DSNL IN ARCHITECTURE– A DEEP LEARNING APPROACH TO DECIPHERING ARCHITECTURAL SKETCHES AND FACILITATING HUMAN-AI INTERACTION

WEI HU[1,2]
[1]*College of Architecture and Urban Planning, Tongji University*
[2]*Urban Computing Lab, Shanghai Artificial Intelligence Laboratory*
[1]*1990huwei@sina.com, ORCID: 0000-0001-7754-1858*

Abstract. The language of interaction between architects and machines has been evolving towards a more user-friendly paradigm. As the capabilities of machines and artificial intelligence have advanced, it has become increasingly feasible for architects to communicate with machines using their customary expressive methods. Consequently, this has led to the development of Domain-Specific Natural Language (DSNL), which, unlike traditional Domain-Specific Language (DSL), places greater emphasis on naturalness. While this naturalness enhances usability for architects, it also presents challenges in machine comprehension. To address this issue, we propose a data-driven approach that utilizes domain-specific data for model training or fine-tuning through unsupervised or weakly supervised methods. Our study, which focuses on teaching AI to learn architectural sketching from architects, demonstrates that our proposed method captures the characteristics of human architectural sketching more effectively than traditional approaches.

Keywords. Domain Specific Natural Language, Human-AI interaction, Architectural sketches, AIGC, Deep learning.

1. Introduction

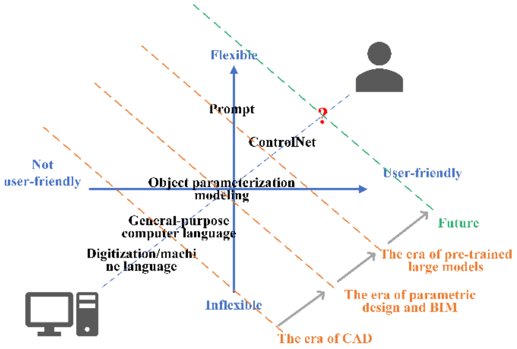

Flexible
Prompt
?
ControlNet
Not user-friendly
Object parameterization modeling
User-friendly
General-purpose computer language
Future
Digitization/machi ne language
The era of pre-trained large models
The era of parametric design and BIM
Inflexible
The era of CAD

Figure 1. Trends in the Development of Interaction Languages between Architects and Machines

ACCELERATED DESIGN, Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2024, Volume 1, 119-128. © 2024 and published by the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

Recent technological advancements have significantly advanced the role of Artificial Intelligence (AI) in creative design fields. Despite some algorithms impressing experts, AI still lacks the capacity to completely replace human experts in design. Consequently, a practical approach is to merge human expertise with AI, fostering effective collaboration. Effective communication stands central to this collaboration. An examination of the evolution in human-machine interaction (Hu, 2023), depicted in Figure 1, reveals a progression from basic machine languages to more complex systems like CAD, BIM, and PLM (Pre-trained Language Model). Advances in machine and AI technologies have led to more human-centric interaction patterns, signifying a shift towards user-friendly interfaces and processes. The ongoing evolution of AI and machine capabilities raises important considerations regarding the future nature of interaction between architects and machines.

Therefore, building upon DSL-based interaction methods like BIM and visual programming (Fowler, 2010), we propose that future human-machine interaction languages will more closely align with human expressive habits, offering greater flexibility and user-friendliness while better addressing professional needs. In the future, design experts could teach computers to understand 'Domain-Specific Natural Language (DSNL)', rather than adapting their own habits to computers and AI. DSNL primarily introduces the characteristic of naturalness, in addition to the linguistic aspects, domain specificity, and limited expressiveness of DSN. DSNL, in contrast to typical computer languages, mirrors human expression habits more closely. This approach makes interacting with computers more intuitive, akin to human-human communication, and reduces the cognitive load on users. While naturalness enhances user-friendliness, it also presents challenges for computer understanding, especially when large amounts of labelled data are difficult to obtain in a particular domain. We propose using a data-driven approach to train miniaturized models in unsupervised or weakly supervised ways or fine-tune pre-trained large models to solve this problem.

This paper investigates the practicality of DSNL through the novel approach of instructing an AI model to sketch buildings. Architectural sketching, a common DSNL among architects, offers insight into their design thinking and cognitive processes (Lawson, 2012). We introduce a method to impart this DSNL to AI, inspired by the Beaux-Arts educational system's master-apprentice teaching model (Chen, 2020). This method posits that, akin to the transmission of sketching techniques in architecture, humans and machines can similarly learn domain-specific natural language. Traditional algorithms struggle to teach ambiguous language to AI due to their reliance on structured rules. This study introduces architectural sketches as a training set for teaching DSNL to AI, utilizing data-driven deep learning techniques. To overcome the scarcity of labelled domain-specific samples, our algorithm efficiently trains smaller networks using more readily available unpaired datasets.

The experimental evaluations of our method reveal significant improvements in performance compared to various traditional approaches. Notably, our method excels in accurately interpreting the often ambiguous and nuanced meanings that professional architects convey through DSNL in their sketches.

2. Related works

Historical research on transforming scene images into sketches has predominantly used computer graphics algorithms for style conversion, such as Lu's method that combines image gradients and forward differences for edge extraction and style transfer (Lu, 2012). Additionally, smartphone camera filters and sketch conversion apps, such as "Photo Sketch Maker" and "Pencil Sketch," have gained popularity. However, these methods only replicate the image's "form" and lack value judgments and aesthetic choices based on the image's "meaning," which are crucial for architects and artists in their creative process.

Recent developments, including DALL-E (Ramesh, 2021; Ramesh, 2022), Stable Diffusion (Rombach, 2021), and ControlNet (Zhang, 2023), have innovated in image generation by introducing natural language prompts and image pair control. However, they struggle to capture the nuanced expression in architectural design, a task that is challenging to achieve through text alone or with precise image pairs. Even with the added edge detection techniques, such as Canny and HED (Xie, 2015), fail to replicate the perceptual richness found in architectural sketches.

3. Method

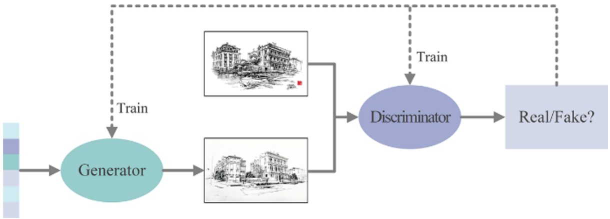


Figure 2. The schematic diagram of GAN

This study attempts to use Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) to transform architectural scene images into sketches. This involves a network comprising a generator and a discriminator, functioning in a zero-sum game to enhance image generation fidelity, as depicted in Figure 2. However, a basic GAN falls short for our goal of generating architectural sketches that correspond accurately to specific scene photos. Consequently, we turn to Conditional GANs (cGANs) (Mirza, Osindero, 2014), trained with scene and sketch image pairs, to ensure the relevance between the generated sketch and the original scene image. Owing to the scarcity of matched scene-sketch pairs, we adopt the Cycle-GAN structure (Zhu et al. 2017) for image-to-image domain mapping without the necessity of matched pairs. Cycle-GANs employ two interconnected GAN networks to guarantee a one-to-one correspondence between actual and generated images, utilizing similarity-based loss functions to regulate the generated content, as illustrated in Figure 3.

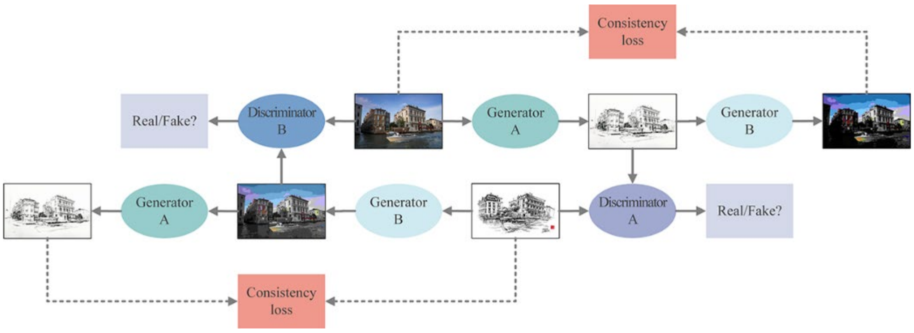


Figure 3. The schematic diagram of Cycle GAN

The generator that transforms images from the photo domain X to the sketch domain Y is denoted as Generator A or $G_A : X \rightarrow Y$. Its objective is to generate corresponding sketch y' from input photo $x \in X$. The quality of the generated sketches is evaluated by the discriminator, referred to as Discriminator A or D_A. On the other hand, the generator that transforms images from the sketch domain to the photo domain is denoted as Generator B or $G_B : Y \rightarrow X$. Its purpose is to generate corresponding photo x' from input sketch $y \in Y$. The quality of the generated photos is assessed by the discriminator, referred to as Discriminator B or D_B.

Generator G_A loss function:

$$\mathcal{L}_{G_A} = \mathcal{L}_{adv}(D_A, y') + \lambda_{cycle}\mathcal{L}_{cycle}^{A\rightarrow B}(x, x")$$

Discriminator D_A loss function:

$$\mathcal{L}_{D_A} = \mathcal{L}_{adv}(D_A, y) + \mathcal{L}_{adv}(D_A, y')$$

Generator G_B loss function:

$$\mathcal{L}_{G_B} = \mathcal{L}_{adv}(D_B, x') + \lambda_{cycle}\mathcal{L}_{cycle}^{B\rightarrow A}(y, y")$$

Discriminator $D_B$loss function:

$$\mathcal{L}_{D_B} = \mathcal{L}_{adv}(D_B, x) + \mathcal{L}_{adv}(D_B, x')$$

A key aspect of this model is its use of cyclic mapping between two domains to generate cyclic losses, effectively addressing the lack of one-to-one corresponding training samples. In the Photo-to-sketch-to-photo cycle, where an input photo x undergoes transformation into a generated sketch $y' = G_A(x)$ and subsequent reconstruction into a photo $x'' = G_B(y')$, the objective is to minimize the dissimilarity between the original photo x and the reconstructed photo x''. Similarly, in the Sketch-to-photo-to-sketch cycle, involving an input sketch y transformed into a generated photo $x' = G_B(y)$ and subsequently reconstructed into a sketch $y'' = G_A(x')$, the objective is to minimize the dissimilarity between the original sketch y and the reconstructed sketch y''.

The cyclic mapping is vital for establishing a bidirectional relationship between the two domains, allowing the model to preserve essential characteristics of the input data. Enforcing cycle consistency objectives enables the model to understand the structure and features of each domain, even without direct correspondence between individual training samples. This method aids in learning meaningful mappings, contributing to successful, cycle-consistent domain translation.

Cycle consistency loss functions:

$$\mathcal{L}_{\text{cycle}}^{A\to B}(x, x'') = MSE(x, x'') = |x - x''|_1$$
$$\mathcal{L}_{\text{cycle}}^{B\to A}(y, y'') = MSE(y, y'') = |y - y''|_1$$

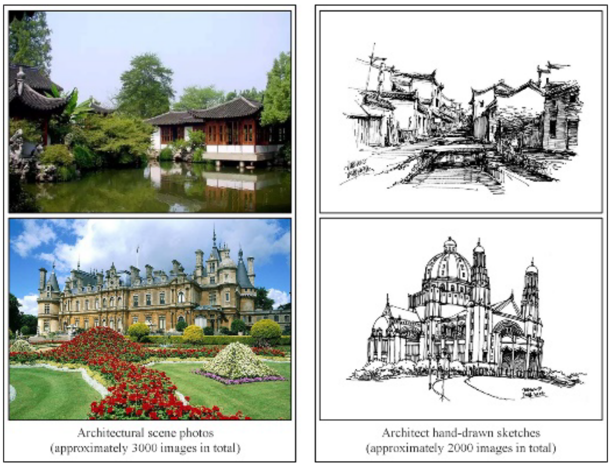
$\mathcal{L}_{adv}$ represents the adversarial loss, in this study we apply binary cross-entropy loss, λ_{adv} and λ_{cycle} are weighting coefficients to balance different components of the generator loss. $| \cdot |_1$ denotes the L1 norm, used to measure the discrepancy between images.

In our research, we incorporate ResNet (He et al., 2016) as the foundational architecture for the generator within the Cycle-GAN network, and employ Patch-Discriminator (Isola et al., 2017) as the discriminator. To balance the generated results and compute resource cost, the input network image size is transformed, and the images are pre-processed and scaled to 512×512 pixels during training and application. The generated result images are then scaled back to their original size.

4. Dataset and Model Training

We developed a web crawler to collect architectural scene and sketch photos, using keywords such as 'architectural scene' and 'architect sketch'. We then refined this dataset by eliminating duplicates, text-laden images, book photos, low-resolution images, and those with watermarks or conspicuous text. This process resulted in a final dataset comprising about 2900 architectural scene photos and 1600 sketches, forming an unpaired image-to-image dataset, as shown in Figure 4.

We configured our experiment with a maximum of 500,000 training epochs, performing validations at every 1,000th epoch. Training parameters yielding the lowest validation loss were saved, with the entire process taking approximately 100 hours. The lowest validation loss occurred at epoch 395,000, and we selected the model parameters from this epoch for subsequent analyses, as illustrated in the photo-to-sketch conversion examples in Figure 5.



Architectural scene photos
(approximately 3000 images in total)

Architect hand-drawn sketches
(approximately 2000 images in total)

Figure 4. The data samples used in this paper

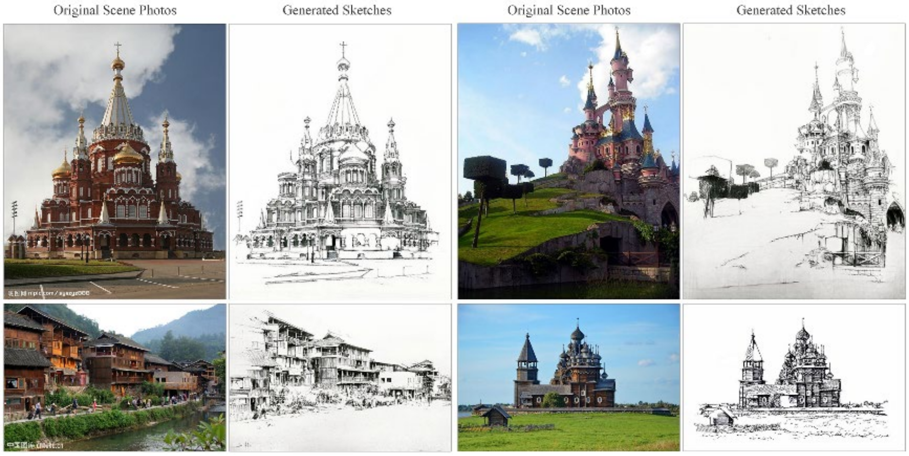


Figure 5. The architectural scene image input and the sketch image generated by the trained model

5. Experimental Results Compared with Existing Methods

This experiment focuses on training AI to interpret architects' specialized sketching language using data-driven methods. We conducted a comparative analysis, involving two distinct approaches. First, we replicated Lu's (2012) rule-based method, which uses image gradients and forward differences for edge extraction and style transfer. Second, we evaluated popular smartphone sketch generation apps, which are based on pre-training algorithms using non-domain-specific datasets.

In our replication of Lu's method using parameter sets A and B, we observed distinct results. Set A focused on outline strokes, while Set B emphasized image edges. However, both approaches predominantly concentrated on the fine details of architectural scene photos, rather than their overall content. In contrast, our method effectively mimics architect sketches in both strokes and lighting and uniquely interprets the image's meaning. For instance, in scenes with trees, lawns, or rivers, it sketches only the edges near buildings, leaving the rest blank. This demonstrates the AI's ability to process sensory semantic information in a manner akin to architects, as exemplified in Figure 6.

We also compared our method with popular sketch generation applications like 'My Sketch' and 'Photo Sketch Maker', which are widely used on iOS and Android platforms. These apps transform scene photos by applying computer graphics principles to aspects like colour, texture, and edges, with 'My Sketch' incorporating an additional simulated sketch stroke mask. Despite their popularity, these applications lack the capability to process semantic information. Conversely, our method not only captures stroke textures but also semantically condenses scenes, simplifying non-architectural elements such as backgrounds, and accentuating architectural details in a distinct manner, as demonstrated in Figure 7.

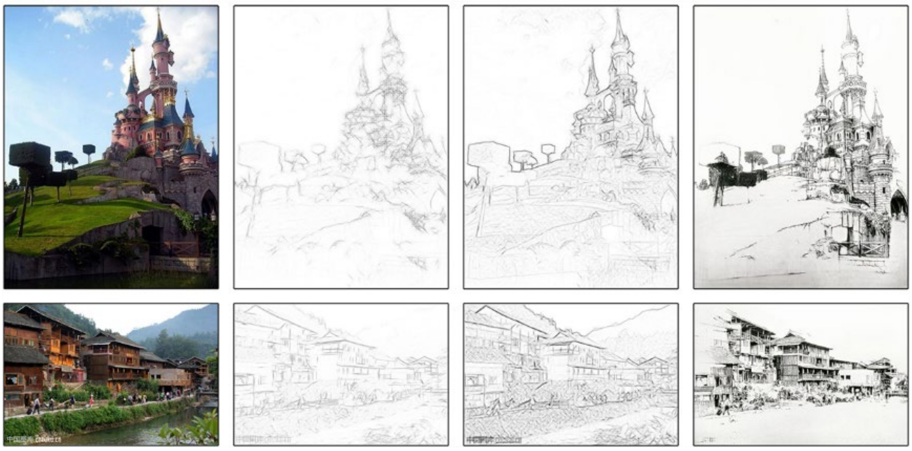


Figure 6. Comparison between Our Method and the Rule-based Method (Left 1: Original Image; Left 2: Parameter A; Left 3: Parameter B; Left 4: Our Method)

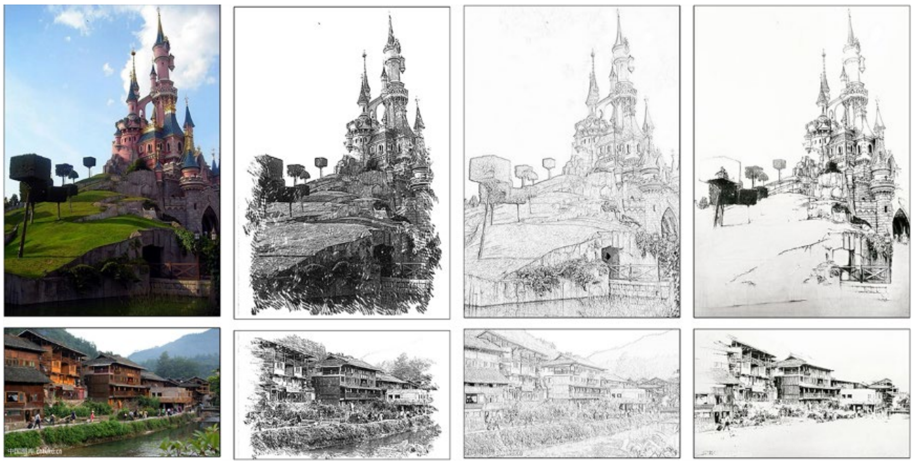


Figure 7. Comparison between Our Method and Apps (Left 1: Original Image; Left 2: My Sketch; Left 3: Photo Sketch Maker; Left 4: Our Method)

Comparative results, shown in Figure 8, reveal that our method more effectively replicates human architects' semantic processing than the compared apps. Our method intricately details key buildings, streamlines distant structures, and often leaves areas such as water surfaces blank. However, due to limited data volume and computational power, a discernible gap remains between our AI-generated results and genuine architect sketches. For example, our model tends to oversimplify elements like yachts, typically detailed in architects' sketches, due to their absence in our training dataset. We anticipate these limitations will lessen as the training dataset expands.

To substantiate the analysis of various methods, we conducted a quantitative survey with a matrix scale questionnaire. This questionnaire was tailored to assess factors architects deem crucial in sketches, like expressive strokes, emphasis variance, vividness, aesthetics, and artistic conception. Participants first viewed architectural scene photos, followed by images generated by My Sketch, the rule-based method, our method, and Photo Sketch Maker. They evaluated six aspects: Visual Appeal, Closeness to Human Stroke, Emphasis Variance, Vividness, and Artistic Conception, on a scale from 'very dissatisfied' (1) to 'very satisfied' (5).

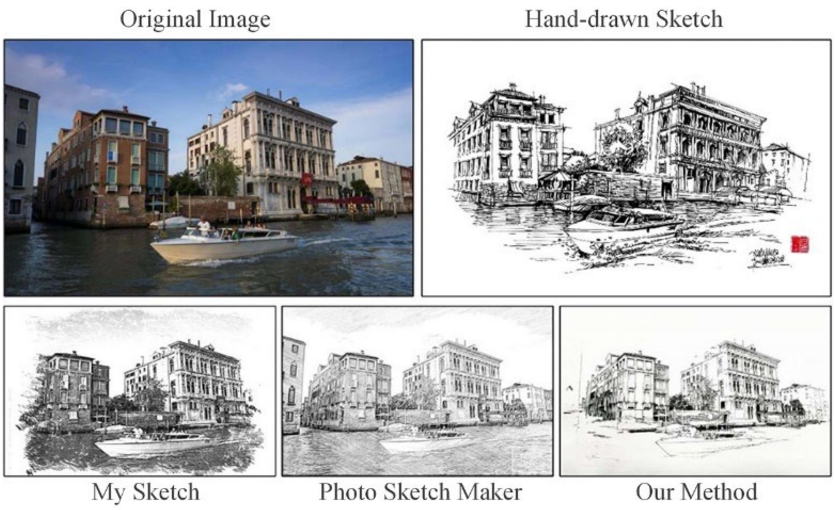


Figure 8. Comparison between Architect Sketch and Machine-generated Sketch

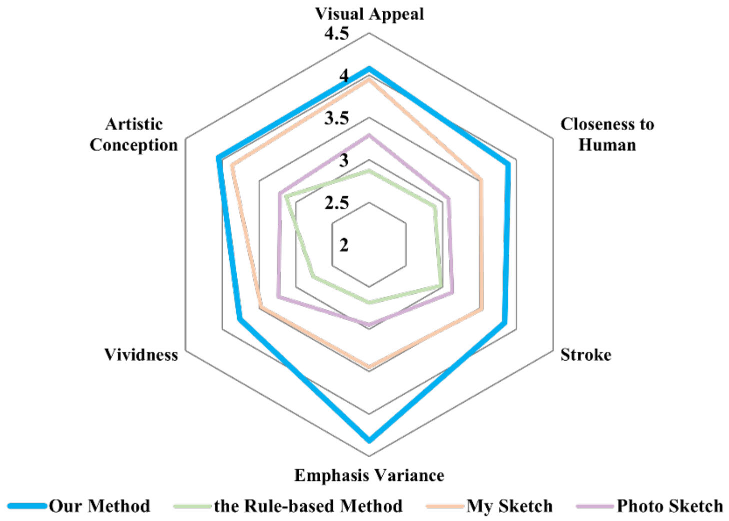


Figure 9. Rating Scores of Four Generated Results by Participants with Professional Background

The survey collected 50 valid matrix scale questionnaires, with 37 respondents from architecture/design backgrounds and 13 from non-architecture/design/fine arts. The evaluation results of participant-rated sketches, as shown in Figure 9, reveal that our proposed sketch generation method achieved the highest average scores across all metrics, especially in Emphasis Variance, among professionals, significantly surpassing existing methods. This highlights its ability to reflect architects' professional knowledge and skills.

Figure 10 displays the average ratings for each method. Participants with architecture or fine arts backgrounds favoured our method over existing ones. Conversely, non-professionals showed a preference for the visually polished sketches from 'My Sketch', with our method as a close second. This indicates that our method effectively captures the subtleties of professional architectural sketches while maintaining broad visual appeal.

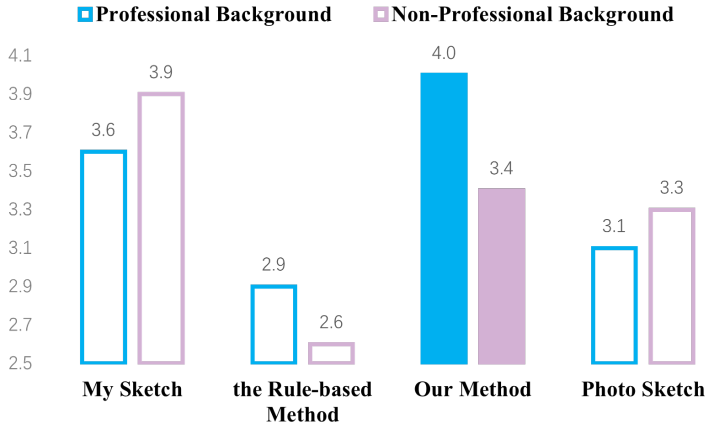


Figure 10. Mean Rating Scores of Four Results by Participants with Different Backgrounds

6. Conclusions

Architectural sketching, a natural language used by field experts, exhibits the proposed DSNL characteristics. We use data-driven deep learning methods to solve problems that computers have difficulty understanding naturalized domain-specific expressions. This involves collecting unpaired domain-specific datasets and employing relatively small neural networks, optimized for training with such datasets.

In this case study, we trained an AI algorithm to learn the sketching language used by architects by using a collected dataset of architectural sketches. The trained algorithm was then used to generate results that resemble hand-drawn sketches by architects. Compared to the results obtained by traditional methods, the results obtained by our proposed method include not only more visual information but also a selective process of semantic information, similar to how architects use their own judgment and aesthetic preferences to selectively depict architectural scenes in sketches, emphasizing certain aspects and differentiating between primary and secondary elements. The findings from the matrix scale survey also support the analysis of the experimental results.

Through this experiment, we have demonstrated that AI algorithms can be used to enable machines to use language that is more free-form, flexible, and inclusive of more ambiguity and generality, similar to that used by human domain experts, thus expanding the possibility space in meaning communication. This freedom and flexibility are crucial in the early stages of creative work, particularly in fields that require the use of DSNL.

References

Hu, W., Wang, X., Wang, D., Yao, S., Mao, Z., Li, L., Wang, F.-Y., & Lin, Y. (2023). Ir design for application-specific natural language: A case study on traffic data. arXiv preprint arXiv:2307.06983.

Fowler, M. (2010). Domain-specific languages. Addison-Wesley Professional.

Lawson, B. (2012). What designers know. Routledge.

CHEN Chi, L. C. (2020). Return with success: The influence of "collegiate" tradition on chinese academic tradition and architectural practice. Journal of World Architecture, No.356, 125–128. https://doi.org/10.16414/j.wa.2020.02.024

Lu, C., Xu, L., & Jia, J. (2012). Combining sketch and tone for pencil drawing production. Proceedings of the symposium on non-photorealistic animation and rendering, 65–73.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-shot text-to-image generation. International Conference on Machine Learning, 8821–8831.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.

Zhang, L., & Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:2302.05543.

Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. Proceedings of the IEEE international conference on computer vision, 1395–1403.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Zhu, J. -Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE international conference on computer vision, 2223–2232.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 1125–1134.