Efficiently teaching counting and cartoonization to InstructPix2Pix

Amardeep Kumar
New York University
amardeep.kumar@nyu.edu

Rahul Raman
New York University
rr4549@nyu.edu

Ritika Saboo
New York University
rss9311@nyu.edu

Abstract

In this project, we explored the capabilities of InstructPix2Pix, a diffusion-based model designed for interpreting human-written instructions in image editing. Through the integration of LoRA techniques, our primary focus was the fine-tuning of InstructPix2Pix to address specific challenges in object counting and image cartoonization. Our contributions encompass successful adaptation to LoRA integration, curation of a specialized counting dataset for precise object editing, and the improvement of InstructPix2Pix's performance in Cartoonization by addressing and rectifying its inherent limitations in this domain. Our implementation can be found here.

1. Introduction

As models continue to grow in size, full fine-tuning on everyday-hardware becomes impractical. The challenge is compounded by the escalating costs of storing and deploying finely-tuned models for each specific task, as these models maintain the same size as the originally pretrained model. To tackle these issues, Parameter-Efficient Fine-tuning (PEFT) approaches have emerged. LoRA (Low-Rank Adaptation of Large Language Models) stands out as a technique designed to facilitate fine-tuning of large models even on limited GPU resources.

While generating images using diffusion models is a common task, our interest was piqued by the idea of editing images based on specific instructions using a diffusion model. Enter InstructPix2Pix [1] — a diffusion-based model capable of interpreting human-written instructions to edit a given image. Leveraging LoRA techniques, we undertook the task of fine-tuning InstructPix2Pix on two specific challenges.

In selecting our tasks, we recognized InstructPix2Pix's proficiency in instruction-based image editing while acknowledging its limitations. We identified two key areas where the model exhibited shortcomings—object counting and image cartoonization—and set out to address and enhance these aspects through this project.

In essence, our approach involves the efficient fine-tuning of InstructPix2Pix using LoRA for tasks related to Cartoonization and Counting. Our contributions are outlined as follows:

1. We successfully adapted InstructPix2Pix to integrate with the LoRA technique.
2. We curated a new counting dataset designed for editing instructions, enabling the fine-tuning of InstructPix2Pix to accurately edit the specified number of objects.
3. We enhanced the performance of InstructPix2Pix in the Cartoonization task, addressing and rectifying its limitations in this domain.

[1]

2. Related Work

2.1. Low-Rank Adaptation of Large Language Models

Microsoft researchers have introduced a novel technique called LoRA (Low-Rank Adaptation of Large Language Models) [3] to address the challenges associated with fine-tuning large language models. Fine-tuning models with billions of parameters, such as GPT-3, is cost-prohibitive. LoRA suggests a solution by preserving pre-trained model weights and introducing trainable layers (rank-decomposition matrices) into each transformer block. This significantly reduces the number of trainable parameters and GPU memory requirements, as gradients do not need to be computed for most model weights. The researchers discovered that by specifically targeting the Transformer attention blocks of large language models, fine-tuning quality using LoRA matched that of full model fine-tuning, but with increased speed and reduced computational demands.

While LoRA was initially introduced for large-language models and showcased in transformer blocks, its applicability extends beyond these scenarios. In the context of

[1] All authors contributed equally.

Stable Diffusion [4] fine-tuning, also in case of InstructPix2Pix, LoRA can be effectively employed in the cross-attention layers responsible for connecting image representations with the corresponding descriptive prompts.

2.2. InstructPix2Pix

InstructPix2Pix [1] effectively follows written instructions provided alongside input images to perform edits. To address the challenge of acquiring sufficient training data, the researchers leverage the knowledge of two pre-trained models, GPT-3 (a language model) and Stable Diffusion (a text-to-image model), resulting in the generation of a substantial dataset of image editing examples. InstructPix2Pix, a conditional diffusion model, is trained on this dataset and demonstrates the ability to generalize to real images and user-written instructions during inference without the need for fine-tuning or inversion, enabling quick image edits in a matter of seconds. The model exhibits compelling results across diverse input images and written instructions, showcasing its versatility in tasks such as object replacement, style changes, setting adjustments, and modifications to artistic mediums.

In this we overcome two major problems of intruct-Pix2Pix by training it efficiently Using LoRA techniques

2.3. White-box-Cartoonization

White-box-Cartoonization [5] paper introduces an image cartoonization method based on observations of cartoon painting behavior and consultations with artists. It proposes identifying three white-box representations—surface, structure, and texture—from images. Using a Generative Adversarial Network (GAN) [2] framework, the method learns these representations to cartoonize images. The learning objectives are separately tailored to each representation, providing controllability and adjustability. The approach meets artists' requirements across styles and use cases. Comprehensive evaluations, including qualitative and quantitative analyses and user studies, confirm the method's effectiveness, surpassing previous approaches. Additionally, an ablation study illustrates the influence of each component in the framework.

Cartoonization datasets used by us leverages this technique - along with chatGPT to generate prompts - to generate cartoonization images and edit instruction.

2.4. Counting data generation using generative models:

Deep learning models typically demand extensive training data, and while internet data collections are often available, they might not align with the required format for supervision, such as paired data of specific modalities. With the ongoing enhancement of generative models, there is a growing interest in leveraging them as a cost-effective and abundant source of training data for subsequent tasks. In our approach, we employed Stable Diffusion to generate datasets for object counting. To transform this dataset into an image editing dataset, we directed the diffusion model to generate on a plain background. This allowed us to present an empty canvas to our InstructPix2Pix model, prompting it to add a specified number of objects. The images generated by Stable Diffusion serve as reliable ground truth for this process.

3. Method

3.1. Injecting LoRA weights into ImagePix2Pix

LoRA falls under the category of "parameter-efficient" (PEFT) techniques, aiming to minimize the impact on the number of trainable parameters during fine-tuning. Its objective is to enhance fine-tuning speed while concurrently reducing the size of the fine-tuned checkpoints.

Rather than making minute adjustments to all the model's weights during fine-tuning, our approach involves the selective freezing of most layers. We focus on training only specific layers within the attention blocks, and to avoid altering the parameters of these layers, we introduce the product of two smaller matrices to the original weights. The weights of these smaller matrices are the ones modified during the fine-tuning process and subsequently saved to disk. Consequently, the model's original parameters remain intact, and the LoRA weights can be loaded on top using an adaptation method. This method ensures the preservation of the model's integrity while achieving efficiency in fine-tuning.

The technique constrains the rank of the update matrix ΔW using its rank decomposition. It represents ΔW_{nk} as the product of 2 low-rank matrices $B_{nr}$and $A_{r,k}$ where $r \ll min(n, k)$. This implies that the forward pass of the layer, originally Wx, is modified to $Wx + BAx$ as shown in the figure 1. A random Gaussian initialization is used for A and B is initially to 0, so $BA = 0$ at the start of training. The update BA is additionally scaled with a factor α/r.

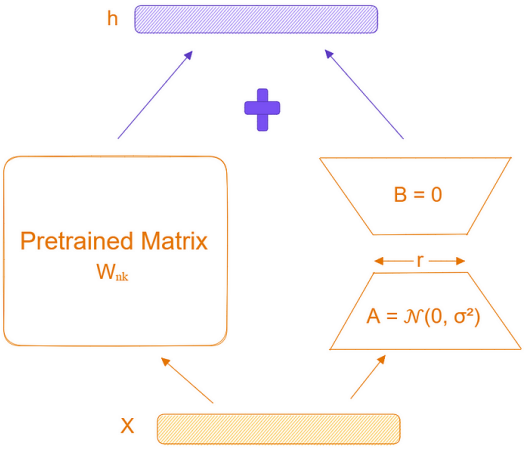


Figure 1. Injecting LoRA weights

Due to the reduced ranks of matrices A and B, the LoRA adapter exhibits a notably diminished weight in comparison to the overall model size. Consequently, the loading process is significantly expedited when compared to the retrieval of the entire base model.

To enhance the efficiency of the fine-tuning process for ImagePix2Pix on limited GPU resources, we integrated LoRA matrices into InstructPix2Pix. In our specific application, given that ImagePix2Pix functions as a Conditional Diffusion model, we found that LoRA can be effectively applied within the cross-attention layers responsible for connecting image representations to corresponding descriptive prompts. A LoRA matrix was specifically designed for the cross-attention layer (comprising query, key, and value matrices) of InstructPix2Pix, as illustrated in Figure 2. During optimization, only LoRA parameters are passed to our optimizer, while the previously pretrained weights remain fixed to mitigate the risk of catastrophic forgetting.

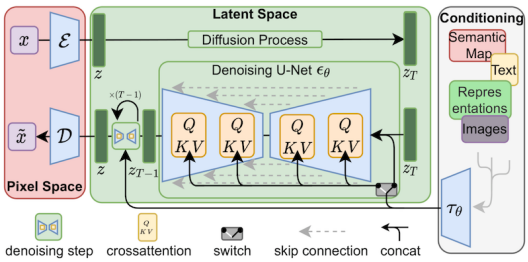


Figure 2. Attention layers of Stable Diffusion model.

The dimensions of these LoRA matrices are substantially smaller compared to the original weights, resulting in a significantly reduced number of trainable parameters. This characteristic contributes to heightened training efficiency. Furthermore, the Rank-decomposition matrices exhibit a marked reduction in parameters compared to the original model, rendering the trained LoRA weights highly portable.

3.2. Counting datasets

Many generative image models encounter challenges when it comes to accurately counting the objects they generate, often failing to produce the correct count as requested. In the case of InstructPix2Pix, which is based on a stable diffusion model and trained on synthetic data, the issue of hallucinating object counts becomes inevitable.

To address this concern through fine-tuning InstructPix2Pix, a prerequisite is a high-quality dataset containing the accurate count of objects. However, our search for such datasets yielded no relevant results. While some datasets contained counts of specific objects, they were unsuitable for our use case, which required the incorporation of edit instructions. Specifically, we needed a dataset where an input image with 'x' number of cars, with an edit instruction to add 'y' more cars, would yield an output image with 'x+y' cars.

To create this dataset, we implemented a pipeline leveraging stable diffusion to generate images with a specified object count on a plain background. To ensure dataset quality, we employed various object detection models to extract information on both object count and type. Instances of disagreement between models were resolved through human intervention, and random checks on a subset of generated images were conducted for additional verification. Flow of our pipeline is showed in Figure 7

Choosing a plain background for image generation served two purposes: firstly, object detection models are more accurate when images are on plain backgrounds, simplifying the detection task. Secondly, these plain background images could serve as input for our model with instructions to add a specific count of a certain object, aligning with the requirements of our generated datasets. We generated 1200 datasets and edit instruction of them.

3.3. Catoonization datasets

The dataset employed in our study, originated from the random selection of 5000 images within the Imagenette dataset. Utilizing a white-box-cartoonization model, the sampled images were transformed into their cartoon equivalents, facilitated by ChatGPT to generate prompts. The entire workflow is visually represented in the Figure 3 . We use this dataset to finetune ImagePix2Pix on cartoonization task.

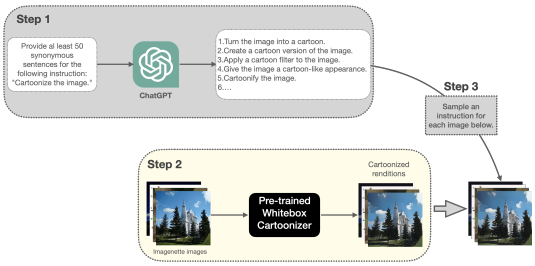


Figure 3. Pipeline for Cartoonization datasets

4. Experiments

4.1. Efficiency of LoRA adapated InstructPix2Pix

We incorporated the Hugging Face library as the foundation of our codebase. To enhance the InstructPix2Pix model, we made modifications to the Query, Key, and Value Matrices of each attention layer. This involved breaking each matrix into two separate matrices, namely A and B. These LoRA decomposed matrices were exclusively passed to our optimizer function, ensuring their trainability. Subsequently, we employed this framework to fine-tune the LoRA-infused InstructPix2Pix model on tasks related to counting and Cartoonization. In Table 1, we present a comparative analysis of two cases, highlighting the efficiency gained through LoRA techniques.

The results depicted in the table unequivocally demonstrate that the integration of LoRA significantly improved the efficiency of InstructPix2Pix training, rendering the fine-tuned weights easily transferable.

4.2. Cartoonization tasks

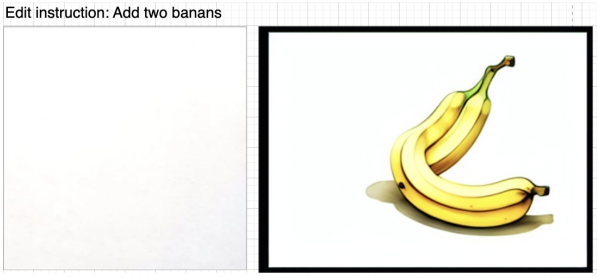
We conducted fine-tuning for 100 epochs on Cartoonization tasks using our LoRA-adapted InstructPix2Pix, setting the learning rate to 5e-5 and the batch size to 32. The training loss curve, illustrating the disparity between the original noise and the noise predicted by the diffusion model, is presented in Figure 6

Qualitative Analysis: As this task involves image generation, quantitative metrics are not applicable. Therefore, we provide a qualitative analysis in Figure 4. The samples clearly demonstrate a substantial improvement in the quality of Cartoon generation after fine-tuning. In *Figure X*, the model accurately cartoonizes intricate patterns, such as marbles on the surface of the Taj Mahal, and effectively captures reflections in water channels. In contrast, the original InstructPix2Pix struggled not only with intricate details but also with the fundamental concept of cartoonization, as evident in other samples.

4.3. Counting tasks

We conducted training for 100 epochs on Counting tasks using our LoRA-adapted InstructPix2Pix, employing a learning rate of 5e-5 and a batch size of 32. Despite experimenting with various hyperparameter tuning approaches, we found limited success in significantly improving the object counting ability of InstructPix2Pix. Comparative ex-

Figure 4. Visual result on Cartoonization Task: 1. Input Image, 2. Output by baseline InstructPix2Pix, 3. Output by fintuned Model. **Edit Instruction** - Apply a cartoon-like filter to the natural image to give it a toon-like appearance.



(a) A case where fine tuned InstructPix2Pix works well.

(b) one of the cases where fine tuned InstructPix2Pix struggles with counting.

Figure 5. Visual result on Counting tasks

Parmeters	InstructPix2Pix	LoRA-InstructPix2Pix
Weight Matrix size of attention layer	1280 x 1280(m xn)	Size of A Matrix: 1280 x 4 (m x 4) Size of B matrix: 4 x 1280(4 x n)
number of trainable parameters	859, 532, 484	797, 184
Size of portable - finetuned weight on disk	4.1 GB	3.08 MB
Time for 1 iteration [T4-GPU Google Colab]	150 seconds	0.23 seconds
Max batch size [T4-GPU Google Colab]	1	32

Table 1. Table Compairing efficiency of InstructPix2Pix v/s LoRA adapted InstructPix2Pix

amples are provided in Figure 5

Qualitative Analysis: Our qualitative examination reveals that our model adeptly generates the correct count of objects on a plain canvas. However, challenges arise when instructed to remove a specific count of one object and add a different count of another object. This nuanced aspect reflects the limitations in certain counting scenarios.

5. Discussion

We achieved successful fine-tuning of InstructPix2Pix for Cartoonization and Counting tasks, obtaining excellent results in both areas. However, challenges surfaced in complex counting scenarios where the model struggled when instructed to remove certain objects and add others. As part of future work, we aim to explore the integration of LoRA weights into CLIP layers of InstructPix2Pix. This experiment seeks to evaluate whether fine-tuning in this context enhances the counting ability of InstructPix2Pix.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 1, 2

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 1

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[5] Xinrui Wang and Jinze Yu. Learning to cartoonize using white-box cartoon representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8087–8096, 2020. 2

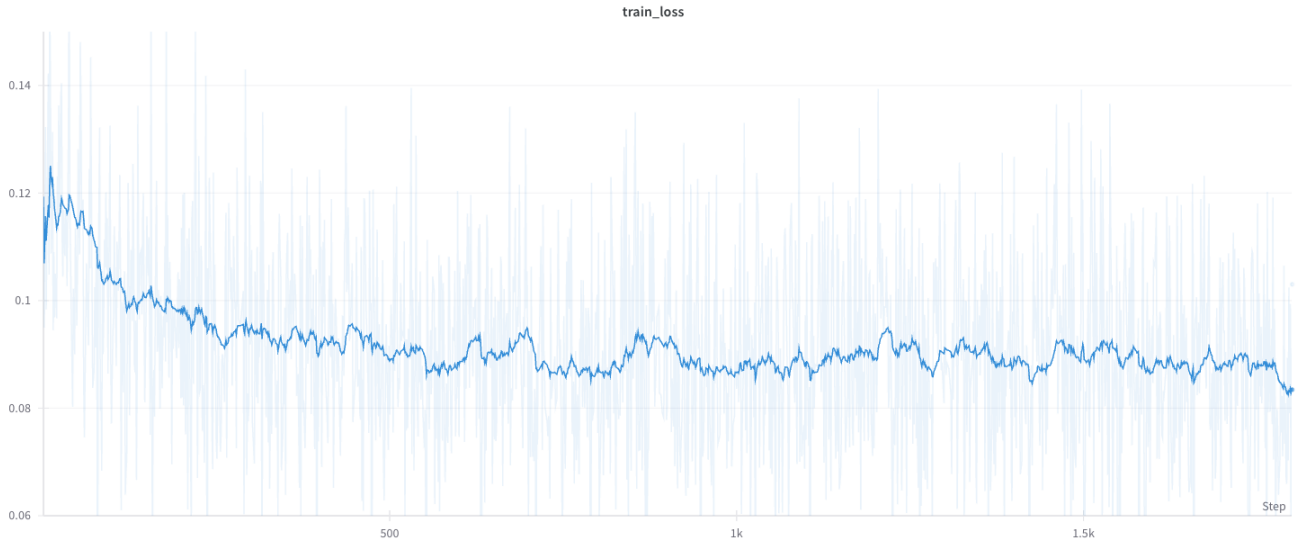


Figure 6. Training Loss curve

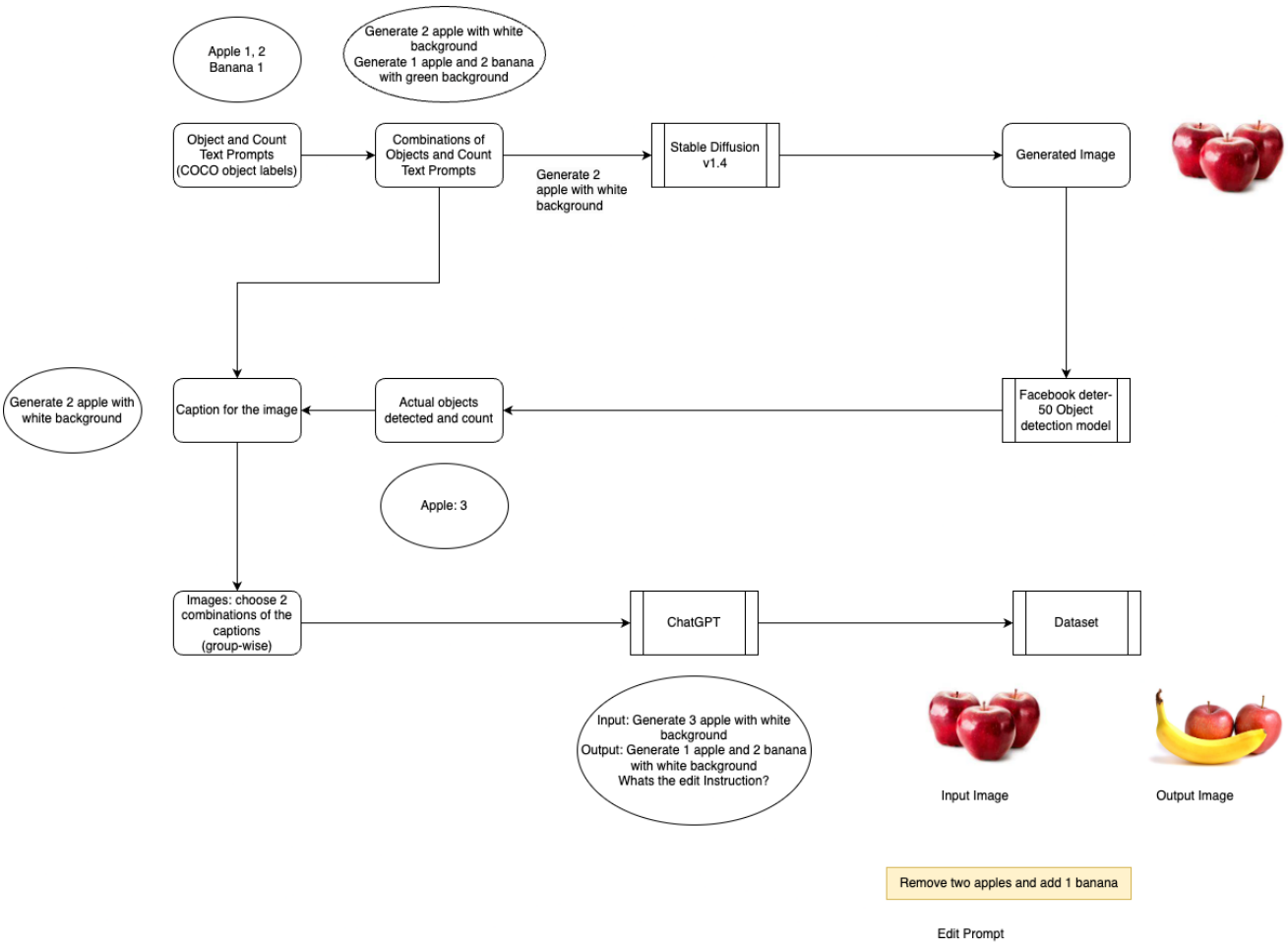


Figure 7. Pipeline for Counting Dataset generation.