HILITE: Human-in-the-loop Interactive Tool for Image Editing

1st Arya Pasumarthi
OpenNLP Labs
arya.p@opennlplabs.org

1st Armaan Sharma
OpenNLP Labs
armaan.s@opennlplabs.org

1st Jainish H. Patel
OpenNLP Labs
jainish.p@opennlplabs.org

1st Ayush Bheemaiah
OpenNLP Labs
ayush.b@opennlplabs.org

1st Subhadra Vadlamannati
Stanford University
subha2@stanford.edu

2nd Seth Chang
OpenNLP Labs
seth.c@opennlplabs.org

2nd Sophia Li
OpenNLP Labs
sophia.l@opennlplabs.org

2nd Eshaan Barkataki
OpenNLP Labs
eshaan.b@opennlplabs.org

3rd Yutong Zhang
Stanford University
yutongz7@stanford.edu

4th Diyi Yang
Stanford University
diyiy@stanford.edu

5th Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu

6th Simran Khanuja
Carnegie Mellon University
skhanuja@cs.cmu.edu

***Abstract*—Image editing tools have a plethora of commercial and creative applications — content-creation, digital photography, advertisements, graphic design, and development of educational media. The shortcomings of image editing software include difficulty of use and, for AI-based software, reliance on single image editing models, which often poses the dilemma of a tradeoff between image editing quality and user-friendliness. While the performances of individual image editing models have improved with their evolution over time, these singular models are often specialized on specific image editing tasks. In this work, we introduce HILITE, an open-source interactive image editing platform with a human-in-the-loop design that combines six diffusion-based image editing models. For one, HILITE's accessible and easily-understandable user interface provides a straightforward user workflow from image input and prompt entry to selection of desired output. Secondly, the combination of several models with diverse specializations in turn allows HILITE to generalize on a wide variety of image editing tasks, essentially creating a "one-stop shop" for image editing. Third, HILITE iteratively takes user feedback, which both enhances the user experience and enables collection of crowd-sourced data for image editing. HILITE outperforms two major image editing softwares, OpenAI's DALL·E 3 and Google's Imagen 3, across two widely-user quantitative metrics for image editing evaluation. Considering the growing demand for readily-available and high-performing image editing tools, HILITE provides a novel platform design with multifaceted use cases in both business and academia. The platform can be found at https://platform.opennlplabs.org/ or https://platform-deployment.vercel.app/.**

***Index Terms*—NLP, diffusion models, image editing, human-in-the-loop, interactive platform**

I. Introduction

The field of image editing has witnessed significant advancements in recent years, driven by breakthroughs in deep learning and neural network architectures. These developments have led to the creation of increasingly sophisticated image generation and manipulation tools, such as Google's Imagen 3 [1] and Adobe Photoshop AI [2]. However, despite these advancements, existing solutions often face limitations in terms of accessibility, flexibility, and user control.

Traditional image editing software like Adobe Photoshop requires extensive technical expertise, limiting its accessibility to users without graphic design skills. Conversely, AI-powered tools like DALL-E 3 [3] and Imagen, while more user-friendly, often generate entirely new images rather than allowing precise edits to existing ones. Additionally, they have a single model in the backend with the implementation details concealed from the user, and are often gated behind a paywall, restricting free access. Open-sourced alternatives often require extensive prompt-engineering, are highly specialized [4] [5], and lack support for iterative, human-in-the-loop editing processes, which are crucial for achieving desired results [6].

To address these gaps, we introduce the **first open-sourced image-editing platform** that combines the strengths of multiple state-of-the-art diffusion models and large language models (LLMs) in a single, easy-to-use interface. Unlike traditional image-editing tools that require graphic design skills or complex prompt-engineering, our platform allows users to simply enter a text instruction or optionally upload a reference image, and the system will generate multiple outputs to choose from. Users can continue the editing process with their chosen outputs, offering feedback and customizing the images without the need for extensive technical expertise.

Our platform's core strength lies in its ability to handle very general image-editing tasks, leveraging the complementary strengths of different diffusion models. Where other models are often limited to a single function, we allow users to benefit from the combined capabilities of multiple models, making it easier to achieve complex edits. We further eliminate the need for users to engage in complex prompt-engineering, by leveraging LLMs to reformat instructions such that they are best-suited for a given model. Furthermore, our platform's human-in-the-loop design enables users to iteratively refine their images, providing feedback at each stage to ensure

the output meets their exact specifications. This interactive feedback mechanism not only increases user satisfaction but also contributes to the training of future visual language models through crowd-sourced input. Unlike proprietary models which are closed-source and require payment, our platform is entirely open-source, providing users with a powerful, cost-free alternative.

Experimentally, image-editing models typically optimize for two key aspects: a) retaining the structure of the original image, and b) closely following the text instruction [7], [8]. Our results demonstrate that HILITE outperforms proprietary models on both of these metrics, further demonstrating its effectiveness as a general-purpose image-editing tool.

II. Related Work

A. Image-editing Models

The quality of image editing models has significantly improved over time, largely driven by advancements in deep learning, particularly diffusion-based models. Early approaches like Pix2Pix [9] required extensive user input and often produced less realistic results. Models such as SPADE [10] improved semantic manipulation and photorealism, but the real leap came with diffusion models. Denoising diffusion models (DDPM) [11] demonstrated high-fidelity image generation, enabling more complex and precise edits. Techniques like Blended Diffusion [12] and Latent Diffusion [13] further refined the ability to perform localized, text-driven edits and high-resolution image synthesis, shifting the focus from basic adjustments to sophisticated manipulations like object reorientation and style transfer, all while maintaining superior image quality.

However, with high quality generations come the limitations of each model being specialized to perform a specific set of tasks. For example, AnyDoor [14] is trained to teleport objects from one image to another at user-specified locations (provided by masks as input), but performs poorly on tasks like style transfer, reorientation of objects, and so on. A similar model is Paint-By-Example [15], which specifically replaces objects masked out in the original image with a user-uploaded exemplar. Another class of models like DEADiff [16] and StyleAdapter [17] specifically focus on manipulating style in images. Certain models [8] [18] have also been trained to work with generic text instructions and perform well on style transfer, object swapping, and making global changes to the image, but struggle with targeted edits or abstract instructions that require semantic understanding [19]. To increase interactivity and precision in editing over text-instruction models, drag-based models [4] have also been proposed that specifically allow for manipulating specific points in an image by dragging them to a new position, while the model adjusts the surrounding content to maintain realism and consistency. However, these are incapable of adding new content to the image like text-instruction models.

Addressing the above limitations while leveraging strengths of individual models, we propose to build a system for image-editing using open-sourced AI. Our system provides generalist capabilities to manipulate images in a wide variety of ways with a simple text input to a user-friendly interface.

B. Human-in-the-loop Design

AI-based image editing can be applied to the creation of creative media by artists, photographers, and content creators [20]. Additionally, automated image editing platforms allow their artistic means to be more accessible to the general public [21]. Considering the diverse and creative needs of AI tools, researchers have employed human-in-the-loop design to enhance both the accuracy of technology and the effectiveness of human creativity [22], [23]. Therefore, a well-designed interface that caters to user needs is essential in image-editing platforms—both for editing and data collection purposes, where the data can allow for training and calibrating improved image-editing models. In the case of crowdsourcing data for machine learning models, it is important to ensure clarity in instructions to preserve quality of collected data—this may include prompting the crowd-coders for their feedback, or rationale, on the labels they assign to their data [24]. SyntaxGym, as an example of a data collection platform for evaluating neural network language models, includes a "gallery" of existing tests for further visualization and clarity to the user [25]. In turn, the integration of the user's feedback, interaction, and demands heightens the quality of an image-editing platform that serves data collection purposes.

Once data is collected from an interface, it can be used for training and, in particular, calibrating models for improved performance [26]. HQ-Edit is an example of a dataset for instruction-based image-editing, and uses a metric called "Alignment," which compares the model's outputs with the given prompt, to evaluate image-editing performance [27]. However, datasets such as HQ-Edit are limited to their entries and require editing to expand in quantity and diversity of data elements. For the intentions of developing and improving image-editing models, in addition to existing datasets, a data-collection platform that enables the dynamic development of an image-editing dataset would prove useful.

III. Framework

An overall depiction of the framework is shown here. We divide the entire framework into six sequential modules as follows: (1) *Input*; (2) *Intent Detection*; (3) *Running Models*; (4) *Output Selection & Feedback Collection*; (5) *Finish & Publish*; and (6) *Gallery*.

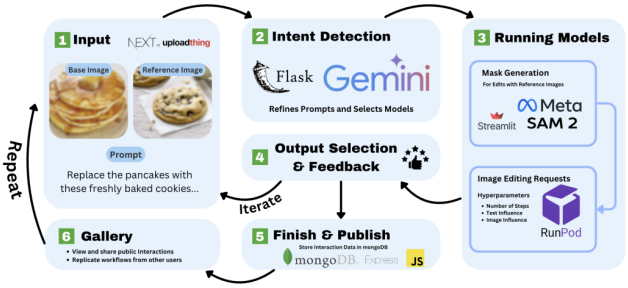

Fig. 1. Diagram of the framework divided into six sequential modules

In a typical user workflow, the user begins by uploading a base image, optionally alongside a reference image, and provides a prompt describing the desired modifications (*Input*). Next, the system processes the prompt and identifies which kind of models to run at the backend, based on the desired changes (*Intent Detection*). The user can then either adjust model parameters or proceed with the default settings to generate the desired image, a process that takes between five to twenty seconds (*Running Models*). After the outputs are generated, the user selects their preferred result, provides feedback, and can review the history of changes made throughout the process (*Output Selection & Feedback Collection*). Once satisfied, the user finalizes the modifications and has the option to publish their work publicly or privately, including all prompts and parameters used (*Finish & Publish*). Finally, the user can explore other published works, filtering them by various criteria such as popularity, date, or keywords, and even replicate the workflows of others (*Gallery*). The details of how each of these modules is implemented are given below:

A. Module 1: Input

The user uploads a base image that they want to modify, and an optional reference image. The reference image is provided for better understanding and control over a target output, but is usually not required for our model to out-perform similar frameworks as compared in Section IV-A. For example, if the prompt says "replace the cat with a specific breed like a Siamese cat," and the user has a particular Siamese cat in mind, they can upload its image. This allows the model to accurately swap the original cat with the specific Siamese cat from the reference.

B. Module 2: Intent Detection

When the user clicks *Translate*, the prompt is run through our Gemini prompt reformatter and intent classifier before being saved in the frontend for our diffusion models. As shown in Figure 1, the prompt reformatter refines the user's prompt to a more elaborate and specific prompt that works best for that specific model, while the intent classifier identifies the user's intent and directs the request to the most suitable models. For example, if the prompt demands stylistic changes, we run models trained to specifically tackle style editing in images. There is a loading queue showing that we are detecting the intent of the user.

We use Gemini 1.5 Flash[1] and a Flask-built endpoint to make the user's prompt more specific for further accuracy in image-editing and intent classification to choose the proper model for the user's request. We use various prompting techniques and place the prompts in a structured LangChain[2] template to make our output more clear.

First, Gemini detects the intent of the user and categorizes it into an 'action', which is simply a type of image editing (object swapping, object insertion, etc.). Along with providing the user's prompt we also input the image. Since Gemini is a Vision-Language Model (VLM), it can semantically understand and connect the objects in the prompt with the image to give truly accurate prompts for the text-based diffusion models.

Then, after classifying the intent of the user, an appropriate ranking of models that are best suited for the user's use case is determined, and Gemini modifies the user's prompt so that it is more tailored towards the task at hand. The prompt can be more specific because the previous intent classification allows us to retain the general idea of the user's prompt, allowing Gemini to do more aggressive prompt editing whilst still maintaining the user's baseline instructions.

C. Module 3: Running models

Before prompting the diffusion models, users have the option to adjust key hyperparameters using interactive sliders that abstract away technical terms. The sliders are intuitive to understand from a user perspective. Up to three hyperparameters can be controlled: number of steps, text influence, and image influence. To remove bias, the intervals between each step are standard across all models. Guidance scale is labeled as text influence in the platform interface; this parameter controls how much the text or mask influences the image generation, i.e. how strictly the model has to follow the given instruction. Control strength is labeled as image influence in the platform interface, which controls how much of the original image should be retained in the generated image.

If a reference image is uploaded in addition to the base image, the models need further knowledge on which specific parts of the reference image need to be placed onto the base image. Hence, we also provide a canvas to the user where they can apply masks on objects in the base and reference images, which will be needed as input for reference-image based models like Paint-by-example [15] and AnyDoor [14].

The diffusion models are provided with the selected parameters, images, prompts, and/or masks. We then send requests from our NextJS frontend to our RunPod endpoints in parallel, using JavaScript Promises to handle them asynchronously, and display the results once all outputs are received.

The mask data is edited using a Streamlit drawable canvas and passed in to the models that require masks to pinpoint the location of the desired changes.

[1]https://ai.google.dev/gemini-api/docs
[2]https://python.langchain.com/docs/introduction/

All of the model endpoints are deployed on Runpod's [3]serverless services. This ensures that the platform has a minimal queue time and always has GPU instances available on the server even if zero users are using the platform, nullifying the issue of cold start times.

D. Module 4: Output Selection & Feedback Collection

Based on the input and detected intent, we display multiple model outputs to the user and they can select the one that most accurately represents their desired intent. We also collect feedback from the user about: *a)* how closely the image matches their expectations; *b)* how closely the image structure matches the base image; *c)* whether the image is offensive (including a subjective description); *d)* written feedback about anything else they might want to share. This feedback can be used in the future to improve end-to-end image editing models.

Users can repeat this process as many times as necessary, iteratively fine-tuning and adjusting the image until they are satisfied with the result. To minimize bias in model selection, the output images are presented in a randomized order, and the model names are anonymized. All data related to the user's actions, including their original prompt, hyperparameter settings, and comprehensive feedback, is stored in a MongoDB database formatted in JSON for ease of organization and retrieval.

E. Module 5: Finish / Publish

Each time a user makes one edit, it is recorded as a 'Step' and added to the sidebar to track the editing history. This allows the user to revisit previous steps or revert to an earlier version if they are dissatisfied with their current edit. Once a user is satisfied with their image, they can publish it in public or private mode. The user can add a title or description to the image to contextualize the image in case they want to display it to the public. Although the prompt is modified inside of the intent classifier, the original prompt is displayed inside of the gallery. The publish panel is located on the bottom right of the default platform screen, and consists of several fields, including title, public and private radio buttons, and a dropdown to indicate overall satisfaction. Our backend stores and transmits all data recorded in the interaction to our MongoDB database for future reference.

F. Module 6: Gallery

The gallery showcases images that users choose to publish, displaying the original image alongside the final modified version. Each image edit in the gallery includes a complete history of every step's prompt and the precise numerical hyperparameters used during the translation process, allowing for a high level of reproducibility. However, due to the stochastic nature of the models, some randomness is involved in the generation, meaning that while the results are highly consistent, they may not be exactly identical across different runs.

[3]https://docs.runpod.io/serverless/overview

Our gallery is equipped with a search bar to filter posts by keywords and a dropdown filter to filter posts by number of likes or date. When a user clicks a post on the gallery, a modal is opened where the user can review all the images, prompts, and parameters that resulted in the output image. The MongoDB database handles image and feedback data through a standardized JSON format that can be easily pulled and displayed inside the gallery, ensuring full transparency for all publicly edited images.

G. Models

We use a collection of six diffusion-based models, 4 text-guided and 2 mask-guided.

- *InstructPix2Pix*[4]: InstructPix2Pix is an image editing model specifically trained to make edits on images using human instructions. It is a latent-diffusion model trained using a unique process which utilizes GPT-3 to create prompt input and output captions pairs along with editing instructions. Using GPT-3 allows for the creation of a large dataset consisting of 450k images and a variety of editing instructions. Additional input channels are added into the first convolutional layer of the latent diffusion model to allow introduction of two guidance scales, S_T and S_I that control how well the sample images correspond to the input image, and how strongly the output images follow the instructions. Ablations was created utilizing 10% and 1% of the dataset and the original trained diffusion model yielded the highest trade-off between cosine similarity and text-image direction similarity (how much the change in text captions agrees with the change in the images) of CLIP embeddings of text and image pairs, indicating the large dataset allows the model to be versatile.
- *DEADiff*[5]: DEADiff is a novel style-transfer model that generates images while preserving the original semantic landscape. It uses a CLIP-based image encoder, a Q-Former, and a U-Net. The Q-Former extracts key style elements from an input image, which are then passed to the U-Net. The U-Net employs a Disentangled Conditioning Mechanism (DCM), conditioning coarse layers on semantics and fine layers on style to maintain intricate details. Additionally, the paper devises a joint text-image cross-attention layer that concatenates the key and value matrices from text and image features, initiating a single cross-attention operation with the U-Net query features. The model trains on two tasks: one for extracting style from images of the same style and another for extracting content from the target image and its caption.
- *Inversion-free Image Editing*[6]: Inversion-free or InfEdit is a general image manipulation model aimed to provide fast generation times. The quicker generation times are due to the replacement of the time-consuming inversion

[4]https://www.timothybrooks.com/instruct-pix2pix
[5]https://tianhao-qi.github.io/DEADiff/
[6]https://sled-group.github.io/InfEdit/

process found in popular Stable Diffusion models with a Denoising Diffusion Consistent Model (DDCM). It also presents a Unified Attention Control (UAC), a tuning-free method that unifies attention control mechanisms for text-guided editing. The combination of both of the mechanisms allows text-guided image manipulation while keeping the original image's details consistent. The UAC is a combination of a hybrid self-attention and cross-attention mechanism with an additional layout branch as an intermediate to host the desired composition and structural information in the target image.

- *ControlNet with Stable Diffusion XL*[7]: ControlNet with Stable Diffusion XL - ControlNet is an image editing model that allows for explicit control over the features of the generated image by providing auxiliary control inputs. ControlNet auxiliary inputs are preprocessed versions of the input image that preserve a given subset of features from the image. Canny edge detection, for example, preserves structural features like edges. These auxiliary control inputs are passed through a conditional branch parallel to the backbone (Stable Diffusion XL). The features are fused in a U-Net architecture across various layers, transforming the output image with structural fidelity. This makes it ideal for effective image editing, where maintaining key visual elements while adapting style and content is crucial.
- *AnyDoor*[8]: AnyDoor is a mask-based image editing model that specializes in zero-shot object-level customization, allowing users to seamlessly place objects from reference images into new scenes with high fidelity and feature consistency. It uses identity tokens and detail maps to capture fine details and key characteristics of objects, which are injected during the diffusion process to preserve important object features. AnyDoor supports tasks such as object swapping, multi-object composition, and shape editing, without the need for parameter tuning.
- *PaintByExample*[9]: PaintByExample is a mask-based image editing model that excels in inserting objects and altering target images by integrating features from reference objects. The model accomplishes this through the use of an "Information Bottleneck" which condenses the reference image into a compressed representation. This forces the model to focus on essential semantic information and interpret it in terms of the target mask and image, which allows it to insert the intended features.

IV. Applications

A. Results

Image-editing models typically report results on primarily one quantitative metric to demonstrate the capabilities of their model: *image-image similarity*. True comprehensive editing involves editing certain regions while keeping other regions relative the same. Even in image-wide edits such as the action of style transfer, requires keeping the objects' properties relatively (composition, position, luster, etc) the same. It is common to use CLIP Embeddings and the SSIM (Structural Similarity Index Measure) metric. The SSIM metric uses edge detection on two images and compares those edges overall to measure the structural similarity of the image. It's ability to use edges for comparison allows finding differences in images that do not meet the human eye. The use of CLIP Embeddings allows the evaluation of edits on a conceptual level and seeing if regions in the output image maintain the same human-defined relationship as the base image. In our experimental setup, we use OpenAI's DALL·E 3, Google's Imagen 3, and HILITE to edit a set of images. Next, we compute *image-image similarity* of outputs from all three models using CLIP and SSIM. We find that our platform significantly outperforms both models. Specifically, we see a gain of 2.67% on cosine similarity and 23.53% on SSIM in respect to Imagen 3. We also see a gain of 2.67% on cosine similarity and 28.57% on SSIM in respect to DALL·E-3. The higher SSIM indicates our model can make specific edits while keeping the semantic landscape.

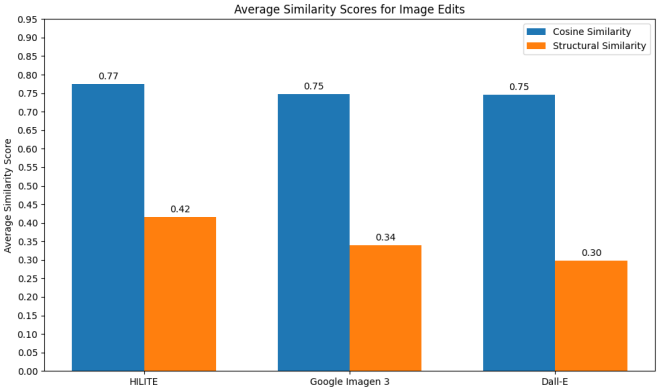

Fig. 2. Results

B. Use Cases

Our work has various applications for commercial and business creative image editing. As a whole, the platform serves as a high-performance tool for general image-editing instructions. It overcomes the specialization of existing individual image-editing models: for example, InstructPix2Pix struggles to process natural language in the editing instructions. The combination of intent-detection via Gemini and the use of multiple image-editing models enables the platform to be used for generic edits or more abstract editing instructions, including style-swapping, orientation-changing, and object-swapping.

In turn, the platform sees indirect applications for the typical uses of image-editing. These include creative media, such as digital art, photography, and graphic design. They also serve broader practical purposes, such as marketing/advertisements and educational content creation.

[7]https://arxiv.org/abs/2302.05543
[8]https://arxiv.org/abs/2307.09481
[9]https://arxiv.org/abs/2211.13227

Finally, a crucial component of our platform was the collection of user feedback, which enables the assemblage of a parallel dataset of high-quality edited images for future training of image-editing models. The feedback can also be used to evaluate individual models for applications in model-calibration to improve performance.

V. Conclusion & Future Work

In this paper, we present a high-quality image-editing platform with data collection capabilities due to the retrieval of user feedback. We employed a collection of six state-of-the-art diffusion models for image-editing within our platform and utilized the Gemini VLM for detection of user intent. The interface of our platform has been systematically designed to add the "human-in-the-loop" to automated image editing, as the user's input is considered from end-to-end.

There are numerous possibilities that we aim to achieve with our platform. First, our platform is completely open source, so other like-minded developers can continue to improve and add features to the platform. For example, models can be removed or added, allowing our platform to stay up-to-date with the latest diffusion models.

Additionally, due to our platform's data collection feature, our platform will be used to fine-tune a VLM to create a generic image editing model based on human feedback to route more effective requests to the different diffusion models. Based on human preferences and flaws of current image editing models, we will create a effective image transcreation pipeline utilizing intent and object detection to modify images more accurately than current diffusion models due to our platform's human feedback. Based on the tuned hyperparameters from the feedback, our pipeline will have minimal human interaction and will be completely autonomous, revolutionizing image editing. Since we also collect individualized model feedback, the current data collection could be used to fine-tune or create Low-Rank Adaptation (LoRA) models for the diffusion models themselves to be more effective on their specific image editing tasks.

References

[1] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022.

[2] Adobe Inc., "Adobe photoshop." [Online]. Available: https://www.adobe.com/products/photoshop.html

[3] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[4] P. Ling, L. Chen, P. Zhang, H. Chen, Y. Jin, and J. Zheng, "Freedrag: Feature dragging for reliable point-based image editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6860–6870.

[5] Y. Deng, F. Tang, W. Dong, C. Ma, X. Pan, L. Wang, and C. Xu, "Stytr2: Image style transfer with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 326–11 336.

[6] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, "Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation," in *2018 IEEE conference on computational intelligence and games (CIG)*. IEEE, 2018, pp. 1–8.

[7] T.-J. Fu, W. Hu, X. Du, W. Y. Wang, Y. Yang, and Z. Gan, "Guiding instruction-based image editing via multimodal large language models," *arXiv preprint arXiv:2309.17102*, 2023.

[8] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 392–18 402.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[10] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[12] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 208–18 218.

[13] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[14] X. Chen, L. Huang, Y. Liu, Y. Shen, D. Zhao, and H. Zhao, "Anydoor: Zero-shot object-level image customization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6593–6602.

[15] B. Yang, S. Gu, B. Zhang, T. Zhang, X. Chen, X. Sun, D. Chen, and F. Wen, "Paint by example: Exemplar-based image editing with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 381–18 391.

[16] T. Qi, S. Fang, Y. Wu, H. Xie, J. Liu, L. Chen, Q. He, and Y. Zhang, "Deadiff: An efficient stylization diffusion model with disentangled representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8693–8702.

[17] Z. Wang, X. Wang, L. Xie, Z. Qi, Y. Shan, W. Wang, and P. Luo, "Styleadapter: A single-pass lora-free model for stylized image generation," *arXiv preprint arXiv:2309.01770*, 2023.

[18] S. Xu, Y. Huang, J. Pan, Z. Ma, and J. Chai, "Inversion-free image editing with natural language," *arXiv preprint arXiv:2312.04965*, 2023.

[19] S. Khanuja, S. Ramamoorthy, Y. Song, and G. Neubig, "An image speaks a thousand words, but can everyone listen? on translating images for cultural relevance," *arXiv preprint arXiv:2404.01247*, 2024.

[20] J. Bailey, "The tools of generative art, from flash to neural networks," *Art in America*, vol. 8, p. 1, 2020.

[21] F. Zhan, Y. Yu, R. Wu, J. Zhang, S. Lu, L. Liu, A. Kortylewski, C. Theobalt, and E. Xing, "Multimodal image synthesis and editing: The generative ai era," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 15 098–15 119, 2023.

[22] R. Munro, *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Manning, 2021.

[23] N. Pangakis and S. Wolken, "Keeping humans in the loop: Human-centered automated annotation with generative ai," *arXiv preprint arXiv:2409.09467*, 2024.

[24] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.

[25] J. Gauthier, J. Hu, E. Wilcox, P. Qian, and R. Levy, "Syntaxgym: An online platform for targeted evaluation of language models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020, pp. 70–76.

[26] J. Vaicenavicius, D. Widmann, C. Andersson, F. Lindsten, J. Roll, and T. Schön, "Evaluating model calibration in classification," in *The 22nd international conference on artificial intelligence and statistics*. PMLR, 2019, pp. 3459–3467.

[27] M. Hui, S. Yang, B. Zhao, Y. Shi, H. Wang, P. Wang, Y. Zhou, and C. Xie, "Hq-edit: A high-quality dataset for instruction-based image editing," *arXiv preprint arXiv:2404.09990*, 2024.