SCIPEDIA

Vectorization for Tower Sketches Using Large Model and Deep Learning

Ziqiang Tang[1], Chao Han[1], Hongwu Li[1], Zhou Fan[1], Ke Sun[1], Yuntian Huang[1], Zhewei Xu[2] and Chenxing Wang[2,*]

[1] China State Grid Jiangsu Electric Power Co., Ltd., Construction Branch, Nanjing, 210011, China

[2] School of Automation, Southeast University, Nanjing, 210096, China

INFORMATION

Keywords:
Image generation
vector extraction
stable diffusion
ControlNet

DOI: 10.23967/j.rimni.2025.10.60398

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

RIMNI Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería

Vectorization for Tower Sketches Using Large Model and Deep Learning

Ziqiang Tang[1], Chao Han[1], Hongwu Li[1], Zhou Fan[1], Ke Sun[1], Yuntian Huang[1], Zhewei Xu[2] and Chenxing Wang[2,*]

[1]China State Grid Jiangsu Electric Power Co., Ltd., Construction Branch, Nanjing, 210011, China

[2]School of Automation, Southeast University, Nanjing, 210096, China

ABSTRACT

Background: 3D modeling of transmission towers from hand-drawn sketches is occasionally needed in the exploration stage, construction stage, or operation of transmission lines. To achieve automatic 3D modeling, extracting standard vectors from the sketches is very important, however, this is still lacking for real applications. **Methods:** In this article, we propose a method for automatically converting the hand-drawn design drawings of a tower into standard vector diagrams. The network ControlNet is introduced to control the image generation model Stable Diffusion, to convert hand-drawn sketches into standard line-structural diagrams. Then, a vectorization network is used to vectorize the line structure, and some designed optimization modules are proposed to refine the vector extraction results. **Results:** Experiments prove that our method can output the most accurate and robust vectorization results compared with some state-of-the-art methods. **Conclusion:** The proposed method is automatic and does not need manual operations, which can bring convenience for the intelligent design and modeling of transmission towers. Furthermore, it can also provide technical support for developing relevant digital products for the smart grid.

OPEN ACCESS

Received: 31/10/2024

Accepted: 13/02/2025

Published: 30/05/2025

DOI
10.23967/j.rimni.2025.10.60398

Keywords:
Image generation
vector extraction
stable diffusion
ControlNet

1 Introduction

During the construction of power transmission lines [1], designers sometimes sketch design drawings by hand at the initial stage of on-site exploration. At the end of construction, route drawings by hand may also be used for drone track planning during on-site intelligent acceptance or inspection. If the hand-drawn sketches can be converted into standard vector diagrams and then regenerated into corresponding 3D models, it will be easier to intuitively understand the expected effects and improve the intelligence of related work. There have been relevant achievements in automatic 3D modeling using vector drawings [2], however, the vectorization for hand-drawn tower sketches is still lacking.

Some image-processing methods have been developed to process sketches of specific symbols [3]. Murase et al. [4] proposed a sketch recognition algorithm to recognize hand-drawn symbols for flowchart diagrams. Durgun [5] identified various elements from architectural sketches. However, there

*Correspondence: Chenxing Wang (cxwang@seu.edu.cn).

is rare work for processing hand-drawn tower sketches, while making every stroke of the sketches a straight line is important for the following vectorization. Also, rare traditional methods conducted vectorization referring to extracting nodes and lines from drawings used for 3D modeling. He [6] proposed a dynamic space transformation method based on Freeman coding, which can accurately and quickly extract the skeleton and obtain high-precision vector data using arc segment measurement. In recent years, Zhang [7] extracted and reconstructed the center lines of line drawings with arbitrary widths based on boundary information and then vectorized them. Zhang [8] proposed an image vectorization algorithm based on boundary optimization and color interpolation. These methods based on traditional image processing highly rely on the setting of prior parameters, so they easily lose effect in practical cases having various interference. For a specific target task and application data, data-driven deep learning techniques are more robust because they can extract more deep features to complete target prediction or classification.

The vectorization of line drawings based on deep learning has developed to an extent recently. Bessmeltsev et al. [9] proposed a new image vectorization method called polyvector field, which can eliminate the ambiguity of vector nodes without compromising quality. Mo [10] constructed a general framework that learns the mapping from pixels to vectors, and so it can generate vector line drawings from different types of images. However, these two methods focused only on vectorizing the curves in stick figures. Ran [11] proposed the GASRNet model, which improved the ability to extract features of different scales in images and thus improved the accuracy of vectorization. Liu [12] proposed a structured end-to-end framework based on Transformer for online and efficient construction of vectorized high-definition (HD) maps. Other researchers conducted the vectorization based on the semantic segmentation tasks. Egiazarian et al. [13] pre-processed the drawings using the U-Net network [14], which includes eliminating redundant backgrounds, filling in missing parts, and segmenting line elements. They then encoded the image blocks with a Resnet-based feature estimator [15], decoded them with a Transformer module [16], and finally output the vector results. Wu et al. [17] used Mask-RCNN [18] to segment geometric shape contours and vectorize all rectangular detection boxes, then, they obtained the optimized vector topology by aligning the vertex coordinates of the vectors. These methods are robust and effective, but cannot be directly used for extracting the vectors of sketches for transmission towers.

A transmission tower is a kind of truss structure, thus the vector extraction of wireframes in images is very close to the task in this study. Zhou et al. [19] proposed an algorithm for detecting line frames in an image, known as L-CNN, which can directly output meaningful nodes and connect the nodes to form lines. This type of method is suitable for processing the drawings of transmission towers. However, it focuses on the position of nodes, affecting the topology of line vectors. A holistically-attracted wireframe parsing (HAWP) method [20] also detects the line segments effectively with some improvements for L-CNN. Recently, another improvement of L-CNN named DeepLSD [21] was proposed to detect line segments quickly and accurately. However, both HAWP and DeepLSD inevitably lead to many redundancies that interfere with the following tasks. To solve these problems, we reorganized the procedures of the sketch vectorization and designed some modules to refine and verify the output vectors.

The vectorization for the hand-drawn tower sketch aims to extract the nodes and line vectors to make the sketch drawings become standard drawings. Until now, we have not found similar work to handle this issue. Therefore, we propose a two-stage vector extraction strategy for tower sketches. In the first stage, a conditional control network named ControlNet is used to guide the large model Stable Diffusion to generate the straight-line diagrams from the hand-drawn sketches. In the second stage, the straight-line diagram is vectorized through an end-to-end trainable vectorization network, where

a line thinning module and some refinement modules are designed to remove redundant vectors. The experimental results demonstrate that the proposed method can successfully transform the sketches of tower design into standard vector diagrams, laying a foundation for the intelligent design of transmission lines and related automation applications.

2 The Proposed Method

2.1 The Overall Flowchart

The proposed vectorization of hand-drawn sketches for transmission towers has two stages: the first stage is to convert the sketch into a straight-line diagram; the second stage is the vectorization of extracting the nodes and line vectors. Fig. 1 shows the overall flow chart of the proposed method in this paper. In the first stage, the hand-drawn sketch is processed to generate a new drawing of straight lines using a large model of stable diffusion, where the generation process of the Diffusion model [22] is controlled by another network ControlNet [23]. Then, in the second stage, taking the generated straight-line drawing as input, the proposed vectorization method is applied to output the nodes and line vectors. The next sections will elaborate on the methods of the two stages in detail.

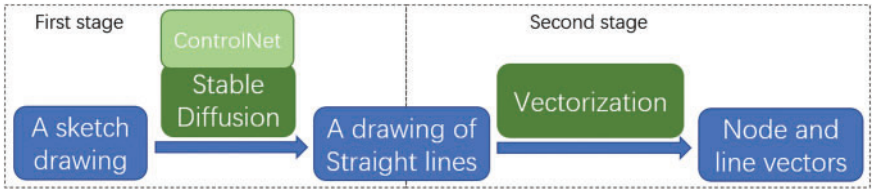

Figure 1: Flowchart of the proposed method in this paper

2.2 Straight-Line Conversion of Hand-Drawn Sketches Based on Large Model

The design drawings of a transmission tower are all straight-line structures. So, the vectorization of hand-drawn sketches serves two purposes: to make each stroke a standard line and to record the coordinates of each line and its endpoints. Traditional methods achieve these purposes with image processing techniques, which have many limitations, for example, the resulting shape cannot be standardized and still highly like the hand-drawn shape, the boundary points detection relies on the thresholding parameters setting, and the final line connections easily make mistakes. To illustrate these limitations, an example can be found in Fig. 2, where the traditional methods are implemented referring to [6–8].

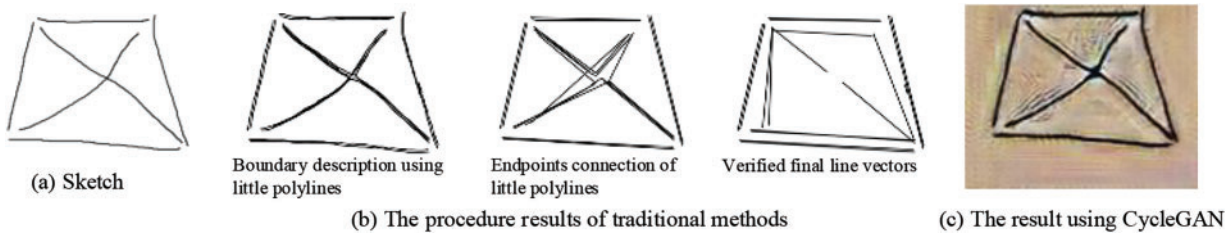

Figure 2: Illustration of issues using traditional methods and GAN model

Converting a hand-drawn sketch into a standard drawing of straight lines is equivalent to generating a new image. In recent years, generative network models have demonstrated powerful capabilities, therefore, we proposed a two-stage strategy that includes converting the sketch into

straight lines first and then extracting the nodes and line vectors from the straight lines. The generative network usually has two types: the generative adversarial network (GAN) and the diffusion model. CycleGAN [24] used to be a popular model for transferring image styles in these years, however, it generates information on all pixels, while the supervision of the lines is lacking, so the effect cannot be ensured. An example is shown in Fig. 2c, where the CycleGAN is trained using our training data.

To solve the issues above, we introduce the recently most powerful generative model, the diffusion model [25], to generate images of straight lines. Diffusion models can suppress semantically meaningless information by minimizing the relevant loss term, However, gradients and neural network backbones used during training and inference still need to be evaluated on all pixels. This process leads to redundant calculations and unnecessary optimization and reasoning. Therefore, we introduce a large model based on Stable Diffusion to generate standard straight lines, with the ControlNet to guide the generation of a shape. The flowchart is shown in Fig. 3.

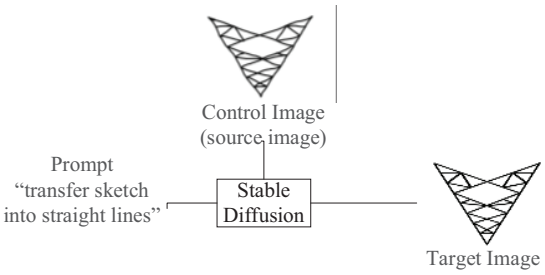

Figure 3: The flowchart of the conversion of hand-drawn pictures with stable diffusion + ControlNet

Stable Diffusion is an image generation model based on the diffusion process [22], designed to achieve high-quality and continuous image generation. It works by gradually adding noise to the input image and randomly blurring it, then progressively reducing the noise level, making the image clearer and more realistic. The key to this method is to achieve image smoothing and detail restoration through appropriate diffusion steps and noise control strategies, thereby ensuring that the image generation process has good stability and generalizability, and the results of image generation consistently maintain high quality.

The framework of Stable Diffusion is displayed in Fig. 4, which shows a conditional image generator by integrating a cross-attention mechanism into the UNet architecture of the diffusion model. This mechanism handles various input modality strings, such as language prompts. To handle these different modalities, Stable Diffusion introduces a domain-specific encoder that transforms the input into an intermediate representation. Then, this representation is utilized in the middle layer of UNet through a cross-attention layer, enabling the model to effectively integrate and focus on information from images and language prompts. This architecture includes a learnable projection matrix to facilitate the interaction between modalities, thereby improving the model's ability to generate relevant images based on input conditions.

ControlNet [23] is a neural network architecture designed to add spatial conditional control to large and pre-trained text-to-image diffusion models, thereby constraining content generation. The overall structure of the process is shown in Fig. 5. We use ControlNet to create trainable copies of 12 encoding blocks and 1 intermediate block of Stable Diffusion while locking the original neural network modules and then connecting these modules using 1×1 zero convolutional layers. In addition, during the connection process, a conditional vector is introduced as a condition added to the network.

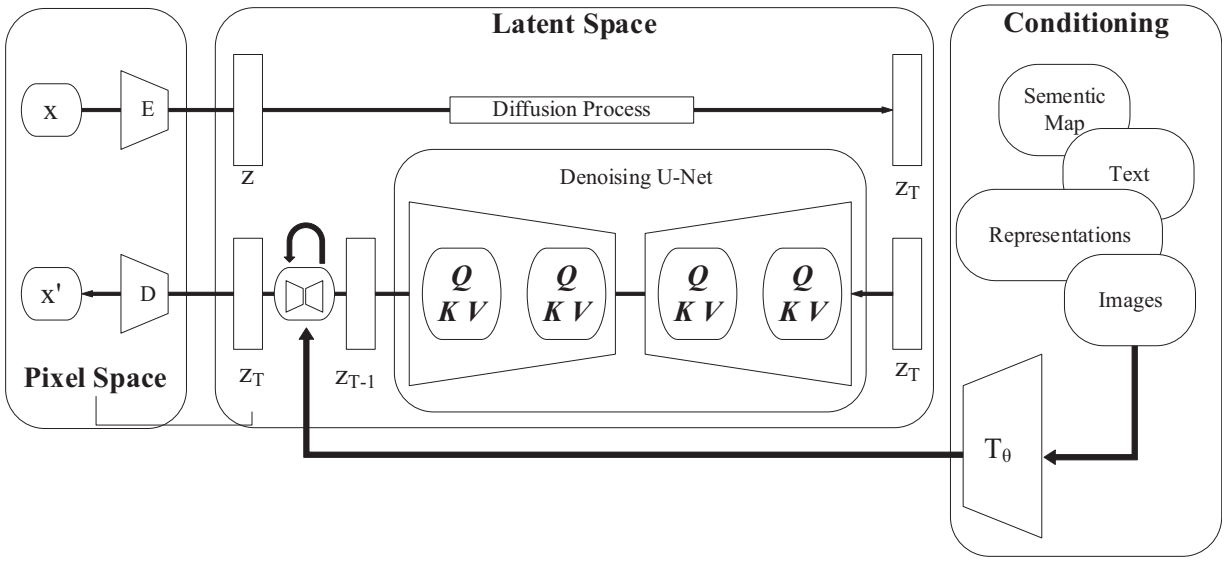

Figure 4: Framework of stable diffusion

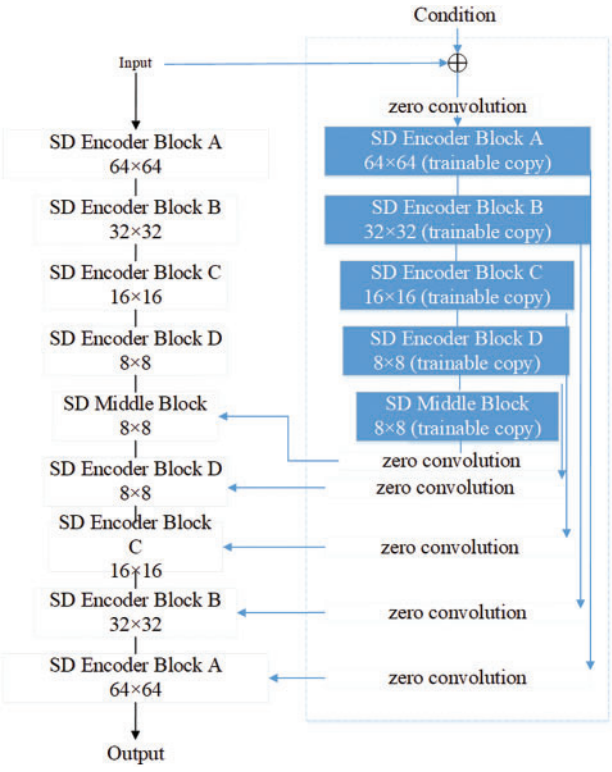

Figure 5: The process of ControlNet to control stable diffusion

During the training process of ControlNet, given an input image z_0, the image diffusion algorithm gradually adds noise to z_0, ultimately generating a noisy image z_t where t represents the number of times

noise is added. In this process, the algorithm learns a network ϵ_θ based on a set of conditions, including time step t, text prompt c_t, and task-specific condition c_f:

$$\mathcal{L} = \mathrm{E}_{z_0,t,c_t,c_f,\epsilon\sim\mathcal{N}(0,1)}\left[||\epsilon - \epsilon_\theta\left(z_0, t, c_t, c_f\right)||_2^2\right] \tag{1}$$

where $\mathcal{L}$ is the overall learning objective of the entire diffusion model, which is used to fine-tune the diffusion model and ControlNet. This network aims to predict the noise added to the image z_t.

Based on the above framework, Stable Diffusion and ControlNet are combined to convert the hand-drawn sketches to straight-line drawings. With ControlNet, Stable Diffusion can generate target images based on the input sketch and prompt words. Since this method aims to convert the sketch into a standard straight-line structure diagram, the prompt can be set to "transfer the sketch into straight lines".

2.3 The Vectorization Method

After converting the sketch into a straight-line drawing, we propose a vectorization method based on HAWP to extract the node and line vectors from the drawings. The complete vectorization method is shown in Fig. 6, which includes five main parts: the line thinning module, the initial proposal prediction, the proposal refinement, the proposal verification, and the postprocessing. The main principle and structure of these parts will be explained in the following.

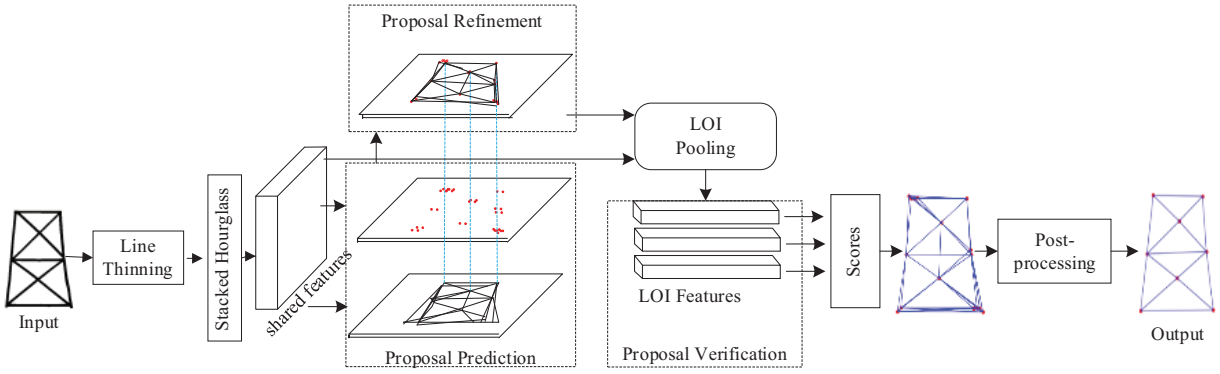

Figure 6: Framework of the vectorization method

Line-thinning module. If the sketch line has a width, there may be many candidates for predicting the nodes. The redundant nodes further lead to redundant line connections. Noticing this, we design a line-thinning module and add it in front of the network backbone. The thinning module is designed by referring to the morphological operations, i.e., eroding first and then dilating, with a cross-type convolution kernel having size 3×3, as shown in Fig. 7. The resulting map will be compared pixel-by-pixel with the one before line thinning until it is not changed. This thinning module can reduce the invalid nodes and lines without complicated calculations.

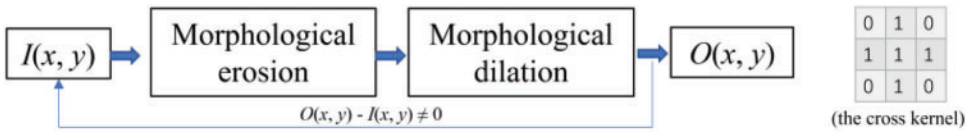

Figure 7: The flowchart of line thinning

Initial proposal prediction. The stacked Hourglass network [26] is adopted as the backbone network to extract deep features, which has been widely used in human joint estimation or corner-point-based object detection. This structure consists of multiple Hourglass modules, each including downsampling and upsampling processes. This enables the network to effectively capture multi-scale features, resulting in more accurate proposals of nodes and line vectors. More details can be found in [26]. Then, a head regressor based on heatmap representation is used to predict the proposals of connection points (the nodes) of the drawings, where the heatmap describes the probability of an object appearing at this distribution. In detecting the node points, a node mask map and a node offset map are obtained from the features output by the backbone, and these two types of maps are used in the heatmap-based head regressor through a loss constraint, to avoid the missing detection of nodes. Then, the top K connection points are selected as initial connection point proposals. So, a line vector is predicted with

$$\hat{d}_1(k) = \hat{d} + \lambda \cdot \Delta\hat{d} \tag{2}$$

where d is a line connected between two nodes, Δd is the distance residual mapping calculated by a 1×1 convolutional layer and an s-shaped layer, $\lambda = -1, 0, 1$. Therefore, according to $0 < \hat{d}_1 < d_{\max}$, the line proposals cannot be more than 3 for each distant point.

Proposal refinement. The node and line proposals are calculated separately using different information, and then matching the nodes and line proposals can provide more accurate meaningful alignment in wireframe parsing. The matching strategy is designed based on the Euclidean distance between the endpoints of a line proposal and the corresponding node proposals, respectively. A proposal will be retained only if the Euclidean distance is small enough.

Proposal verification. The refinement above screens unreasonable node and line proposals, while the obtained node and line proposals still need to be verified further with the ground truth. Therefore, we utilize a lightweight verification head classifier to classify the coupled node and line proposals separately and apply the line of interest (LOI) pooling operation to calculate the line segment features, then, we verify these proposals also by calculating the distance between two lines and then comparing the distance values. The verification head classifier is trained by assigning the positive and negative labels to line proposals (after refinement) based on their distances to the ground truth. The distance is computed by matching the two pairs of endpoints based on the minimum Euclidean distance. A threshold η ($\eta = 1.5$) is set to evaluate whether the line proposal is positive or negative.

Postprocessing module. After the above procedures, almost all nodes and lines can be detected and many redundant results have been filtered out. However, some false nodes still appear due to the repeated and fragmented lines in the straightening or line-thinning process. To solve this issue, we designed another postprocessing module to enhance the quality of vector extraction. As the coordinates of each node have been obtained, the line vectors can be recorded by the sequence number of nodes and the matching degree of a line vector described by weights. If the nodes are refined carefully, the invalid line vectors can also be removed with the deletion of invalid nodes. Thus, the postprocessing focuses on the recognition of nodes. We employ a single-link algorithm [27] to conduct hierarchical clustering of nodes. The dendrogram for clustering the nodes is displayed in Fig. 8. The nodes are clustered by evaluating the similarity between different nodes, where the similarity is calculated as the distance. The redundant nodes are removed according to the clustering results. Then, the relevant lines are updated according to the changes in the sequence numbers of nodes, so the duplicate edges can also be removed. After this postprocessing, the vectorization results are more concise, accurate, and structurally coherent.

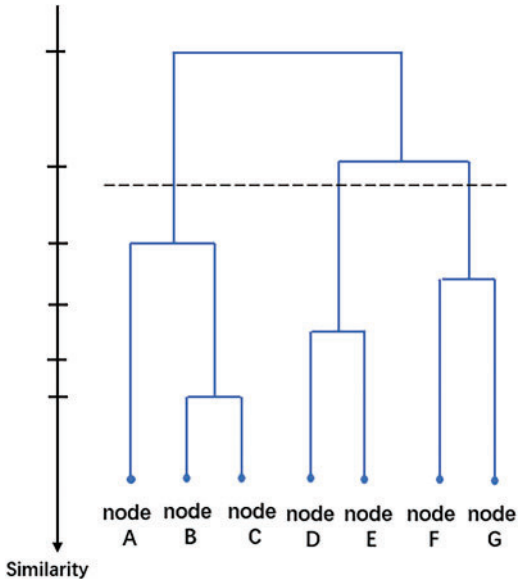

Figure 8: The dendrogram using a single-link algorithm

3 Experiments

3.1 Experiment Settings and Datasets

The experiments are conducted with GTX1650, Python 3.10.6, and Torch 2.0.1 + cu118. The version of Stable Diffusion is v1.5, the batch size is 4, and the learning rate is 10^{-5}.

Training the conversion of hand-drawn sketches into standard drawings requires numerous pairs of original hand-drawn sketches and standard drawings as the data foundation. Currently, no dataset can meet the requirements of this task scenario. Therefore, it is necessary to construct the corresponding dataset. In China, the difference in various towers focuses on the size of the tower body and the shape of the cross-arm, while the cross-arm also mainly shows a structure of straight lines. Fig. 9 below shows the outline of the common tower body and the common types of cross-arms respectively. There are also other unusual types of shapes different from Fig. 9. However, no matter the usual or unusual shapes, they are usually divided into several partial elements further, and each partial element shows a small polygon with a grid structure in the inner. Thus, the description of different types of towers is changed to the combinations of various polygons. We manually drew the corresponding draft drawings based on more than one thousand partial polygon drawings of the tower.

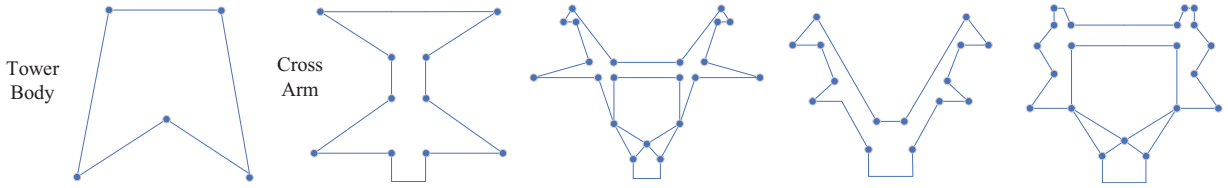

Figure 9: The outline shapes of the common tower body and the common cross-arms

Part of the dataset is shown in Fig. 10a, and some details are shown in Table 1. Our datasets were created referring to about 1157 practical design drawings from 20 different types of towers. These drawings cover most types of partial elements. We extract the mask from these drawings and process it

into straight-line drawings. It is worth mentioning that the light intensity of sketches might be diverse in real cases, like the cases in Fig. 10b, however, this interference does not affect its guidance of generating straight lines using Stable Diffusion. Further, we label the node and line vector on each straight-line drawing to create the datasets for training the vectorization network.

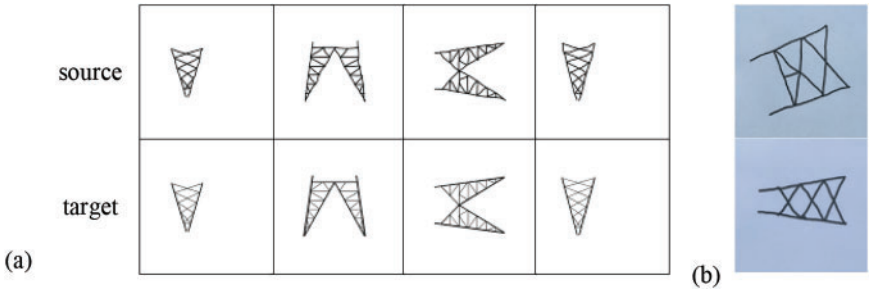

Figure 10: Illustration of some samples in: (**a**) dataset; (**b**) real case

Table 1: Details about the dataset

Number of drawings	Drawing resolution	Prompt word
1157	512 × 512	Transfer the sketch into straight lines

3.2 Evaluation of the Results of Straight-Line Conversion

We first display some temporary results in the training process, as shown in Fig. 11. In the first epoch, the generated results are uncontrollable and show no relation to the input. Along with the training, the results become close to the target in the 60th epoch.

We also conduct some comparisons to illustrate the effect of our method. There has not been a related network like this paper. However, the official ControlNet site provides a variety of models capable of transforming different types of line segments [25,28], which primarily include Canny edge detection, Lineart, and Scribble. Therefore, we compare the results of these models with our model all trained on our dataset. It is worth mentioning that these official models need to be processed by the corresponding preprocessor first and converted into the corresponding type of image, but our method needs not. The comparison results of some samples are shown in Fig. 12.

The results indicate that our method has successfully transformed hand-drawn curves into straight lines, while the results generated by the official ControlNet models are not satisfactory. The compared models focus more on the generation of textures and content rather than the transformation of straight lines, thus misinterpreting the prompt, leading to unnecessary lines in the background or framework of the results.

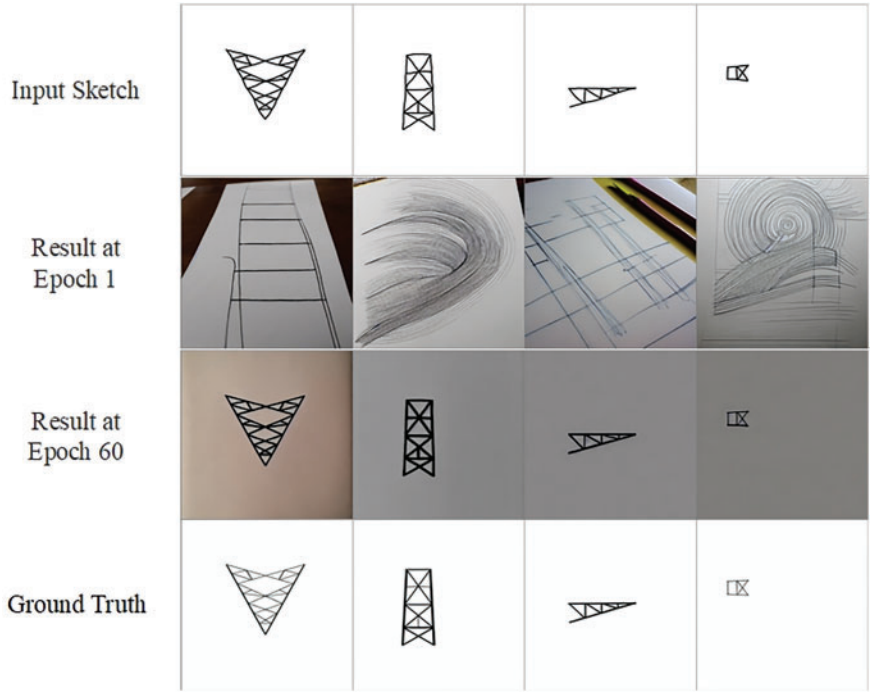

Figure 11: Visualization of some temporary training data

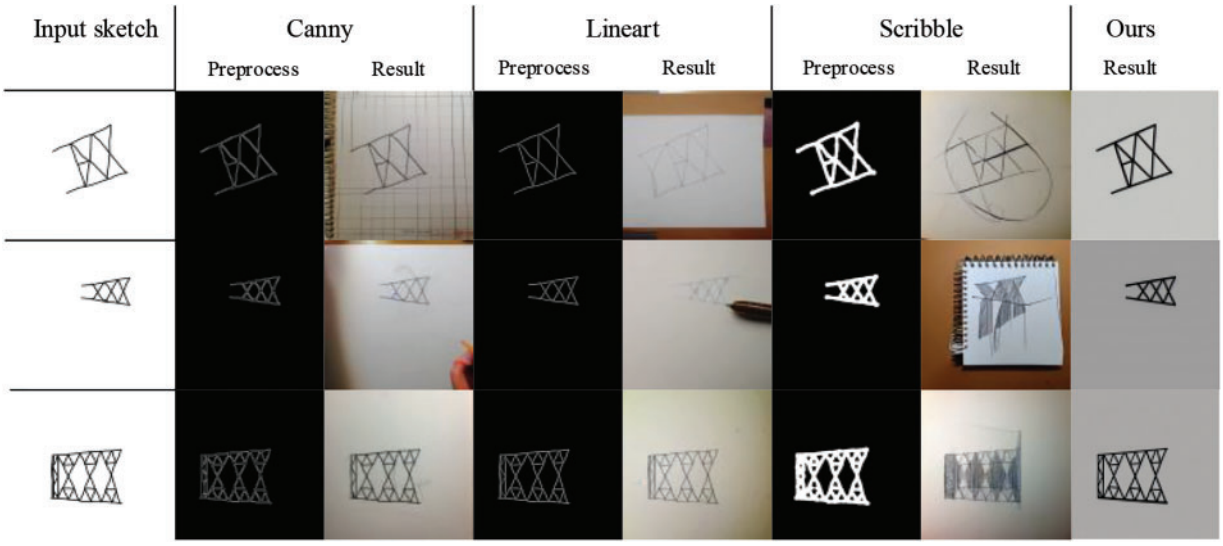

Figure 12: Evaluation of results on transforming sketches to straight lines using different models

To further quantify the results, we introduce two indices to compare these methods, the structural similarity index measure (SSIM) and the Hausdorff distance. SSIM [29] is a widely used index in

image processing, which measures the similarity of two images and is calculated by sliding windows on images. The measure between two windows x and y on two images respectively is

$$SSIM\,(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x{}^2 + \mu_y{}^2 + c_1\right)\left(\sigma_x{}^2 + \sigma_y{}^2 + c_2\right)} \tag{3}$$

where μ_x and μ_y are the mean of pixels within x and y, respectively; σ_x and σ_y are the variance of x and y, respectively; σ_{xy} is the covariance of x and y; c_1 and c_2 are two variables to stabilize the division with weak denominator. The SSIM between the sketch and the converted straight-line drawing measures the quality of the conversion. The value of SSIM tends to be 1, meaning that the converted image has a higher similarity to the ground truth. Sometimes, redundant false lines appear during the conversion, then the SSIM value tends to be smaller.

Hausdorff distance [30] is an indicator mainly used to compare two sets representing shape, contour, or path to evaluate the differences in their relative positions and distributions. The Hausdorff distance between a pair of non-empty subsets X and Y is defined as

$$d_H\,(X,\,Y) = \max\left\{\sup_{x\in X} d\,(x,\,Y)\,,\sup_{y\in Y} d\,(X,y)\right\} \tag{4}$$

where sup represents the supremum operator. The smaller the value of Hausdorff distance, the higher the similarity between the two sets, indicating that their shapes, positions, and sizes in space are relatively close.

Table 2 compares the results of different methods to the ground truth created by the manual, which is consistent with the visualization results in Fig. 12. Due to the false lines generated, the SSIM for the other three methods shows very low and the Hausdorff distance is large, while for our method, the SSIM is large and the Hausdorff distance is the lowest, meaning that the generated result is reasonable and maintains high accuracy to the ground truth.

Table 2: Quantification of results for different ControlNet models

ControlNet model	SSIM	Hausdorff distance
Canny	0.688	112.887
Lineart	0.744	78.103
Scribble	0.499	136.987
Ours	0.828	4.903

3.3 Evaluation of Vector Extraction

I. Comparisons

Since there is no work having the same task as this paper, to demonstrate the effectiveness, we selected the state-of-the-art most related methods, L-CNN [19], DeepLSD [21], and HAWP [20] for comparison. These methods are based on deep learning networks and have shown good results in vectorization for other applications in recent years. The compared methods are trained and tested on the same datasets constructed in this paper. We try to adjust the parameters best to train the model for these methods. The quantification is conducted with the commonly used metric, the structural average precision (sAP). Specifically, the sAP refers to the proportion of data correctly identified as positive

cases (i.e., true values) among all positive cases. A high sAP value means higher precision. To identify the correctness of the extracted line vector, a Euclidean distance is calculated and evaluated by

$$\min\left(||p_1^* - p_1||_2^2 + ||p_2^* - p_2||_2^2\right) \leq \theta \tag{5}$$

where (p_1^*, p_2^*) is the predicted endpoints of a line vector, (p_1, p_2) is the ground truth, and θ is a threshold. The metrics of sAP10 and sAP15 are calculated with thresholds set at 10 and 15 [19], respectively. The detailed results are shown in Table 3, which shows that our result is the best. Interestingly, our method and LCNN perform stable at the two different thresholds, but DeepLSD and HAWP both show distinct in the two situations. To further evaluate these methods clearly, we also visualize the results as in Fig. 13. There is a severe lack of nodes and line vectors for LCNN. The DeepLSD predicts the nodes and line connections of edge lines for both sides of each line, however, the ground truth just corresponds to those of skeleton lines. The HAWP also shows severe redundancy. For a strict threshold in sAP10, the redundant false detections reduced the ratio, so the DeepLSD and HAWP performed worse than LCNN even. For a loose constraint with a larger threshold, some redundant line vectors are thought correct and thus the indices of these two methods increase conversely. Overall, our method shows the best result.

Table 3: Comparison of quantitative indicators for vector extraction

Method	sAP10	sAP15
LCNN	75.0	77.1
DeepLSD	62.6	76.2
HAWP	64.7	85.3
Ours	92.1	94.9

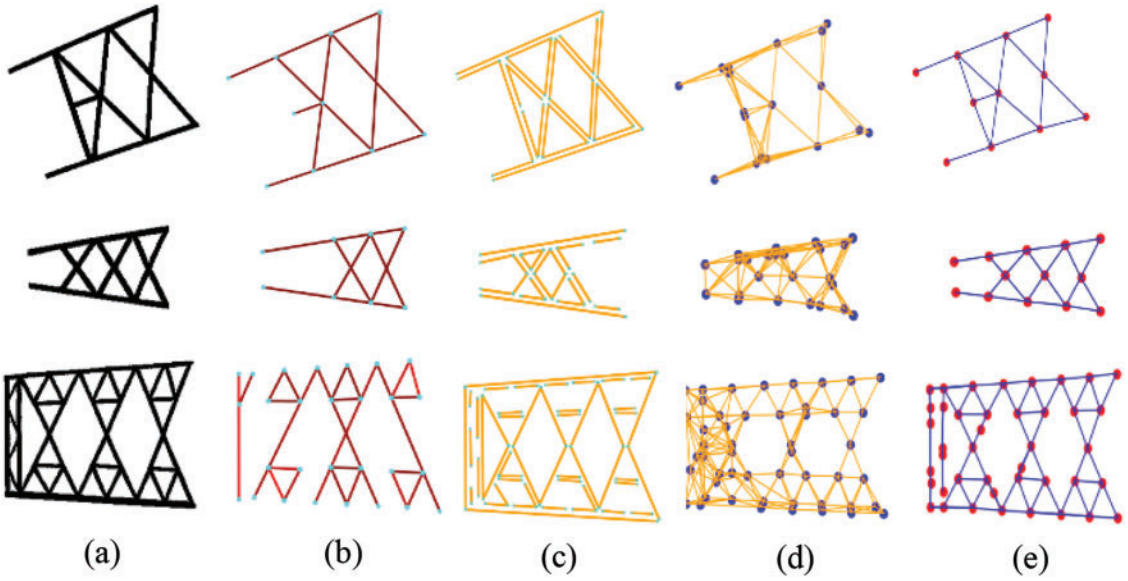

Figure 13: The vectorization results: (**a**) Input; (**b**) LCNN; (**c**) DeepLSD; (**d**) HAWP; (**e**) Our method

II. Ablation Experiment

We also conduct the ablation experiment for further analysis. The quantities and visualizations are displayed in Table 4 and Fig. 14 separately. Since our method is based on the HAWP method, it also has the problem that redundant node and line vectors easily occur. We add the line thinning module to reduce the redundancy caused by the width of lines first, then a postprocessing module is also added to reduce the unnecessary vectors further. The indices in Table 4 show the effectiveness of our added modules, and the visualization results in Fig. 14 are consistent with Table 4, which means the necessity of our modules.

Table 4: Results of the ablation experiment

Line thinning	Postprocessing	sAP10	sAP15
√		79.3	87.1
	√	88.4	93.4
√	√	92.1	94.9

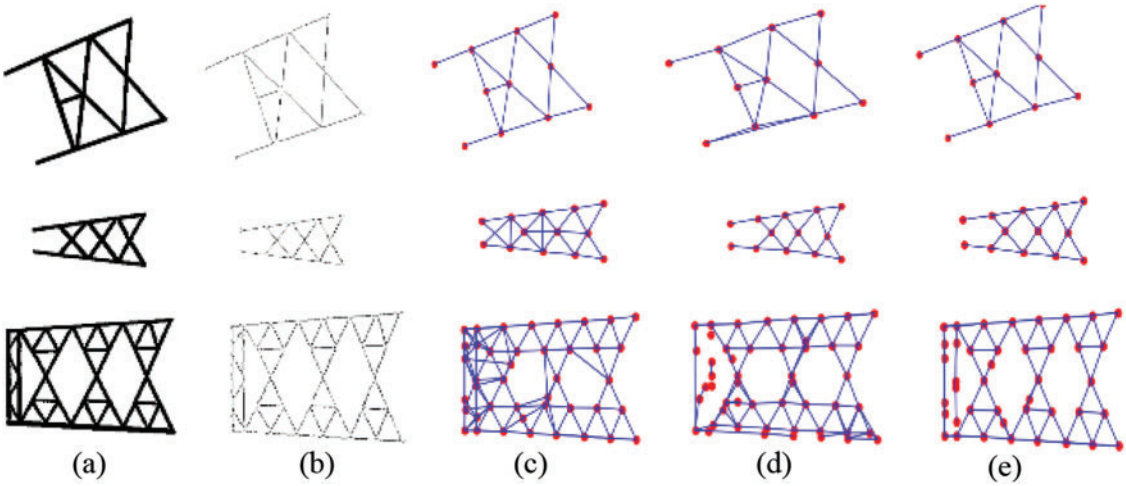

Figure 14: Ablation results: (**a**) Input; (**b**) Line thinning; (**c**) Result with lining thinning but without postprocessing; (**d**) Result without line thinning but with postprocessing; (**e**) Result with line thinning and with postprocessing

4 Conclusion and Discussion

In this paper, we propose a two-stage strategy for the standard vectorization of the hand-drawn sketches of transmission towers. Firstly, the hand-drawn sketch is converted into a straight-line drawing using the Stable Diffusion controlled by ControlNet. Then, a vectorization method is proposed, where a line-thinning module and a postprocessing module are designed to obtain a concise and accurate vector result. The experimental results demonstrate that this method can convert hand-drawn sketches into standard vector diagrams successfully. Due to the introduction of a large model, this method has applicability and robustness. The entire process is highly automated, providing convenience for the design and application of high-voltage transmission towers, and it holds significant importance for the comprehensive advancement of the digital transformation of power systems.

However, there are also some limitations for practical use. In the first stage, we convert a sketch into a straight-line drawing using the image-generative model. If the user draws a line without severe distortion, converting it into a straight line has no problem, otherwise, a line may be broken into two pieces of straight lines. As the example in Fig. 15 below, even though the generative shape has been standardized better, one of the cross lines is mistakenly bent due to the large arc of the sketch, which should be generated like the redline. In addition, the scale constraint of the shape wholly is still lacking, which means the sketch for drawing a square may be converted into a rectangle mistakenly.

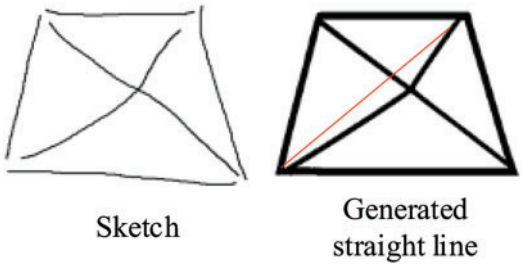

Figure 15: Illustration of some limitations

For the future, a lightweight classification network model can be constructed to verify each stroke draws a line or two lines. Furthermore, currently, the prompt for the Stable Diffusion is "transfer the sketch into straight lines". In the future, a prompt table can be created and provided to help guide the generation of more standardized shapes. In addition, exploring better networks for the task in this study is worthwhile.

Acknowledgement: We thank the Master's student Enzhi Xu for the help in creating the training datasets.

Funding Statement: This research was funded by the Chinese State Grid Jiangsu Electric Power Co., Ltd. Science and Technology Project Funding, grant number J2023031.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Ziqiang Tang, Zhewei Xu, Chenxing Wang; data collection: Zhou Fan, Yuntian Huang, Zhewei Xu; analysis and interpretation of results: Chao Han, Hongwu Li, Ke Sun; draft manuscript preparation: Zhewei Xu, Chenxing Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, C Wang, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. An X, Hou J, Gosling PD, Dai D, Xiao L, Zhou X. Improved failure mode identification and reliability estimates for electricity transmission towers. Comput Model Eng Sci. 2015;107(1):1–25. doi:10.3970/cmes.2015.107.001.

2. Tang Z, Han C, Li H, Fan Z, Sun K, Huang Y, et al. Automatic 3D modeling technique for transmission towers from 2D drawings. Mathematic. 2024;12(23):3767. doi:10.3390/math12233767.
3. Paul H, inventors; Paul H, assignee. Method and means for recognizing complex patterns. United States patent 3069654; 1962 Dec 18.
4. Murase H, Wakahara T. Online hand-sketched figure recognition. Pattern Recognize. 1986;19(2):147–60. doi:10.1016/0031-3203(86)90019-1.
5. Durgun FB. Architectural sketch recognition. Archit Sci Rev. 1990;33(1):3–16. doi:10.1080/00038628.1990.9696661.
6. He Y. Research on image vectorization based on mathematical morphology. Softw Guide. 2012;11(9):175–7. doi:10.1109/MEC.2011.6025677.
7. Zhang ZB. Vectorization algorithm for extracting centerlines of line drawings and its applications [master's thesis]. Shenzhen, China: Shenzhen University; 2022.
8. Zhang Z. Image vectorization method based on boundary optimization and color interpolation [master's thesis]. Ji'nan, China: Shandong University; 2022.
9. Bessmeltsev M, Solomon J. Vectorization of line drawings via polyvector fields. ACM Trans Graph. 2019;38(1):1–12. doi:10.1145/3202661.
10. Mo HR. General virtual sketching framework for vector line art. ACM Trans Graph. 2021;40(4):1–14. doi:10.1145/3450626.345983.
11. Ran WJ. Research on vectorization method of scan map line features based on deep learning [master's thesis]. China: Yunnan Normal University; 2023.
12. Liu QW. Construction of vectorized high-definition maps based on Transformer [master's thesis]. Liaoning, China: Liaoning University; 2023.
13. Egiazarian V, Voynov O, Artemov A, Volkhonskiy D, Safin A, Taktasheva M, et al. Deep vectorization of technical drawings. In: Proceedings of the European Conference on Computer Vision; 2020 Aug 23–28; Glasgow, UK. p. 582–98.
14. Ronnberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention; 2015 Oct 5–9; Munich, Germany. p. 234–41.
15. He KM, Zhang XY, Ren SQ. Deep residual learning for image recognition. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA. p. 770–8.
16. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceeding of 31st Conference of Neural Information Processing Systems; 2017 Dec 4–9; Long Beach, CA, USA. p. 1–11. doi:10.5555/3295222.3295349.
17. Wu Y, Shang J, Chen P, Zlatanova S, Hu X, Zhou Z. Indoor mapping and modeling by parsing floor plan images. Int J Geog Inf Sci. 2021;35(6):1205–31. doi:10.1080/13658816.2020.1781130.
18. He KM, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision; 2017 Oct 22–29; Venice, Italy; 2017. p. 2961–9. doi:10.1109/ICCV.2017.322.
19. Zhou YC, Qi HZ, Ma Y. End-to-end wireframe parsing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019 Oct 27–28; Seoul, Republic of Korea. p. 962–71. doi:10.1109/ICCV.2019.00105.
20. Xue N, Wu T, Bai S, Wang F, Xia GS, Zhang LP, et al. Holistically-attracted wireframe parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Seattle, WA, USA. p. 2788–97.
21. Pautrat R, Barath D, Larsson V, Oswald S, Pollefeys M. Line segment detection and refinement with deep image gradients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 17327–36.

22. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 18–24; Vancouver, BC, Canada. p. 10684–95.
23. Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023 Oct 1–6; Paris, France. p.3836–47.
24. Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision; 2017 Oct 22–29; Venice, Italy. p. 2242–51. doi:10.1109/ICCV.2017.244.
25. Daneshfar F, Bartani A, Lotfi P. Image captioning by diffusion models: a survey. Eng Appl Artif Intell. 2024;138(A):109288. doi:10.1016/j.engappai.2024.109288.
26. Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. In: Computer Vision—ECCV 2016. ECCV 2016. Lecture Notes in Computer Science; 2016 Oct 11–16; Amsterdam, the Netherlands. p. 483–99. doi:10.1007/978-3-319-46484-8_29.
27. Bridges J, Cecil C. Hierarchical cluster analysis. Psychol Rep. 1966;18(3):851–4. doi:10.2466/pr0.1966.18.3.851.
28. Lllyasviel. [cited 2024 Oct 31]. Available from: https://huggingface.co/lllyasviel/ControlNet-v1-1/tree/main.
29. Geometrical transformations and registration [Internet]. [cited 2025 Jan 5]. Available from: https://scikitimage.org/docs/0.24.x/auto_examples/transform/plot_ssim.html.
30. Munkres J. Topology. 2nd ed. London, UK: Pearson; 2003.