A Transformer-Based Generalization Pipeline for Inpainting Models

Kiarash Joolaei[1], Mehrshad Taji[2], Arad Mahdinezhad[2] and Ali Akbari[2]

[1]Computer Engineering Department, Sharif Uninversity of Technology
[2]Electrical Engineering Department, Sharif Uninversity of Technology

Abstract

Generic Inpainting Models have shown impressive results throughout the years since the rise of Deep CNNs. Though, this progress has not come without its shortcomings. These challenges consist of three main problems: ***Robustness*** *,* ***Stereo Consistency in Video Inpainting*** *and* ***Generalization over Semantics****. The first two issues have been resolved in the past years, while the last one still remains; the reason being that inpainting models have no mechanism to truly "understand" the context of image background, and for that reason, the inpainted region might have low cohesion with its background. With this insight, inpainting models can perform much better when guided with text. This has led to the use of Conditional Generative Models in inpainting such that the text input is embedded and used as the latent condition. Now, with the rise of transformers, text generation has become a feasible, high-quality feat. In this paper, we take advantage of this fact and develop a transformer-based architecture that informs the inpainting model of the surrounding background by text generation. We will show that this architecture can improve the quality of inpainted outputs when dealing with different semantics and is a step in the right direction for boosting generalization.*

Introduction

Traditional image inpainting aims to fill the missing area in images, conditioned on surrounding pixels. This task has been around the corner even before deep neural networks gained public attention. Deep generative models such as GAN[1] and DDPM[2] were massive breakthroughs in Deep Learning and have shown promising results in various tasks including image generation, style transfer, video prediction, etc. Diffusion models were also adapted in image inpainting by replacing the random noise in the background region with a noisy version of the original image during the diffusion reverse process, and this method managed to outperform classical inpainting methods by a large margin. Later, transformer and attention-based models such as Stable Diffusion[3] were introduced to develop inpainting and image generation tasks even further.

Considering the advancements, transformer-based inpainting faced several challenges. Elharrouss et al. [4] in their survey found the main challenges of these models to be **Robustness** , **Stereo Consistency in Video Inpainting** and **Generalization over Semantics**. Li et al. [5] developed a transformer-based architecture that addressed the robustness of these types of models and succeeded in generating high-quality inpainted images with large masks. In addition, [6] made a remarkable attempt to design a completely transformer-based video inpainting model with temporal consistency. Despite these incredible achievements, generalization still remains a challenge as inpainting models cannot observe and comprehend the background of the image in a literal sense.

One way to address this problem is to use text input to guide the inpainting model towards a more reasonable output. In fact, this helps and results in images that align more with the background. It is noteworthy that this happens only if the input text is actually *related* to the image. Conditional Diffusion Models such as Classifier Guidance [7] and Classifier Free Guidance [8] have contributed greatly to the generation of labeled inputs such as images conditioned on texts.

To address the generalization issue, we employ text guidance and try to give the model knowledge about the mask's surroundings using generated text. This text is generated by a transformer that is tasked with captioning images. In the next section, we will delve deeper into the details of our pipeline.

Methods

1. Background Context Extraction via Captioning

Raw inpainting models are often trained by randomly masking different regions of the input image and trying to fill out the mask. This method highly biases the model towards the context of the dataset and if this dataset lacks diversity in terms of semantics, the model is doomed to fail in generalization.

Another issue these models face, is that they do not have a deep understanding of the semantic context of

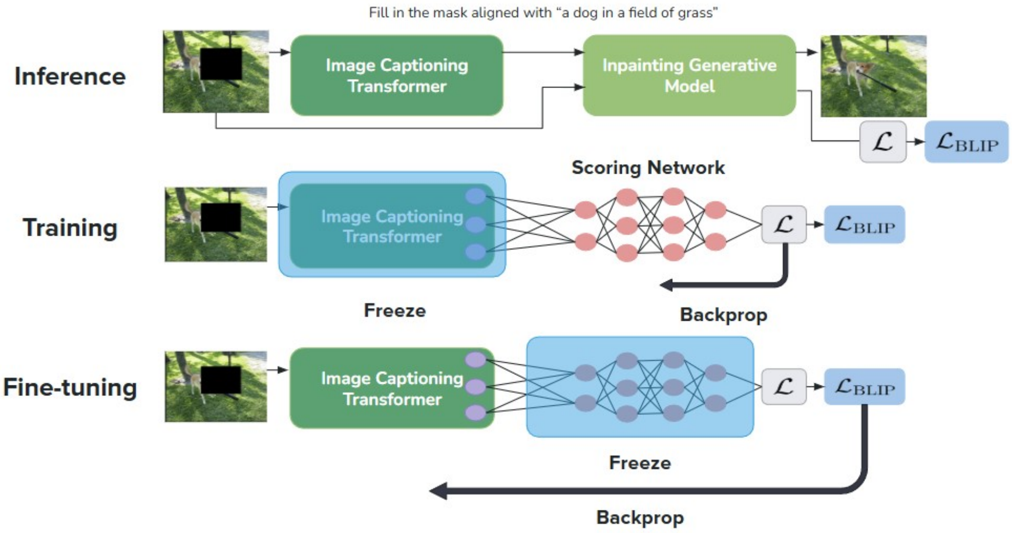
Fill in the mask aligned with "a dog in a field of grass"
Inference
Image Captioning Transformer
Inpainting Generative Model
$\mathcal{L}$
$\mathcal{L}_{\text{BLIP}}$
Scoring Network
Training
Image Captioning Transformer
$\mathcal{L}$
$\mathcal{L}_{\text{BLIP}}$
Freeze
Backprop
Fine-tuning
Image Captioning Transformer
$\mathcal{L}$
$\mathcal{L}_{\text{BLIP}}$
Freeze
Backprop

Figure 1: *Our pipeline consists of three main phases. The first phase requires inference using our model in order to calculate output metrics $\mathcal{L} = \{\mathcal{L}_{CLIP}, \mathcal{L}_{PSNR}, \mathcal{L}_{SSIM}\}$. The next phase uses the information gained from the previous phase to train the scoring network that estimates the mapping between input image and inference scores. In the last phase, after the network is trained, the transformer is fine tuned on this task using the loss function $\mathcal{L}_{BLIP}$.*

the background in the masked image. In this case, conditional generative models come to aid and labeling using text prompts can guide them towards significantly better results. However, we want the inpainting model to perform well on its own as well.

To address this problem, we utilized BLIP Image Captioning model [9]. This model generates captions based on the semantics of the input image. In our architecture this model outputs text, describing the context of the unmasked background in the input image. This text is later used as the conditional label in the generative model. To prevent misconceptions, the text is actually generated during inference and adds a level of explainability to the model which is a major advantage. However, when fine-tuning the transformer, the outputs of the second to last layer of the transformer are used as input to the next layers of the pipeline instead of readable plain-text, essentially using the text embedding. This will be further discussed in the following sections.

2. Transformer Fine Tuning

In order to boost the generalizability of the model further as a whole, we can also fine-tune BLIP on this specific defined task. This fine-tuning does not add additional bias to the model towards a certain direction as the task this transformer is involved in will not change and its only purpose is to generate texts that better describe the background in order to help the inpainting model. The loss used for this regard can be written as shown below:

$$\mathcal{L}_{\text{BLIP}} = \lambda_1(1 - \mathcal{L}_{\text{SSIM}}) - \lambda_2\mathcal{L}_{\text{PSNR}} - \lambda_3\mathcal{L}_{\text{CLIP}}$$

In which $\lambda_3 \gg \lambda_2, \lambda_1$ and each loss term is defined in the following way:

$$\mathcal{L}_{\text{SSIM}}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

SSIM is a perceptual metric used to measure the similarity between images x and y. In this case, the loss is calculated between the ground-truth image and the inpainted image.

$$\mathcal{L}_{\text{PSNR}}(I, K) = 10\log_{10}\frac{(\max I(i,j))^2}{MSE}$$

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(I(i,j) - K(i,j))^2$$

Here, I is the original image and K is the respective inpainted image. This metric measures image quality, particularly in comparing a compressed or reconstructed image to its original version.

Lastly, the CLIP Score is defined as :

$$\mathcal{L}_{\text{CLIP}}(I, C) = \max(A\cos(E_I, E_C), 0)$$

This score evaluates the similarity between the image I and the generated caption C. E_I and E_C are the images' respective embeddings.

3. Scoring Network

Consider the following naive approach when fine-tuning: In this phase, a text is generated regarding the background of the masked image, and then this text is fed to the generative model to generate the inpainted

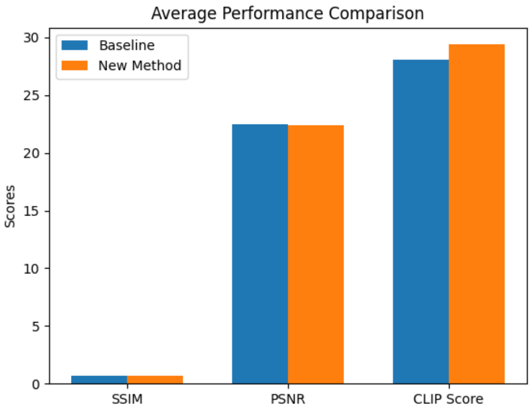

Figure 2: *SSIM , PSNR and CLIP Score after training on VisualGenome dataset.*

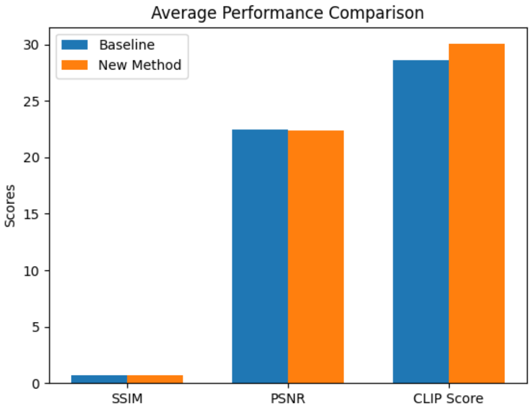

Figure 3: *SSIM , PSNR and CLIP Score after training on MSCOCO dataset.*

image. Finally, the loss function is calculated between the inpainted image, ground-truth image and the caption. This loss is backproped through the model, and the parameters of BLIP are updated.

This approach faces a major block, however. Since there is a disconnection between BLIP and the diffusion model, backpropagation *cannot* be implemented in practice. To tackle this issue, a **Scoring Network** is added to the architecture. This is a multi-layer perceptron (MLP) that is trained to be an estimator for the loss function. This network is attached to BLIP after its last layer is removed. It outputs $\mathcal{L}_{\text{CLIP}}, \mathcal{L}_{\text{PSNR}}$ and $\mathcal{L}_{\text{SSIM}}$. This way, the model becomes a completely connected network that is trainable. Note that during the training procedure of the Scoring Network, BLIP is frozen. Similarly, the Scoring Network is frozen while fine-tuning the transformer.

Experiments and Results

We first used the pipeline for inference and the calculation of loss terms. After this phase, the last layer of transformer was detached and a multi-layer neural network was added instead. Our goal here was to train the neural "score" network. Therefore, the transformer's parameters were frozen. Using Adam optimizer and $\text{lr} = 10^{-4}$, the network was trained for 50 epochs. After this step, the transformer was fine-tuned. Before fine-tuning, we set $\lambda_3 = 4, \quad \lambda_2 = 0.2, \quad \lambda_1 = 0.5$. In this phase, the score network's parameters were frozen and the parameters of the transformer's text decoder were updated. The fine-tuning process was terminated after 3 epochs with $lr = 10^{-6}$ and Batch Size $= 8$. The learning rate is selected small in order to prevent the parameters from changing too rapidly.

1. Baselines

The transformer for our pipeline was **BLIP**[9] and the diffusion model used in the pipeline and as baseline as well was **Stable Diffusion Inpaint Base**[3]. In contrast to the pipeline, baseline stable diffusion does not receive any text as input. The input images used were 512×512 in size. Using smaller images often resulted in blurriness since Stable Diffusion works best with 512×512 images.

2. Datasets

Two datasets were used during inference. These datasets were *MSCOCO*[10] and *VisualGenome* [11]. These datasets are filled with images containing rich contexts that consist of diverse semantics.

3. Evaluation Metrics

Our main evaluation metric is the **Frechet Inception Distance (FID)**[12] which is representative of the quality of images generated by a model, and how aligned the output is with the ground-truth. This metric is also commonly used when dealing with generalization evaluation in generative models. Lower FID indicates better outputs.

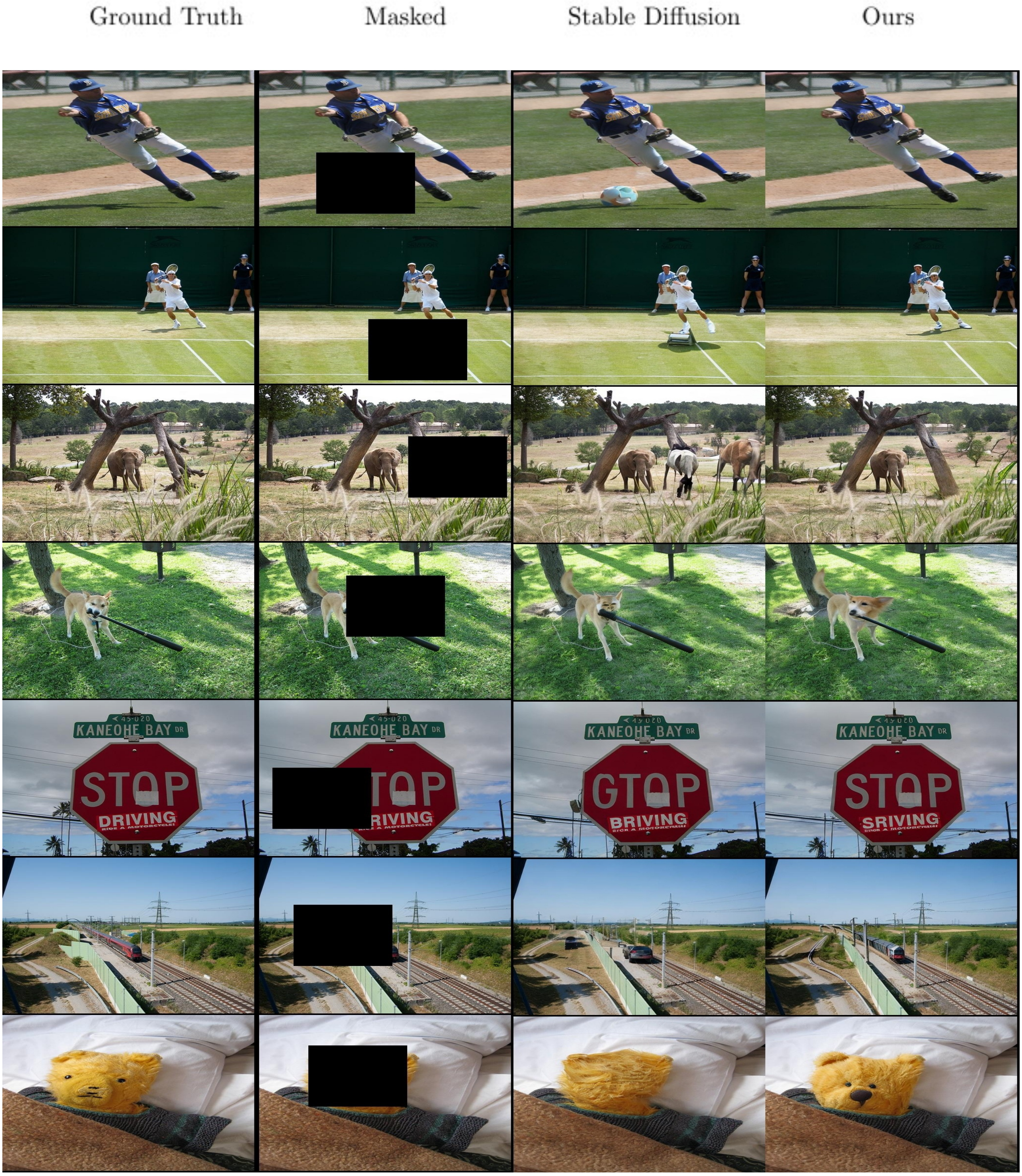

Figure 4: *In this figure some of the results are shown. We can observe that some of our outputs are more aligned with the context of the background. In the fourth line, our model was able to understand that this is in fact a dog and not a cat. Similarly, in the next line, the stop sign is properly filled with the word "STOP". Then in the second to last line, the car is instead replaced by a train because of the context of the railway.*

	VisualGenome		MSCOCO	
	CLIP Score ↑	FID ↓	CLIP Score ↑	FID ↓
Stable Diffusion [3]	27.14	46.68	27.23	49.25
Ours	**29.88**	**44.15**	**29.84**	**46.73**

Table 1: *Comparison of different inpainting models on VisualGenome and MSCOCO datasets. Lower FID and higher CLIP Score are better.*

4. Results

In the final inference phase we can see that FIDs have decreased for both datasets of our model compared to the baseline while CLIP Scores have increased as shown in 1. Aside from numerical evaluation, some example outputs are also shown in 4 which indicate that the model has indeed reached a better understanding of the context of the image before inpainting and therefore produced a better output.

Conlcusion and Limitations

In this work, we addressed the challenge of generalization in image inpainting models, specifically focusing on the problem of understanding the semantic context of the image background. By leveraging a transformer-based architecture, we introduced a novel method to generate text descriptions of the unmasked background, which serves as a guiding condition for the inpainting process. The generated text embeddings are used to improve the model's understanding of the surrounding context, thereby enhancing the quality of the inpainted images, particularly when dealing with diverse semantics.

We demonstrated the effectiveness of our approach by integrating BLIP, a pre-trained image captioning model, into the inpainting pipeline. The model not only provides textual descriptions that reflect the context of the masked regions but also contributes to the generalization of the inpainting model. Through the fine-tuning process, we ensured that the text embeddings generated by BLIP better align with the inpainting task, resulting in more accurate and context-aware inpainted outputs.

Furthermore, we introduced a scoring network to overcome the challenge of backpropagation between the captioning model and the diffusion-based inpainting model. This network allows the entire architecture to be trained end-to-end, ensuring seamless optimization of both components. Our experiments validate that text-guided inpainting significantly improves image quality, especially when the input image contains complex or varied semantic contexts.

The main limitation of our pipeline is the large mask case, where the mask is so big in size that no context can be derived from it and therefore the generated text is quite possibly irrelevant. This means that the generated text might even mislead the diffusion model and its performance drops below the baseline.

The integration of text guidance through transformers offers a promising direction for the advancement of inpainting models, to eventually surmount the challenge of generalization and semantic alignment. This work represents a step forward in the quest to build more robust, context-aware inpainting systems, capable of handling diverse and intricate real-world scenarios. Future work may explore further optimizations to the captioning model and investigate the potential of multi-modal guidance for even more sophisticated inpainting results.

References

[1] Ian Goodfellow et al. “Generative adversarial networks”. In: *Advances in neural information processing systems*. Vol. 27. 2014. URL: `https://doi.org/10.48550/arXiv.1406.2661`.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 6840–6851. URL: `https://arxiv.org/abs/2006.11239`.

[3] Robin Rombach et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. In: *arXiv preprint arXiv:2112.10752* (2022). URL: `https://arxiv.org/abs/2112.10752`.

[4] Omar Elharrouss et al. “Transformer-based Image and Video Inpainting: Current Challenges and Future Directions”. In: *arXiv preprint arXiv:2407.00226* (2024). URL: `https://arxiv.org/abs/2407.00226`.

[5] Wenbo Li et al. “MAT: Mask-Aware Transformer for Large Hole Image Inpainting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10758–10767. DOI: `10.1109/CVPR52688.2022.01051`. URL: `https://openaccess.thecvf.com/content/CVPR2022/papers/Li_MAT_Mask-`

Aware_Transformer_for_Large_Hole_Image_Inpainting_CVPR_2022_paper.pdf.

[6] Jingjing Ren et al. “DLFormer: Discrete Latent Transformer for Video Inpainting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3511–3520. URL: https://openaccess.thecvf.com/content/CVPR2022/papers/Ren_DLFormer_Discrete_Latent_Transformer_for_Video_Inpainting_CVPR_2022_paper.pdf.

[7] Prafulla Dhariwal and Alex Nichol. “Diffusion Models Beat GANs on Image Synthesis”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 8780–8794. URL: https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

[8] Jonathan Ho and Tim Salimans. “Classifier-Free Diffusion Guidance”. In: *arXiv preprint arXiv:2207.12598* (2022). URL: https://arxiv.org/abs/2207.12598.

[9] Junnan Li et al. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. DOI: 10.48550/ARXIV.2201.12086. URL: https://arxiv.org/abs/2201.12086.

[10] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision (ECCV)*. Springer. 2014, pp. 740–755.

[11] Ranjay Krishna et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4268–4276.

[12] Martin Heusel et al. “GANs trained by a two time-scale update rule converge to a local Nash equilibrium”. In: *Neural Information Processing Systems (NeurIPS)*. 2017, pp. 6626–6637.