.org/10.1109/access.2019.2905130

[13] Ma, D., Li, G., & Wang, L. (2018). Rapid reconstruction of a three-dimensional mesh model based on oblique images in the internet of things. *IEEE Access, 6*, 61686-61699. https://doi.org/10.1109/access.2018.2876508

[14] Yang, L., Zhu, Z., Lin, X., Nong, J., & Liang, Y. (2023). Long-range grouping transformer for multi-view 3D reconstruction. *arXiv.org*. https://doi.org/10.1109/ICCV51070.2023.01674

[15] Zou Q. Y. & Liu F. Y. (2023).3D reconstruction of optical building images based on improved 3D-R2N2 algorithm. *Tehnicki Vjesnik- Technical Gazette, 30*(5). https://doi.org/10.17559/tv-20230323000469

[16] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., & Zhou, Y. (2021). TransUNet: Transformers make strong encoders for medical image segmentation. *arXiv.org*.

[17] Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., & Rueckert, D. (2019). Attention Gated Networks: Learning to leverage salient regions in medical images. *Medical Image Analysis, 53*, 197-207. https://doi.org/10.1016/j.media.2019.01.012

[18] Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., Zhang, Y., Shi, Z., Fan, J., & He, Z. (2022). A survey of Visual Transformers. *arXiv.org.*

[19] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: Learning dense volumetric segmentation from sparse annotation. *arXiv.org*. https://doi.org/10.1007/978-3-319-46723-8_49

[20] Zhao, H., Jia, J., & Koltun, V. (2020). Exploring self-attention for image recognition. *arXiv.org*. https://doi.org/10.1109/CVPR42600.2020.01009

[21] Baum, E. B. (1988). On the capabilities of Multilayer Perceptrons. *Journal of Complexity, 4*(3), 193-215. https://doi.org/10.1016/0885-064x(88)90020-9

[22] Nan, Y. (2012). An improved ant colony optimization algorithm based on immunization strategy. *Advanced Materials Research, 490-495*, 66-70. https://doi.org/10.4028/www.scientific.net/amr.490-495.66

[23] Jith, O. U. & Babu, R. V. (2014). Joint bilateral filtering based non-local means image denoising. *2014 International Conference on Signal Processing and Communications (SPCOM)*. https://doi.org/10.1109/spcom.2014.6984014

[24] Chowdhury, D., Das, S. K., Nandy, S., Chakraborty, A., Goswami, R., & Chakraborty, A. (2019). An atomic technique for removal of gaussian noise from a noisy gray scale image using lowpass-convoluted gaussian filter. *2019 International Conference on Opto-Electronics and Applied Optics (Optronix)*. https://doi.org/10.1109/optronix.2019.8862330

[25] Lu, M., Xiao, X., Song, H., Liu, G., Lu, H., & Kikkawa, T. (2021). Accurate construction of 3-D numerical breast models with anatomical information through MRI scans.

Computers in Biology and Medicine, 130, 104205. https://doi.org/10.1016/j.compbiomed.2020.104205

[26] Muhsen, H. & Tanninah, I. (2021). Analysis and simulation of maximum power point tracking based on gradient ascent method. *12th International Renewable Engineering Conference (IREC)*. https://doi.org/10.1109/irec51415.2021.9427806

[27] Wang, X., Yan, L., & Zhang, Q. (2021). Research on the application of gradient descent algorithm in machine learning. *International Conference on Computer Network, Electronic and Automation (ICCNEA)*. https://doi.org/10.1109/iccnea53019.2021.00014

[28] Wan, S., Liang, Y., & Zhang, Y. (2018). Deep convolutional neural networks for diabetic retinopathy detection by image classification. *Computers & Electrical Engineering, 72*, 274-282. https://doi.org/10.1016/j.compeleceng.2018.07.042

[29] Xiao, X., Yan, M., Basodi, S., Ji, C., & Pan, Y. (2020). Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. *arXiv.org*.

Contact information:

Xiaofang WANG
(Corresponding author)
Geely University of China,
Chengdu, Sichuan 641423, China
E-mail: 939549393@qq.com

Zhihao LUO
Geely University of China,
Chengdu, Sichuan 641423, China

Mingrui GOU
Geely University of China,
Chengdu, Sichuan 641423, China

Kerui MAO
Geely University of China,
Chengdu, Sichuan 641423, China

Liang ZHOU
Geely University of China,
Chengdu, Sichuan 641423, China