Computer Science and Information Systems 21(4):1457–1481 https://doi.org/10.2298/CSIS230725043Z

Enhancing Architectural Image Processing: A Novel 2D to 3D Algorithm Using Improved Convolutional Neural Networks

Qianying Zou[1], Fengyu Liu[2,⋆], and Yuan Liao[1]

[1] Geely University of China
Chengdu, 641423, China
{56471843,29497521}@qq.com
[2] Geely University of China
Chengdu, 641423, China
i@liufengyu.cn

Abstract. In light of the escalating advancements in architectural intelligence and information technology, the construction of smart cities increasingly necessitates a higher degree of precision in architectural measurements. Conventional approaches to architectural measurement, characterized by their low efficiency and protracted execution time, need to be revised to meet these burgeoning demands. To address this gap, we introduce a novel architectural image processing model that synergistically integrates Restricted Boltzmann Machines (RBMs) with Convolutional Neural Networks (CNNs) to facilitate the conversion of 2D architectural images into 3D. In the implementation phase of the model, an initial preprocessing of the architectural images is performed, followed by depth map conversion via bilateral filtering. Subsequently, minor voids in the images are rectified through a neighborhood interpolation algorithm. Finally, the preprocessed 2D images are input into the integrated model of RBMs and CNNs, realizing the 2D to 3D conversion. Experimental outcomes substantiate that this novel model attains a precision rate of 97%, and significantly outperforms comparative algorithms in terms of both runtime and efficiency. These results compellingly corroborate our model's superiority in architectural image processing, enhancing measurement accuracy and drastically reducing execution time.

Keywords: Building image; Boltzmann machine; Convolution neural network; Bilateral filtering; Neighborhood difference; 2D to 3D.

1. Introduction

As smart cities undergo rapid development, the digitization of architecture has emerged as a pivotal trend in modern urban development and transformation within the real estate industry. In this context, the enhancement of architectural measurement accuracy becomes increasingly critical. However, traditional methods of measurement, such as manual and mechanical techniques, are inefficient and cumbersome, failing to meet the demands of contemporary society [25]. These approaches are labor-intensive and time-consuming and

⋆ Corresponding Author

fall short in ensuring accuracy when dealing with complex architectural forms. Furthermore, the increasing complexity of modern architectural forms exacerbates the limitations of conventional image processing methods, including low accuracy and extended execution time [33].

In the realm of three-dimensional reconstruction, deep learning has manifested significant potential and a broad spectrum of application prospects. Initially, theoretical advantages of deep learning can be observed. For instance, the article [23] introduces a Convolutional Restricted Boltzmann Machine model for image segmentation. Despite its high computational complexity and slow training, hindering its widespread application in 3D reconstruction, the theoretical merits of this approach warrant attention. Secondarily, deep learning technologies have also unveiled new possibilities in practical applications. The article [37] employs an AlexNet Convolutional Neural Network model infused with transfer learning principles, maintaining high accuracy even with limited data, thereby offering a novel perspective for 3D reconstruction. However, certain technical challenges remain to be addressed. For example, the image transformation time in the 11-layer deep Convolutional Neural Network model built upon the GoogLeNet model, as described in the article [43], is excessively lengthy, compromising real-time application efficiency. A small DNN architecture called SqueezeNet is proposed in the literature [30], which deep neural networks for 3D reconstruction applications have improved accuracy, achieving comparable accuracy to AlexNet on the ImageNet dataset but with 50 times fewer number of parameters, which opens up a new path for efficient and accurate 3D model construction, but the architecture suffers from size constraints, computationally intensive problems. A new channel pruning method, Xception, is proposed in the literature [28], which not only significantly reduces the cumulative error, but also enhances the compatibility of the network with a variety of architectures, and can be used to accelerate very deep convolutional neural networks, which significantly improves the efficiency of the network operation while maintaining the accuracy, but the model has limitations in the small-sample constraints, and has a large number of parameters. An efficient model MobileNet for mobile and embedded vision applications is proposed in the literature [29], which uses depth-separable convolution to build lightweight deep neural networks and has wide applicability and superior performance in advanced vision tasks such as 3D reconstruction, but has significant limitations in terms of computational and spatial complexity.

Further research includes the multi-view stereo matching method based on deep learning proposed in the article [10], achieving precise 3D reconstruction through multi-angle image fusion; the in-depth exploration of deep learning in point cloud data processing in the article [24], resulting in fine-grained 3D model reconstruction; and the article [32] overcoming some of the challenging issues in traditional 3D reconstruction by combining deep learning with geometric learning. While these studies offer new perspectives and methods, such as those discussed in articles [31] and [9] concerning indoor and outdoor architectural 3D reconstruction, challenges persist in model generalization, data acquisition, and processing. Consequently, there remains a necessity to explore more efficient and precise methodologies in the field of architectural measurement and to identify suitable deep learning models for solving associated problems. This will not only enhance the accuracy of architectural measurements but also contribute to breakthroughs in the development of architectural digitization. Within this context, the processing of architectural images from two to three dimensions is of particular significance. By transforming

2D architectural images into 3D models, one can both improve measurement accuracy and significantly enhance efficiency, thereby fulfilling the needs of contemporary society [26] [13] [1]. When selecting deep learning models, factors such as complexity, training speed, and accuracy should be considered [5] [20], and the impact of training data volume on model performance should be noted [35] [17].

This study introduces an innovative approach based on a deep learning model aimed at generating corresponding 3D architectural images from inputted 2D architectural images. Compared to existing deep learning methods for 3D reconstruction, the distinctiveness and innovative contributions of this research can be summarized in three main aspects. Firstly, in terms of model architecture, this study amalgamates Restricted Boltzmann Machines (RBMs) with Convolutional Neural Networks (CNNs), creating a stark contrast with standalone deep learning models such as GoogLeNet and AlexNet mentioned in the literature. Secondly, the image preprocessing stage in this research incorporates bilateral filtering algorithms and neighborhood interpolation algorithms, specific preprocessing techniques not explicitly discussed in existing articles. Lastly, this research focuses on the transformation of 2D architectural images into 3D, whereas the scope of literature encompasses a broader array of 3D reconstruction applications, including image segmentation and multi-view stereo matching. The confluence of these three aspects ensures a marked distinction of this study in terms of model architecture, preprocessing techniques, and application domain when compared to extant research, thereby exhibiting unique innovative value. More specifically, the workflow of the model is divided into two modules: image preprocessing and model transformation. In the image preprocessing module, depth acquisition and void-filling are achieved for the original 2D architectural images through bilateral filtering and neighborhood interpolation. Subsequently, in the model transformation module, these preprocessed images are input into the fused RBM and CNN model for 3D conversion. This targeted and specialized approach not only enhances the accuracy of measurements but also significantly boosts efficiency, thus better catering to the specific demands of architectural measurement in modern society.

2. Related Work

2.1. Bilateral Filtering Algorithm

Filtering constitutes a crucial technique in image processing [11], effectively eliminating image disturbances such as noise and blur to enhance image quality. Filters, encompassing both frequency-domain and spatial-domain filters, smooth the image by summing or convolving the pixels, thereby reducing noise and unnecessary detail. Frequency-domain filters operate in the image frequency domain, eliminating specific frequency components; spatial-domain filters operate in the spatial domain, altering pixel values, as indicated in equation1.

$$GC[I]_p = \sum G_\sigma(\|p - q\|) I_q \tag{1}$$

Gaussian filtering represents a significant image processing technique that accomplishes linear smoothing of the image through weighted averaging, effectively eliminating Gaussian noise and enhancing image quality. In Gaussian filtering, the magnitude

of the weights is associated with the spatial distance between pixels, implying that pixels closer to the center pixel exert greater influence on the filtering outcome due to their larger weights. However, this approach has its limitations. If the distance between pixels is excessively large, even adjacent pixels could exhibit considerable disparities in their weights [39] This could result in a loss of detail in the filtering outcome, diminishing the local visual effects of the image. More importantly, such blurring in the filtering results could adversely impact subsequent image processing steps. Therefore, when employing Gaussian filtering, it is imperative to meticulously select appropriate weights to ensure the quality of the filtering outcome.

Bilateral filtering is a nonlinear filtering technique extensively employed in image processing, introduced by Tomasi et al. in 1998 [38]. The uniqueness of bilateral filtering lies in its simultaneous consideration of both the image domain and the value range, enabling it to better preserve image detail during processing. The definition of bilateral filtering closely parallels that of Gaussian filtering, both being realized through weighted averaging based on neighborhood definitions [19]. However, the method of weight measurement in bilateral filtering diverges from that in Gaussian filtering. In bilateral filtering, researchers opt for dual weight values to handle pixel values, thereby more adequately accounting for the variances between adjacent pixels. This approach, nevertheless, has its limitations. If the discrepancies between two pixels are excessively large, they may reciprocally affect each other, inducing unwarranted errors. To circumvent this, researchers introduce a sufficient number of influencing factors in both pixel values and spatial positions to ensure the quality of the filtering result. This method yields favorable outcomes when applied to images with complex textures and details. The definition of bilateral filtering is as shown in equation 2.

$$BF[I]_p = \frac{1}{W_p} \sum\nolimits_{q \in S} G_{\sigma s}(\|p - q\|) G_{\sigma r}(|I_p - I_q|) I_q \tag{2}$$

Herein, W_p represents the normalization factor; I_q denotes the input image; $BF[I]_p$ signifies the filtered image; $\sum\limits_{q \in S} G_{\sigma s}\,||\mathrm{p} - \mathrm{q}||$ is the spatial weight function; and $G_{\sigma r}\,(|I_p - I_q|)\,I_q$ is the distance weight, with the sum of the weights equating to 1, as illustrated in equation 3.

$$W_p = \sum\nolimits_{q \in S} G_{\sigma s}(\|p - q\|) G_{\sigma r}(|I_p - I_q|) \tag{3}$$

Bilateral filtering is particularly well-suited for the preprocessing of depth maps [21]. Depth maps, whether directly acquired through cameras or generated algorithmically, are susceptible to noise. Such noise may result in small voids in the target image during virtual viewpoint synthesis utilizing depth maps. To mitigate this noise, research employs filtering to preprocess depth maps, effectively reducing distortions and voids within them. However, some conventional filtering methods, such as Gaussian or median filtering, risk the loss of edge information in images, leading to cracks and voids. Therefore, the retention of edge information is of paramount importance, as it allows for image smoothing while preserving salient features. Bilateral filters consist of two filter components: one associated with geometric spatial distance and the other with pixel differences. This design enables the bilateral filter to effectively reduce noise and voids while preserving edge in-

formation when processing depth maps, thereby enhancing the overall image processing outcome. The complete bilateral filter is as delineated in equation 4.

$$f(x,y) = \frac{\sum_{(i,j)\in S_{x,y}} w(i,j)g(i,j)}{\sum_{(i,j)\in S_{x,y}} w(i,j)} \tag{4}$$

Herein, $S_{x,y}$ represents the neighborhood of the central point at coordinates (x, y) with a dimension of $(2N + 1) * (2N + 1)$, N is an integer, $g(i, j)$ denotes the original image, and $f(x, y)$ signifies the image post-filtering. The weight coefficient $w(i, j)$ is composed of the product of two parts: the range filtering coefficient $w_r(i, j)$ and the spatial filtering coefficient $w_s(i, j)$, as illustrated in equations 5 and 6.

$$W_r(x,y) = \exp(-\frac{\left|g(i,j) - g(x,y)^2\right|}{2\sigma_r^2}) \tag{5}$$

$$W_s(i,j) = \exp(-\frac{(i-x)^2 + (j-y)^2}{2\sigma_S^2}) \tag{6}$$

Herein σ_S is the spatial proximity factor and σ_r is the luminance similarity factor. For a given image, its value is fixed.

2.2. RBM

Restricted Boltzmann Machines (RBMs) are a derivative of Boltzmann Machines (BMs), which are generative stochastic neural network models capable of learning and representing complex data distributions, thereby generating new samples similar to the training data [42]. However, their training process typically requires extensive computational resources and time. To address this issue, researchers introduced Restricted Boltzmann Machines (RBMs). RBMs substantially simplify the model's architecture and learning process by removing connections within the same layer. In the present study, RBMs are primarily employed for feature extraction from images. The energy function serves as the foundational descriptor of the overall structure and operations of the RBM, as delineated in equation 7.

$$\mathrm{E}(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{7}$$

Herein, v and h represent the states of the visible and hidden layers, respectively. a_i and b_i are the bias terms, and w_{ij} denotes the weights.

Layers of the RBM The Restricted Boltzmann Machine (RBM) is composed of two primary layers: the visible layer and the hidden layer [12]. The visible layer serves as the input interface of the RBM and is alternatively referred to as the observational layer. This layer comprises all observable data nodes, each of which represents a specific feature of the data. The principal function of the visible layer is to receive input data and relay this information to the hidden layer. By forming weighted connections with the hidden layer, the visible layer is capable of capturing the rudimentary structure and patterns inherent in the data.

The term "restricted" in Restricted Boltzmann Machine refers to the absence of connections between nodes within the same layer. This distinguishes RBMs from traditional Boltzmann Machines, where intralayer node connections are fully permitted. By eliminating these intralayer connections, RBMs substantially reduce both model complexity and computational requirements. This streamlined architecture not only simplifies the training process but also helps mitigate the risk of overfitting, making the model more scalable to large datasets. RBM achieves effective learning and representation of data distributions through the collaborative training of its visible and hidden layers, and by optimizing the weights between them. The advanced features and concepts captured by the hidden layer facilitate the model's understanding and reconstruction of complex data structures. This makes RBMs a potent tool for feature extraction and generative modeling, and they are widely employed in tasks related to unsupervised learning and deep learning.

The integration of a two-layer architecture—comprising the visible and hidden layers—and restricted connectivity design offers an efficient, flexible, and robust approach for learning and representing complex data distributions. The visible layer directly engages with the input data, while the hidden layer is responsible for capturing higher-order features and structures within the data. This design imbues RBMs with extensive applicability and superiority in handling a wide array of tasks in unsupervised learning and deep learning.

RBM activation function In RBMs, the probability distribution of the nodes in the hidden layer can be articulated using the Sigmoid activation function. Given the state v of the visible layer, the activation probability for the hidden layer nodes h_i is delineated as shown in equation 8.

$$P(h_i = 1|\, v) = \sigma(\sum_i v_i w_{ij} + b_j) \tag{8}$$

Herein, σ represents the Sigmoid activation function, v_i denotes the state in the visible layer nodes i, w_{ij} is the weight between the visible layer nodes i and the hidden layer nodes j, and b_j signifies the bias term for the hidden layer nodes j.

The Sigmoid activation function possesses the characteristic of mapping continuous inputs into a probability distribution. By employing the Sigmoid function, RBMs are capable of converting the linear combinations between the visible and hidden layers into probability values, representing the activation probabilities of the hidden layer nodes. These probability values are subsequently utilized in the sampling and inference processes, aiding the model in learning and generating the underlying data distribution. Through such an activation mechanism, RBMs effectively capture complex patterns and structures within the data, encoding and interpreting them probabilistically.

Loss Function RBMs employ Contrastive Divergence (CD) as their loss function. This study analyzes the limitations of the Contrastive Divergence method and introduces a feedback mechanism to optimize the training process.

The Contrastive Divergence loss function is described as shown in equation 9.

$$CD_n = \langle E(v,h)\rangle_{data} - \langle E(v',h')\rangle_{model} \tag{9}$$

2.3. Convolutional Neural Networks (CNN)

Layers Convolutional Neural Networks (CNNs) are a form of deep feed-forward neural networks [2], where n represents the number of Gibbs sampling steps, and $\langle \cdot \rangle$ symbolizes expectation. The training procedure primarily comprises three key phases: the forward pass, starting from the visible layer and calculating the probabilities in the hidden layer using the Sigmoid activation function, followed by sampling; the backward pass, commencing from the hidden layer and employing the same Sigmoid function to compute the reconstruction probabilities of the visible layer, also followed by sampling; and finally, weight updates, where gradients are computed based on the results of the forward and backward passes and applied using gradient descent methods.

However, two significant drawbacks exist in this training regimen. Firstly, Contrastive Divergence (CD) employs approximation techniques, which may not accurately capture the true data distribution. Secondly, CD is susceptible to local optima, leading to suboptimal training results. To mitigate these issues, this study introduces a feedback mechanism to optimize the training process. Specific improvements include: dynamic adjustment of the learning rate by monitoring loss variations to avoid local minima and accelerate convergence; the incorporation of regularization techniques, specifically L1 regularization, to enhance the model's generalization capabilities and prevent overfitting; and an increase in the number of Gibbs sampling steps to improve the accuracy of CD training and better approximate the true data distribution. This comprehensive approach ensures training accuracy and efficiency, effectively overcoming the limitations of traditional CD methods.

Artificial Neural Networks [7], particularly Convolutional Neural Networks (CNNs), are exceptionally suited for image recognition tasks [6]. The fundamental architecture of CNNs comprises convolutional layers, pooling layers, and fully connected layers.

The objective of the convolutional layer is to extract features from the image through convolution operations, as indicated in equation 10.

$$F_{i,j} = \sum_{m} \sum_{n} K_{m,n} \cdot I_{i+m,j+n} + b \tag{10}$$

Herein, $F_{i,j}$ represents an element of the feature map, $K_{m,n}$ denotes an element of the convolutional kernel, $I_{i+m,j+n}$ signifies an element of the input image, and b is the bias term.

This study employs max pooling, as illustrated in equation 11.

$$P_{i,j} = \max_{m,n} I_{i+m,j+n} \tag{11}$$

The fully connected layer transforms the output from the preceding layer into a one-dimensional array, which is then fed into the subsequent layer. Given N inputs and M outputs, the mathematical representation of the fully connected layer is as indicated in equation 12.

$$O_j = \sum_{i=1}^{N} W_{i,j} \cdot I_i + b_j \tag{12}$$

Herein, O_j represents the value of the output node, $W_{i,j}$ denotes an element of the weight matrix, I_i signifies the value of the input node, and b_j is the bias term.

Activation Functions This study employs the ReLU (Rectified Linear Unit) [3] nonlinear activation function to enhance the nonlinear expressive capability of the Convolutional Neural Network (CNN) model. The mathematical expression of the ReLU activation function is $f(x) = \max(0, x)$, such that its graphical representation appears as a piecewise linear curve. Specifically, the output is zero when the input is less than zero, and the output is equal to the input when it is greater than or equal to zero. Although the ReLU function may appear linear graphically, it is inherently nonlinear, enabling it to handle complex problems, especially within deep learning models. The computation of ReLU is remarkably simple and efficient, requiring only a check to determine whether the input is greater than zero, thereby reducing the demands on training time and computational resources. Additionally, given that the gradient of the ReLU function is 1 in the positive interval, it helps to mitigate the vanishing gradient problem, particularly in deep networks. In CNNs, the ReLU activation function is commonly applied following the convolutional and fully connected layers to increase the model's nonlinear expressive power, thereby enhancing predictive accuracy and generalization capabilities. Overall, due to its straightforward mathematical formulation, nonlinear properties, computational efficiency, and ability to alleviate the vanishing gradient problem, the ReLU activation function finds extensive application in the realm of deep learning. By utilizing ReLU in CNNs, this study aims to construct a more effective and robust model [15].

Loss Function The loss function serves as a metric for assessing the disparity between the model's predictions and the actual labels, and its selection is contingent upon the nature and objectives of the task at hand [14]. In this study, which focuses on classification tasks, Cross-Entropy Loss [34] is employed as the loss function. Cross-Entropy Loss is commonly used in multi-classification problems, and its mathematical expression is denoted as $Loss = -\sum_i y_i \log(\hat{y}_i)$, where y_i represents the i th element of the actual label, usually taking values of 0 or 1, and $\hat{y}_i$ is the i th element of the model's prediction, with a value range between 0 and 1. The role of Cross-Entropy Loss is to measure the divergence between the predicted probability distribution and the true probability distribution of the labels, making it suitable for scenarios where the model's output is a probability distribution. In the context of binary classification, it is equivalent to the log-likelihood loss in logistic regression and imposes a more severe penalty for confidently incorrect classifications. By judiciously selecting and applying Cross-Entropy Loss, this study aims to accomplish accurate classification tasks and effectively guide model training.

3. Method

3.1. Improved Neighborhood Interpolation Method for Small Hole-Filling

This study employs a neighborhood interpolation method for filling small holes, as cited in [27]. Given that most small holes are vertically contiguous—meaning holes exist both above and below the target hole—relying solely on the 8 adjacent pixels of the hole would result in underutilization of surrounding information. Therefore, we extend laterally to include two additional pixels, denoted as A_2 and E_2, as reference pixels for hole-filling, as proposed in [22]. Considering that the correlation between a pixel and the hole increases as they are closer to each other, the weights are assigned in such a manner that pixels

farther away from the hole have smaller weights, while those closer have larger weights, thereby ensuring effective hole-filling. The DIBR method, as per references [40] [36], generates left and right views of the image with different disparities, employing the same filling methodology. The relationship between the holes and the adjacent pixels used for filling is summarized in Table 1.

Table 1. Relationship Diagram between Holes and Adjacent Pixels Used for Filling

A_1	B_1	C_1	D_1	E_1
A_2	B_2	C_2	D_2	E_2
A_3	B_3	C_3	D_3	E_3

Assuming C_2 is a hole point, it is filled using the ten pixels B_1, C_1, D_1, B_2, D_2, B_3, C_3, D_3, A_2 and E_2. However, it is possible that some of the pixels used for filling may also be holes, and not all of these ten pixels may be empty. Therefore, it is essential to discuss the number of hole points within these pixels. Different filling methods are proposed based on the number of filled points within these pixels. If none of the ten pixels are holes, the filling formula is as indicated in equation 13.

$$C_2 = (B_1 + C_1 + D_1 + B_2 + D_2 + B_3 + C_3 + D_3)\,\omega_1 + (A_2 + E_2)\,\omega_2 \tag{13}$$

Where ω_1 and ω_2 are the filling parameters. Let m denote the number of holes in the 8-pixel neighborhood of C_2, and n denote the number of holes in the pixels A_2 and E_2. The number of hole points may have some influence on the filling outcome. If m is less than 8 and n is less than or equal to 2, the hole-filling formula is as indicated in equation 14.

$$C_2 = \frac{8\omega_1}{8-m}\,(B_1 + C_1 + D_1 + B_2 + D_2 + B_3 + C_3 + D_3) + \frac{2\omega_2}{2-n}\,(A_2 + E_2) \tag{14}$$

If $m = 8$ and $n < 2$, the hole-filling formula is as indicated in equation 15.

$$C_2 = \frac{10\omega_2}{2-n}\,(A_2 + E_2) \tag{15}$$

If $m < 8$ and $n = 2$, the hole-filling formula is as indicated in equation 16.

$$C_2 = \frac{10\omega_1}{8-m}\,(B_1 + C_1 + D_1 + B_2 + D_2 + B_3 + C_3 + D_3) \tag{16}$$

Through the methods described above, holes within the image can be effectively filled by utilizing the adjacent pixels, thereby resolving issues related to the presence of holes in the image. However, if the area of the hole is large and there is insufficient pixel information surrounding it, such holes are considered unrepairable large holes. Edge portions generally do not contain critical information; thus, they can be filled using corresponding positions from the original image.

3.2. RBM+CNN for Image Recognition

This study integrates Restricted Boltzmann Machines (RBM) with Convolutional Neural Networks (CNN) algorithms for training on input images. The RBM is responsible for extracting crucial features from the raw images, while the CNN takes on the task of transforming these features into specific recognition outcomes. The architecture of the integrated network is illustrated in Figure 1.

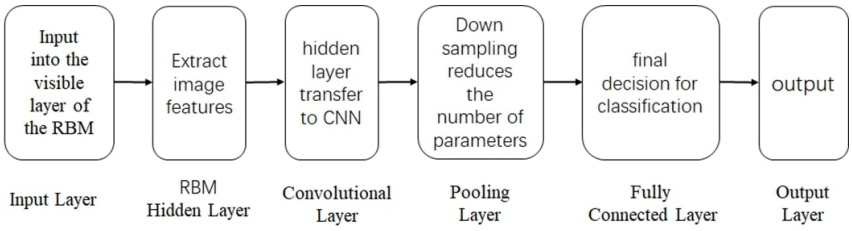


Fig. 1. Conversion effect

As shown in Figure 1, the RBM and CNN integrated network in this study comprises the following key components:

1) Input Layer: Image data are fed into the visible layer of the RBM, with each pixel corresponding to a visible unit. This layer serves to present the raw image data to the network.
2) RBM Hidden Layer: The hidden layer of the RBM is responsible for extracting features from the images. Each hidden unit acts as a feature detector and is not interconnected with other hidden units, only sharing bidirectional connections with the visible units. This design enhances computational efficiency and feature extraction capabilities.
3) Convolutional Layer: Transfers from the RBM's hidden layer to the CNN's convolutional layer for further spatial feature extraction. Each neuron in the convolutional layer is connected to a localized region in the preceding layer and utilizes a shared weight matrix (i.e., convolutional kernel) for computation. This helps capture spatial dependencies within the image.
4) Pooling Layer: Performs downsampling to reduce the number of parameters. Each neuron in the pooling layer connects to a small region in the preceding layer and employs a max-pooling function for computation. This layer minimizes computational demands while retaining feature significance.
5) Fully Connected Layer: Serves as the final decision-making layer for classification. Each neuron in this layer is connected to all neurons in the preceding layer and employs a ReLU activation function for computation. This ensures the network can perform complex nonlinear mappings.
6) Output Layer: Presents the final classification outcome. Each neuron in the output layer corresponds to a specific class and employs a softmax activation function for computation. This ensures that the output is represented as a probability distribution, with each element indicating the predicted probability of the corresponding class.

Overall, the integrated RBM and CNN network structure leverages the strengths of both models, offering a hierarchical approach for image feature extraction and classification, and demonstrating considerable flexibility and performance [16].

This study begins with an analysis of the shortcomings of the Contrastive Divergence method using a Restricted Boltzmann Machine (a sub-model of the Boltzmann Machine) and introduces a feedback mechanism, as cited in [41]. Given the high computational cost of direct gradient use, stochastic sampling is employed for gradient estimation. A subset X' of the original X set is obtained to ensure that their distributions are approximately consistent. The gradient parameters (e, f, v) indicated here require continuous updates in the Contrastive Divergence method of the RBM algorithm. The learning process in the Boltzmann Machine $(\mathrm{e}, \mathrm{f}, \mathrm{v})$ can also be represented as an updating process. Therefore, RBM (e, f, v) can likewise be depicted as an iterative process.

The training and learning process of the model relies on samples obtained through sampling, which directly influence the effectiveness of the training classification. Upon completion of sampling, a feedback mechanism [4] can be introduced to better optimize the parameters. The RBM model is considered in terms of network input (X) and network output (Y), and sampling is performed for both inputs and outputs. The sampling process for X. and Y is defined as sample_positive. It calculates the similarity between Y, X, and X', and the sampled output of Y given X is defined as sample_negative. To obtain optimal parameters, multiple samples are required. The best parameters are identified when the sample distance is reduced or approaches the first node. When the distance changes, the bias could have optimal parameters, and sampling terminates. The number of iterations (C) is arbitrary and depends on the treatment of the sampled examples. For any training sample Xi, the current position of the sample is represented as $X\,(0) = Xi$. The Contrastive Divergence method based on the feedback mechanism is as indicated in equation 17.

$$\Delta W_{ji} = Or_j b_{ji} \tag{17}$$

In the convolutional computations, r_j represents the adjustment coefficient of the sampling element j in the overall connection gradient bias. b_{ji} represents the amount of data acquired by element w_{ji} in a given direction. The corresponding learning rate is denoted by O. The adjustment amount for the i th input weight of the sampling element j can be represented as indicated in equation 18.

$$\Delta P^{l}_{fmap-i} = \sum_{j-1}^{N} convn\left(\Delta P^{l+1}_{fmap-j}, inv\left(K^{l+1}_{ij}\right)\right) \tag{18}$$

In the convolutional computation process, K denotes the convolutional kernel matrix. The feature changes in the convolutional layer must be acquired through backtracking methods. If I represents the sampling layer, then I signifies the convolutional connections. Node i, corresponding to the non-wired layer, has N convolutional rollbacks and can be viewed as the sum of N elements. During hierarchical transmission, the forward convolutional matrix is $PI + 1$ with dimensions $\mathrm{Size}\left(\mathrm{P^I}, 1\right) - \mathrm{Size}\left(\mathrm{K}, 1\right) + 1$. The backward convolutional matrix is PiPi, with corresponding dimensions described as $\mathrm{Size}\left(\mathrm{P^{I+1}}, 1\right) + \mathrm{Size}\left(\mathrm{K}, 1\right) - 1$. Convolutional kernel flipping is employed, and dur-

ing this flipping process, the bias matrix of the ith feature map in the lth layer of M-dimensional convolutional operation is as indicated in equation 19.

$$\Delta K_{ij}^{l} = convn\Delta P_{fmap-j}^{l}, inv\left(P_{fmap-i}^{l-1}\right) \tag{19}$$

The output bias (ΔP_{fmap-i}^{l-1}) is derived from the flipped convolutional kernel $inv\left(K_{ij}^{l+1}\right)$ and the input bias matrix (ΔP_{fmap-i}^{l-1}). Using backtracking methods, the bias of the convolutional kernel matrix can be further obtained from the flipped input bias matrix and output bias matrix, as indicated in equation 20.

$$K_{ij}^{(t)} = K_{ij}^{(t-1)} - \Delta K_{ij} \tag{20}$$

When progressing through layers from $I = 2$ forward, the kernel state at a given time $t - 1$ is obtained by applying the convolutional kernel bias to the kernel state ${K_{ij}}^{(t-1)}$ at the preceding time t, as indicated in equation 21.

$$\Delta W_{ji}^{(t)} = W_{ij}^{(t-1)} - \Delta W_{ij} \tag{21}$$

Through the derived kernel bias function, one can further extend to the states of the ith feature volume matrix at the first layer for different time instances, as indicated in equation 22.

$$\Delta W_{ji}^{(t)} = Or_j b_{ji} + \beta \Delta W_{ji}^{(t-1)} \tag{22}$$

Here, a momentum term is introduced to describe the state of the corresponding feature volume bias matrix, aiming to accelerate convergence. β represents the momentum coefficient, with a value range of [0, 1]. When $\beta = 0$, it is capable of representing the state of the corresponding feature volume bias matrix at time t.

3.3. Image Transformation Results

In the experiment, we selected an indoor corridor as the scene and achieved the transformation from 2D to 3D through different stages of image processing. As depicted in Figure 2, the four stages are as follows: Fig0.2(a) The original 2D grayscale image: This serves as the starting point of the experiment and showcases the basic appearance and shape of the corridor. Fig0.2(b) Foreground and background segmentation: At this stage, the image is segmented into foreground and background, with white representing the foreground and black representing the background. This helps to emphasize the main features of the corridor. Fig0.2(c) Pre-processed depth map: The image undergoes pre-processing to generate a depth map, which contains spatial depth information of the corridor and lays the groundwork for 3D reconstruction. Fig0.2(d) 2D-to-3D reconstruction: Utilizing the previous processing steps, the 2D image is successfully transformed into a 3D effect image, revealing the three-dimensional appearance of the corridor.

Through these four sequential stages, we are able to start with the original 2D grayscale image, progressively unveil the structure of the corridor, and ultimately achieve a 2D-to-3D transformation. Each stage provides necessary information and a foundation for the subsequent stage, forming a coherent and logical processing workflow.

(a) The original 2D grayscale image (b) Foreground and background segmentation (c) Pre-processed depth map (d) 2D-to-3D reconstruction

Fig. 2. Effects of the 2D-to-3D Processing Workflow

4. Result and Discussion

4.1. Experimental Environment

The experiments are conducted using a Dell Precision T7960 graphics workstation, the main parameters of which include the use of an Intel Xeon W7-3465X 28-core processor with a frequency of 2.5 GHz, equipped with a 1 TB or 2 TB SSD and 16 TB mechanical hard disk storage solution, as well as a high-end four NVIDIA RTX 6000 Ada graphics cards. The experiments runs on a Linux operating system, the datasets utilized include the GlobalMLBuildingFootprints open-source dataset, architectural images from ADE20K, and the WHU Building Dataset for building detection.

The model was trained on a curated subset of the GlobalMLBuildingFootprints open-source dataset and validated on architectural images from ADE20K as well as the WHU Building Dataset. In total, 1,317,000 architectural images were used, with 790,000 images in the training set, 263,000 images in the test set, and 261,800 images in the validation set.

4.2. Comparative Experiment on Hole-filling

To enhance the accuracy of image transformation, a hole-filling algorithm was introduced for image pre-processing, specifically targeting the repair of small holes at the edges of the depth map [8]. The evaluation metrics used in this experiment are Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE), defined by equations 23, 24, respectively.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I^2}{\sqrt{MSE}}\right) \tag{23}$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|\left|I\left(i,j\right) - K\left(i,j\right)\right|\right|^2 \tag{24}$$

In these equations, $I\left(i,j\right)$ and $K\left(i,j\right)$ represent the pixel values at corresponding coordinates. m and n denote the height and width of the image, respectively. MAX_I is the maximum numerical value representing the color of a pixel in the image.

As well as the structural similarity index (SSIM) and the mean absolute error (MAE), which are defined by the formulas shown in equations 25 and 26:

$$SSIM(P_1, P_2) = \frac{(2\mu_{P_1}\mu_{P_2} + C_1)(2\sigma_{P_1P_2} + C_2)}{(\mu_{P_1}^2 + \mu_{P_2}^2 + C_1)(\sigma_{P_1}^2 + \sigma_{P_2}^2 + C_2)} \tag{25}$$

Where P_1, P_2 represents the original and enhanced images respectively; μ_{P_1}, μ_{P_2} represents the mean of the images P_1 and P_2 respectively; $\sigma_{P_1}^2, \sigma_{P_2}^2$ represents the variance of the images P_1 and P_2 respectively; $\sigma_{P_1P_2}$ represents the covariance of the P_1 and P_2; and C_1, C_2 represents the small constant used to stabilize the denominator.

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|P_{1ij} - P_{2ij}| \tag{26}$$

Where P_{1ij}, P_{1ij} represent the pixel values of the original and enhanced images at the location (i, j) respectively; m and n are the height and width of the image respectively. The main objective of this study is to measure the quality of the enhanced image by calculating the difference between the gray values of the corresponding pixel points of the enhanced image and the original image by using a statistical method.Both PSNR and MSE measure the good quality of the image by calculating the global magnitude of the pixel error between the enhanced image and the original image. Specifically, a larger value of PSNR indicates that the distortion between the enhanced image and the original image is smaller and the image quality is better. On the contrary, a smaller value of MSE indicates better image quality.SSIM considers the brightness, contrast and structure of an image, and a value closer to 1 indicates better image quality. This metric is useful for evaluating the visual quality of an image during 2D to 3D conversion, and can help determine whether the converted image visually maintains structural features similar to those of the original image.MAE measures the average error at the pixel level, with a smaller value indicating a smaller error, and this metric is valid for evaluating the overall pixel-level error in 2D to 3D conversion.

To verify the effectiveness of this paper's algorithm, by comparing with SqueezeNet algorithm , Xception algorithm and MobileNet algorithm , as shown in Table 4.2. In terms of PSNR, the algorithm in this paper is 38.67 dB, which is an improvement of 2.44% compared with SqueezeNet algorithm, 1.55% compared with Xception algorithm, and 1.31% compared with MobileNet algorithm. Since PSNR is a measure of the quality aspect of image reconstruction, its improvement means that the error of image reconstruction is reduced and the image quality is improved. Although the improvement effect is not obvious, it still indicates a certain progress, which is due to the optimization of the algorithm in image feature extraction and reconstruction model in this paper. In terms of MSE, the mean square error of this algorithm is 8.92, which is 27.71% lower than the SqueezeNet algorithm, 16.24% lower than the Xception algorithm, and 2.19% lower than the MobileNet algorithm. Since MSE is a measure of the difference between images, the lower its value indicates that the error between the reconstructed image and the original image is smaller, the effect of this paper's algorithm is more obvious compared with the previous two algorithms, which should be attributed to the more accurate model training and more appropriate loss function selection. Although the effect is almost the same with MobileNet, there is also a slight improvement. In terms of SSIM, the structural similarity index of this paper's algorithm is 0.9277, which is improved by 7.43% compared with SqueezeNet algorithm, 5.21% compared with Xception algorithm, and 3.72% compared with MobileNet algorithm. Since SSIM is a metric in evaluating the structural fidelity of

an image, its improvement means that the image reconstruction process better preserves the structural features of the original image. Although the improvement is not extremely significant, it shows that the algorithm in this paper has made some progress in this area. This effect is due to the optimization of the algorithm in using more efficient activation functions and more efficient data enhancement techniques. The more efficient activation function may help the network to better capture and convey important features in the image, while the efficient data enhancement technique may enhance the robustness of the model to various changes and perturbations, and thus perform better in preserving the image structure. In terms of MAE, the mean square error of the algorithm in this paper is 3.29, which is 10.35% lower compared to SqueezeNet algorithm, 11.56% lower compared to Xception algorithm and 3.52% lower compared to MobileNet algorithm. Since the MAE is a measure of the difference between images, the lower its value indicates that the error between the reconstructed image and the original image is smaller, the effect of this paper's algorithm is more obvious compared with the previous two algorithms, which should be attributed to the more optimized network architecture of this paper's algorithm and optimization of the specific dataset. Although the effect of this algorithm is similar to that of MobileNet, there is a slight improvement. This suggests that the algorithm in this paper may be more refined in terms of network design, which can handle specific types of image data more effectively, and the optimization for specific datasets may help the algorithm better adapt to and handle these data, and thus achieve better results in terms of MAE metrics.

Table 2. Comparative Experimental Results on Parameters for Hole-filling

Algorithms	PSNR/DB	MSE	SSIM	MAE
SqueezeNet	37.75	12.34	0.8635	3.67
Xception	38.08	10.65	0.8818	3.72
MobileNet	38.17	9.12	0.8944	3.41
Algorithms in this article	38.67	8.92	0.9277	3.29

4.3. Comparison Experiments

Comparison of quantitative results of measurements In this study, the performance of SqueezeNet algorithm, Xception algorithm, MobileNet algorithm and the algorithm proposed in this paper are compared on WHU Building Dataset dataset. To ensure the fairness of the comparison, all algorithms are tested on the same training, test and validation sets. Specifically, each training contains 400 samples and the training process of each algorithm is capped at a maximum of 1000 iterations. The comparison results of the four algorithms are shown in Table 3.

The comparison in terms of number of iterations is discussed first. The number of iterations of this paper's algorithm needs more compared to SqueezeNet algorithm and Xception algorithm, this is because the model structure and optimization strategy of this paper's algorithm needs more iterations to reach convergence. However, this paper's algorithm requires fewer iterations compared to MobileNet algorithm, which indicates that

Table 3. Comparative Experimental Results of Three Algorithmic Models

Algorithms	Number of iterations	Training set loss function	Training set accuracy	Validation set loss function	Validation set accuracy	Measurement accuracy
SqueezeNet	790	0.0231	0.9891	0.2100	0.9579	95.32
Xception	650	0.0229	0.9913	0.0056	0.9714	98.59
MobileNet	920	0.0089	0.9985	0.0039	0.9932	99.21
Algorithms in this article	830	0.0214	0.9946	0.0041	1.0000	99.44

this paper's algorithm outperforms MobileNet algorithm in terms of optimization strategy. Next, the performance of the three aspects of training set loss function, training set accuracy and validation set loss function are analyzed. The algorithm in this paper outperforms SqueezeNet and Xception algorithms in these aspects, which is mainly attributed to the effective loss function optimization, accurate model parameter tuning, excellent feature extraction, and better model generalization ability of the algorithm in this paper. In contrast, the performance of this paper's algorithm in these three aspects is slightly lower compared to the MobileNet algorithm, indicating that the MobileNet algorithm has a slight advantage in these aspects. Finally, consider the performance in both validation set accuracy and measurement accuracy. This paper's algorithm outperforms the other three algorithms in both aspects, which indicates that this paper's algorithm is more accurate and effective in processing validation set data, and has more robustness in practical applications.

In summary, the algorithm in this paper performs well in several aspects, especially in validation set accuracy and measurement accuracy, showing its advantages in practical applications. Although it is slightly inferior to MobileNet in some aspects, overall, the algorithm in this paper demonstrates its effectiveness and robustness in image processing tasks.

Table 4. Quantitative Comparison Experiment Results for 2D Images Across Different Algorithmic Models

Algorithms	Check Accuracy Rate	IoU	Recall Rate	F1 Score	Error rate
SqueezeNet	0.912	0.933	0.954	0.931	0.048
Xception	0.927	0.938	0.96	0.947	0.041
MobileNet	0.958	0.945	0.962	0.962	0.042
Algorithms in this article	0.97	0.948	0.964	0.967	0.041

The experiments in this study involved inputting 8,000 2D images taken in the field into different models for recognition and analyzing them in depth in terms of five aspects: checking accuracy, IoU (intersection and concatenation ratio), recall, F1 score (reconciled mean), and error rate. In terms of detection rate, the detection rate of SqueezeNet algorithm is only 0.912, and after observing the pictures with prediction errors in detail, it is found that the possibility of prediction errors is increased in special areas such as high reflective areas formed by little difference in color, strong external light, or broken concrete floors. In comparison, the algorithm in this paper improves the checking accuracy with

Xception algorithm and MobileNet algorithm by 4.64% and 1.25%, respectively, showing better robustness and discriminative ability for these complex situations. In terms of IoU (intersection-to-union ratio), this paper's algorithm shows a slight improvement compared to the other three algorithms. This result proves that all four algorithms have better ability in locating the target object boundary, while this paper's algorithm can locate the target object boundary more precisely and capture the target morphology more accurately. In terms of recall, this paper's algorithm also has a slight improvement compared with the other three algorithms, this result shows that the four algorithms have comparable abilities in capturing positive samples, and also highlights the slight advantage of this paper's algorithm in recalling positive samples. In terms of F1 scores, this paper's algorithm improves by 3.87%, 2.11% and 0.52% compared to the SqueezeNet algorithm, the Xception algorithm and the MobileNet algorithm, respectively. Since the F1 score is the reconciled average of the detection and recall rates, this improvement reflects the simultaneous optimization of this paper's algorithm in terms of detection and recall rates. In terms of error rate, this paper's algorithm has the lowest error rate with Xception algorithm, which is 14.58% and 2.38% lower than SqueezeNet algorithm and MobileNet algorithm, respectively. This result proves the superiority of this paper's algorithm in providing more accurate classification decisions and more robust feature extraction.

Combining the experimental results in Tables 3 and 4, this paper's algorithm performs better in several key performance metrics, especially in validation set accuracy, measurement accuracy, checking accuracy, and F1 scores than the other compared algorithms. These results show that this paper's algorithm has advantages in model training, generalization ability and accuracy in practical applications. The subtle advantages of this paper's algorithm in other metrics also reflect its overall algorithmic robustness and efficiency. These performance improvements are attributed to the innovation of algorithm design, refinement of feature engineering, and optimization of the training process.

Table 5. Experimental results comparing 2D to 3D running time and efficiency of different algorithm models

Algorithms	Total Time(s)	FPS	FLOPS
SqueezeNet	59.22	51	53.97
Xception	56.37	55	57.81
MobileNet	51.93	62	67.38
Algorithms in this article	36.11	87	87.27

Runtime & Efficiency Comparison In this study, the total time taken by four different algorithms to convert 2000 2D architectural images into 3D images with the same training, test, and validation sets is compared. The study observes the conversion duration (s), frames per second (FPS) of processed images and peak speed per second (FLOPS). In terms of conversion duration, the conversion time of this paper's algorithm is 36.11s, which is 39.02% lower than the running time of SqueezeNet algorithm, 35.94% lower than the running time of Xception algorithm, and 28.06% lower than the running time of

MobileNet algorithm; this result is due to the fact that the algorithm of this paper has improved the traditional CNN This result is attributed to the improvement of the traditional CNN network structure and the precise setting of the weight initialization value of this algorithm. In terms of FPS, the algorithm in this paper improves significantly compared with the other three algorithms, which is attributed to the improvements in image preprocessing, feature extraction, reduction of redundant computation, optimization of memory access, and enhancement of parallel computation capability. In terms of FLOPS, the algorithm in this paper has a significant advantage over the other three algorithms in terms of processing speed per second, which is attributed to the optimization of the computational density of the algorithm, which reduces unnecessary floating-point operations, improves the parallelism of the algorithm, and optimizes the computationally intensive part of the code. In summary, the significant improvement in conversion time, FPS and FLOPS of this paper's algorithm reflects the optimization and improvement of the algorithm in several key aspects, such as computational efficiency, data preprocessing, feature extraction, parallel computing, resource management and code implementation, which proves that this paper's algorithm can efficiently and accurately implement the 2D to 3D conversion technology.

Ablation Study The present study introduces a novel image reconstruction algorithm that amalgamates the advantages of both Convolutional Neural Networks (CNN) and Restricted Boltzmann Machines (RBM). The objective is to achieve more precise image reconstruction. To validate the superiority of the proposed algorithm over traditional CNN and RBM techniques in terms of image reconstruction quality, we have designed a comparative experiment.

Table 6. 2D picture quantitative ablation comparison experiment results

Algorithms	Check Accuracy Rate	IoU	Recall Rate	F1 Score	Error rate
CNN	0.665	0.701	0.782	0.719	0.081
RBM	0.73	0.715	0.759	0.745	0.076
Algorithms in this article	0.97	0.948	0.964	0.967	0.041

The experiment involved the input of 8,000 on-site captured 2D images into three different algorithms for identification. The results, as shown in Table 6, indicate that the algorithm proposed in this study outperforms traditional CNN and RBM models across five key metrics, substantiating its effectiveness. This superiority can be elucidated from several perspectives:

Architectural Advantage: The proposed algorithm integrates the convolutional layers of CNN with the generative model features of RBM. This amalgamation leverages the strengths of both, resulting in higher precision and robustness in image reconstruction.

Efficient Training: By adopting more precise weight initialization, the training process for our algorithm is both efficient and stable. This enhances the model's ability to fit the

training data more effectively, thereby improving its 3D reconstruction performance on the on-site captured 2D images.

Function Selection: The algorithm employs loss and activation functions that are particularly suited to this specific task. The appropriate choice of loss function better guides model learning, while the optimal activation function enhances the model's non-linear representational capability.

In summary, the proposed algorithm's superiority over traditional CNN and RBM models is primarily attributable to its comprehensive optimization in structural design, training strategy, and function selection. These collectively contribute to a significant advantage in 3D reconstruction tasks for on-site captured 2D images.

Table 7. Ablation Study on Run Time and Efficiency for 2D-to-3D Conversion

Algorithms	Total Time(s)	FPS	FLOPS
CNN	124.89	31	27.92
RBM	119.54	34	31.31
Algorithms in this article	36.11	87	87.27

Under a unified framework of identical training, testing, and validation sets, this study meticulously contrasts the efficacy of three disparate algorithms—our proposed algorithm, traditional Convolutional Neural Networks (CNN), and Restricted Boltzmann Machines (RBM)—in the task of transforming 2,000 2D architectural images into 3D models, as detailed in Table 7. Our observations indicate that the proposed algorithm significantly outperforms the traditional counterparts across various metrics, including total conversion time, frames processed per second (Fps), and peak floating-point operations per second (FLOPs). Firstly, this substantial advantage can be attributed to the architectural refinements incorporated into the proposed algorithm. By synergistically leveraging the convolutional attributes of CNNs and the generative capabilities of RBMs, the algorithm enhances both the efficiency and accuracy of image conversion tasks. Secondly, the marked elevation in computational efficiency underscores the algorithm's prowess in diminishing computational complexity and overall operational time. This not only expedites the training phase but also augments the model's performance in handling intricate image data. Finally, the flexibility and precision of the proposed algorithm are also manifest. Optimized for task-specific needs, the algorithm minimizes unnecessary floating-point calculations and redundant computations. As a result, it significantly amplifies operational speed, establishing new benchmarks in both frames processed per second and peak operational speed.

In summary, the comprehensive optimization of the proposed algorithm in terms of architecture, efficiency, and flexibility renders it exceptionally proficient in 2D-to-3D image conversion tasks. It substantially transcends traditional CNN and RBM algorithms, thereby demonstrating its potent potential and applicability.

Comparative Performance of 2D-to-3D Image Reconstruction Across Algorithms As shown in Figure 3, the results of three different algorithms are demonstrated on the

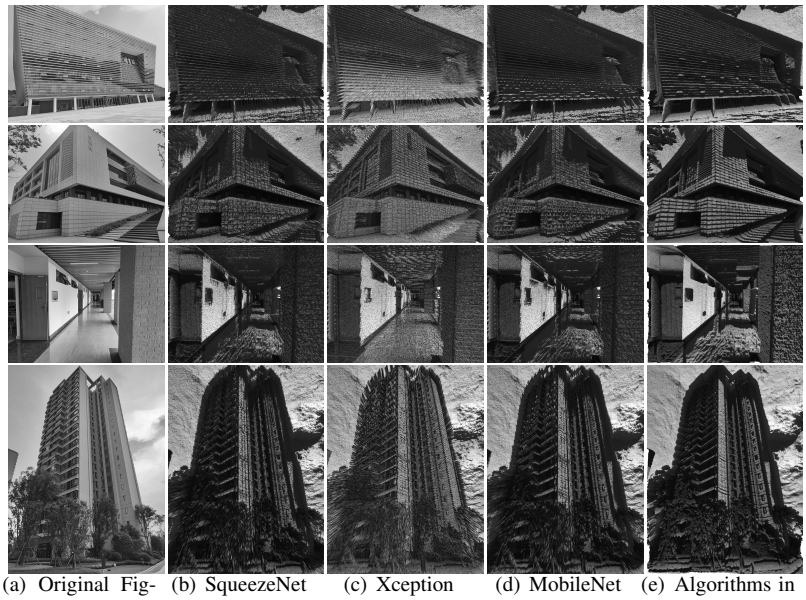
task of 2D to 3D image reconstruction on library image, academic building image, corridor image and residential building image. Image.3(a) is the original image, and images Image.3(c) to Image.3(e) show the processing results of SqueezeNet algorithm, Xception algorithm, MobileNet algorithm, and this paper's algorithm, respectively. SqueezeNet algorithm, as an earlier deep learning model, has a relatively simple structure and cannot capture all the subtle features of the image, so the detail features are weak. The Xception algorithm, with a more complex structure and more parameters, is better than the SqueezeNet algorithm but still has room for improvement. The MobileNet algorithm, an efficient neural network model, achieves minimizing the demand for computational resources while maintaining good performance by employing depth-separable convolutions and specific network structural adjustments, but it is not effective in dealing with extremely complex or fine-grained tasks, there is still room for improving its performance.

In contrast, our proposed algorithm excels in 2D-to-3D image reconstruction, manifesting a marked enhancement in detail fidelity. This superiority can be attributed to three key facets: (1)Task-Specific Optimization: The algorithm has been fine-tuned for specific types of images and content, enabling it to better capture intricate details. (2)Loss Function and Optimization Strategy: The selection of a loss function and optimization techniques particularly tailored for 2D-to-3D conversion tasks further augments the quality of image reconstruction. (3)Precise Weight Initialization and Efficient Training: The algorithm employs a highly accurate weight initialization process and an optimized training methodology, contributing to efficient and stable training.

In summation, the proposed algorithm stands as the most proficient in reconstructing 3D images from 2D precursors, owing to its comprehensive optimization across task-specific requirements, loss function selection, and training procedures. Its exemplary performance underscores its robustness and applicability in complex image transformation tasks.

In this study, we incorporate various performance indicators such as the TOP-5 [18] error rate, the number of convolutional layers, the size of the convolutional kernels, and the number of fully connected layers to compare the efficacy of different algorithmic models. The TOP-5 error rate serves as a critical evaluation metric that considers a prediction as correct if the true label is among the top five probabilities in the final output vector. A lower TOP-5 error rate is indicative of better model performance. Taking the library image reconstruction effect as an example, the Top-5 error rates of different algorithm models can be observed in Table 8, and the four algorithms show different performances. the SqueezeNet algorithm has a relatively high error rate of 6.7%, which is not able to adequately capture the complex features of the image due to its relatively simple structure. the Xception algorithm and MobileNet algorithm have reduced the error rates to 5.9% and 4.9%, which is a significant improvement compared to the first algorithm, but there is still room for improvement. Proposed Algorithm: Demonstrates a significantly reduced TOP-5 error rate of 3.1%, emphasizing the algorithm's superiority in image reconstruction tasks.

The decline in error rates can be attributed to increased algorithmic complexity and task-specific optimizations. The proposed algorithm, through its architectural refinements and innovative methods, achieves a more accurate image reconstruction. Its substantially lower error rate compared to the other algorithms corroborates its excellence in image reconstruction tasks.



(a) Original Figure (b) SqueezeNet (c) Xception (d) MobileNet (e) Algorithms in this article

Fig. 3. Algorithm effect diagram comparison

In the subsequent section, we delve into the significance of convolutional layers. Images are composed of pixels, which may contain noise. Convolutional layers serve the purpose of extracting features from these noisy pixels. As the number of convolutional layers increases, these rudimentary features combine to form higher-level patterns. Consequently, a deeper convolutional network is capable of capturing more abstract and comprehensive features. With 151 convolutional layers, our proposed algorithm excels in exhaustive detail extraction, contributing to the increased accuracy of the generated 3D images.

Furthermore, to reduce computational and parameter overhead while accelerating model training, smaller convolutional kernels are beneficial. They allow for the inclusion of additional convolutional layers, thereby enhancing the network's nonlinear capabilities. On another note, a single fully connected layer is insufficient for condensing feature data to a degree where accurate judgments can be made. The presence of two layers in our algorithm ensures more concentrated feature data.

In summary, when considering multiple evaluation metrics such as the TOP-5 error rate, the number of convolutional layers, the size of the convolutional kernels, and the number of fully connected layers, our proposed algorithm outperforms other models in terms of overall efficacy. The architecture of our model is well-balanced in terms of convolutional kernel size and the number and size of fully connected layers, making it a more potent solution for the task at hand.

Table 8. Experimental results of performance comparison of different algorithms for library image reconstruction

Algorithms	TOP-5 Error Rate	Number Of Convol -utional Layers	Convol -utional Kernel Size	Number Of Fully Connected Layers
SqueezeNet	6.7%	115	7,1,3,5	2
Xception	5.9%	118	7,1,3,5	2
MobileNet	4.9%	132	11,7,5,3	2
Algorithms in this article	3.1%	151	7,1,3,5	2

5. Conclusion

A new model for architectural image processing is developed in this study. This model combines a Restricted Boltzmann Machine (RBM) and a Convolutional Neural Network (CNN) specifically designed to convert two-dimensional (2D) architectural images into three-dimensional (3D) models. The input building images are first preprocessed to ensure their suitability. Then, a bilateral filtering technique is used to convert the image into a depth map, laying the foundation for subsequent 3D reconstruction. Next, small voids in the image are repaired by a domain difference algorithm, thus improving the integrity and quality of the image. The last key step is to input the processed 2D image into the research-designed model. The model incorporates the features of RBM and CNN to effectively realize the conversion from 2D image to 3D model. After thorough testing, the study found that the model achieved 97% in terms of checking accuracy and significantly outperformed other existing algorithms in terms of runtime and efficiency.

However, it is crucial to acknowledge the limitations of our current algorithm. For instance, the likelihood of prediction errors increases when dealing with areas of subtle color variations, strong external lighting causing high reflectivity, or surfaces with damage mimicking water stains. This suggests that further optimizations are required for our algorithm to adeptly handle these challenging scenarios. Looking ahead, we anticipate continual refinement of our algorithm in two primary directions:

Enhanced Models and Techniques: We aim to incorporate more advanced deep learning models and techniques to improve both the accuracy and robustness of our algorithm. Diverse Applications: We will explore the utility of our algorithm in processing other types of architectural images and even in other image processing tasks, with the goal of widespread application and dissemination.

By addressing these aspects, we aim to extend the applicability and fortify the resilience of our algorithm, thereby contributing to its broader adoption in the domain of image processing.

Acknowledgments. Funding project: 2022 Sichuan Province Science and Technology Innovation Project in the field of housing and urban-rural construction, "Research and Application of Intelligent Detection of Underground Space Leakage Based on Computer Vision" (NO: SCJSKJ2022-05).

References

1. 3D scene description and recording mechanis: A comparative study of various 3D create tools. Journal of System and Management Sciences
2. Deep learning to detect image forgery based on image classification. Journal of System and Management Sciences
3. Evaluation of algorithmic metrics with a focus on server Cyber-Risks. Evaluation of Algorithmic Metrics with A Focus on Server Cyber-Risks. Journal of System and Management Sciences
4. Factors affecting students' intention to use mobile learning at universities: an empirical study. Journal of System and Management Sciences
5. Identify faults in road structure zones with deep learning. Journal of System and Management Sciences
6. Inventory policy for electric vehicle battery swapping stations in beijing. Journal of System and Management Sciences
7. rediction of Vibration in the Discharge Ring of a River Type Hydroelectric Power Plant with Bulb Turbine Using Artificial Neural Networks and Support Vector Machine
8. Root cause analysis for it incident using artificial neural network (ANN). Journal of System and Management Sciences
9. 3D reconstruction using SfM with sequence images in hadoop. Journal of System and Management Sciences (2019)
10. Image and graph restoration dependent on generative adversarial network algorithm. Tehnicki Vjesnik 28(6) (2021)
11. Image completion based on edge prediction and improved generator. Tehnicki Vjesnik 28(5) (2021)
12. Neural network driven automated guided vehicle platform development for industry 4.0 environment. Tehnicki Vjesnik 28(6) (2021)
13. An innovative photogrammetric system for 3D digitization of dental models. Tehnicki Vjesnik 29(5) (2022)
14. Motor imagery EEG recognition using deep generative adversarial network with EMD for BCI applications. Tehnicki Vjesnik 29(1) (2022)
15. Corporate online learning as shared service based on ADDIE model. Journal of System and Management Sciences 14(4) (2023)
16. Machine learning approaches for precision medicine: A review. Journal of System and Management Sciences 14(4) (2023)
17. Aggarwal, H.K., Mani, M.P., Jacob, M.: MoDL: Model-based deep learning architecture for inverse problems. IEEE Trans. Med. Imaging 38(2), 394–405 (2019)
18. Ayachi, R., Afif, M., Said, Y., Atri, M.: Strided convolution instead of max pooling for memory efficiency of convolutional neural networks. In: Smart Innovation, Systems and Technologies, pp. 234–243. Springer International Publishing, Cham (2020)
19. Chen, B.H., Tseng, Y.S., Yin, J.L.: Gaussian-Adaptive bilateral filter. IEEE Signal Process. Lett. 27, 1670–1674 (2020)
20. Chen, Y., Hu, S., Mao, H., Deng, W., Gao, X.: Application of the best evacuation model of deep learning in the design of public structures. Image Vis. Comput. 102(103975), 103975 (2020)
21. Chi, X., Meng, Q., Wu, Q.: A laser data compensation algorithm based on indoor depth map enhancement. J]. Electronics 12(12) (2023)
22. Cho, S.I., Kang, S.J.: Temporal incoherence-free video retargeting using foreground aware extrapolation. IEEE Trans. Image Process. 29, 4848–4861 (2020)
23. Eslami, S.M.A., Heess, N., Williams, C.K.I., Winn, J.: The shape boltzmann machine: A strong model of object shape. Int. J. Comput. Vis. 107(2), 155–176 (2014)
24. Fei, B., Yang, W., Chen, W.M., Li, Z., Li, Y., Ma, T., Hu, X., Ma, L.: Comprehensive review of deep learning-based 3D point cloud completion processing and analysis. IEEE Trans. Intell. Transp. Syst. 23(12), 22862–22883 (2022)

25. Figliola, R.S., Beasley, D.E.: Theory and design for mechanical measurements. Meas. Sci. Technol. 7(7) (1996)
26. Gimenez, L., Robert, S., Suard, F., Zreik, K.: Automatic reconstruction of 3D building models from scanned 2D floor plans. Autom. Constr. 63, 48–56 (2016)
27. Guo, W.Y., Wang, Y., Dai, F., Xu, P.: Improved sine cosine algorithm combined with optimal neighborhood and quadratic interpolation strategy. Eng. Appl. Artif. Intell. 94(103779), 103779 (2020)
28. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE (2017)
29. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient convolutional neural networks for mobile vision applications (2017)
30. Iandola, F.N., Han, S., Moskewicz, M.W.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size (2016)
31. Kang, Z., Yang, J., Yang, Z., Cheng, S.: A review of techniques for 3D reconstruction of indoor environments. ISPRS Int. J. Geoinf. 9(5), 330 (2020)
32. Luo, Y., Mi, Z., Tao, W.: DeepDT: Learning geometry from delaunay triangulation for surface reconstruction. Proc. Conf. AAAI Artif. Intell. 35(3), 2277–2285 (2021)
33. Mahdjoubi, L., Moobela, C., Laing, R.: Providing real-estate services through the integration of 3D laser scanning and building information modelling. Comput. Ind. 64(9), 1272–1281 (2013)
34. Mlayeh, H., Othman, K.B.: Nonlinear accommodation of a DC-8 aircraft affected by a complete loss of a control surface(2022). Studies in Informatics and Control 31(3), 107–116 (2022)
35. O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J.: Deep learning vs. traditional computer vision. In: Advances in Intelligent Systems and Computing, pp. 128–144. Springer International Publishing, Cham (2020)
36. Pamucar, D., Simic, V., Lazarević, D., Dobrodolac, M., Deveci, M.: Prioritization of sustainable mobility sharing systems using integrated fuzzy DIBR and fuzzy-rough EDAS model. Sustain. Cities Soc. 82(103910), 103910 (2022)
37. Rafiq, M., Rafiq, G., Agyeman, R., Jin, S.I., Choi, G.S.: Scene classification for sports video summarization using transfer learning. Sensors (Basel) 20(6), 1702 (2020)
38. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). Narosa Publishing House (2002)
39. Vijaya Kumar, S., Nagaraju, C.: T2FCS filter: Type 2 fuzzy and cuckoo search-based filter design for image restoration. J. Vis. Commun. Image Represent. 58, 619–641 (2019)
40. Wang, G., Wang, Z., Gu, K., Li, L., Xia, Z., Wu, L.: Blind quality metric of DIBR-synthesized images in the discrete wavelet transform domain. IEEE Trans. Image Process. 29, 1802–1814 (2019)
41. Xiong, W., Zhu, Y., Zeng, Q., Du, J., Wang, K., Luo, J., Yang, M., Zhou, X.: Dose-effect relationship analysis of TCM based on deep boltzmann machine and partial least squares. Math. Biosci. Eng. 20(8), 14395–14413 (2023)
42. Xu, A., Tian, M.W., Firouzi, B., Alattas, K.A., Mohammadzadeh, A., Ghaderpour, E.: A new deep learning restricted boltzmann machine for energy consumption forecasting. Sustainability 14(16), 10081 (2022)
43. Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B.: A survey of modern deep learning based object detection models. Digit. Signal Process. 126(103514), 103514 (2022)

Qianying Zou (born in 1980) is a Professor from Chengdu, Sichuan, China. She holds a Master's degree and specializes in the fields of big data applications and artificial intelligence. Geely University of China, No. 123, Section 2, Chengjian Avenue, East New District, Chengdu, Sichuan Province. E-mail: 56471843@qq.com

Fengyu Liu (born in 1996) is an Assistant Lecturer from Xi'an, Shaanxi, China (Corresponding author). He holds a Bachelor's degree and focuses on data mining and front-end and back-end design. Geely University of China, No. 123, Section 2, Chengjian Avenue, East New District, Chengdu, Sichuan Province E-mail: i@liufengyu.cn

Yuan Liao (born in 1982) is a Lecturer from Guang'an, Sichuan, China. She holds a Master's degree and her research interests include engineering management and virtual reality. Geely University of China, No. 123, Section 2, Chengjian Avenue, East New District, Chengdu, Sichuan Province. E-mail: 29497521@qq.com

Received: July 25, 2023; Accepted: January , 2024.