Image Inpainting for Irregular Lines Using Partial Convolutions

Chang Shu **Burak Sahinkaya**
University of California, San Diego
La Jolla, CA 92093

Abstract

Deep learning applications have been evolving rapidly and tremendously as various techniques emerge to fulfill image processing tasks. Image Inpainting being one of these techniques holds an important role for image restoration and content generation. This project will therefore reimplement and enhance an Image Inpainting algorithm proposed by Liu et al.[3]. Our primary goal is to achieve the results proposed in the paper using the proposed methodology and a different dataset. We reimplement the Partial Convolution approach that specifically addresses the reconstruction of images with irregular line masks while preserving its semantic context. Partial Convolutions comprise two main components: the encoder and the decoder. The encoder is adept at leveraging the information from unmasked pixels around the masked areas, ensuring the inpainting process is both seamless and contextually relevant. On the other hand, the decoder is responsible for creating the content within any random region of the image. When training Partial Convolution, we have experimented with three different loss: a standard pixel-wise L2 loss, an adversarial loss proposed by Pathak et al.[4], and a perceptual loss. The adversarial loss produces much clear results and the perceptual loss minimizes the difference between the inpainted and original images by utilizing VGG pre-trained CNN. Besides, we also utilize Gaussian noise to improve the robustness and generalization of our neural network. We systematically analyze the model's performance through extensive experiments, demonstrating its advantage over conventional methods in handling complex irregular line masks.

Figure 1: Line masked images and corresponding inpainted results using our Partial Convolution based neural network.

1 Introduction

Image inpainting, the art of filling in missing or damaged regions of visual data, has experienced a renaissance with the advent of deep learning techniques. The task is inherently challenging due to the need for semantic understanding and contextual inference, particularly when dealing with irregular holes or missing segments in images. Conventional convolutional neural networks (CNNs) often struggle with such tasks as they treat all pixels equally, and focused on rectangular regions located around the center of the image. Those CNNs relied on expensive post-processing. To address this limitation, Liu et al. proposed a model called Partial Convolutions for image inpainting that operates on irregular hole patterns (see Fig. 2), and produces meaningful predictions that can incorporate with the surrounding context smoothly without any post-processing operation.

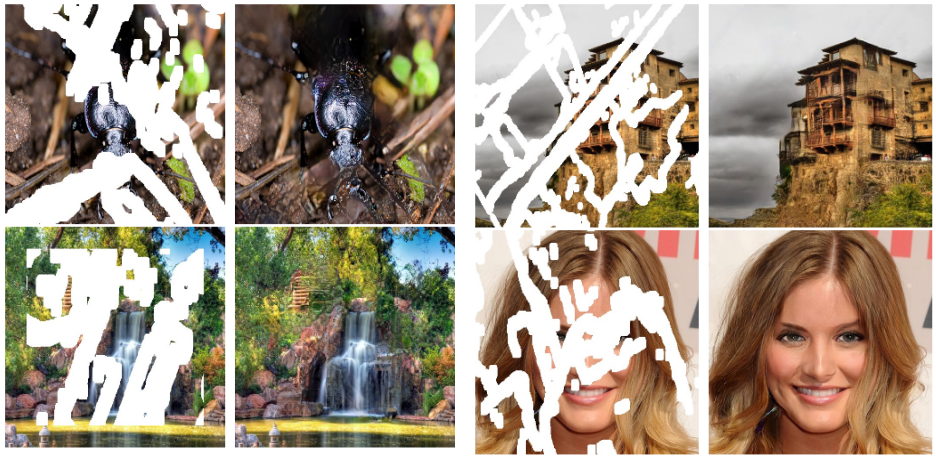


Figure 2: Images with irregular holes and their associated restorations achieved through the Partial Convolution method proposed by Liu et al.

The goal of our project is to reimplement the model proposed by Liu et al (see Fig. 1). Instead of using the UNet-like architecture, we use a simple encoder-decoder pipeline from the paper proposed by Pathak et al.

- Encoder aims to only utilize the unmasked pixels around the masked regions, thereby avoiding the influence of masked parts. It uses two Partial Convolution layers to downsample the image into a lower-dimensional feature space.
- Decoder is to reconstruct the input image from the encoded feature space. It basically takes the output from the previous layer and the corresponding mask, upsamples them, and then concatenates the upsampled output with the features from a corresponding encoder layer (using the skip connection) to preserve spatial information lost during downsampling.

Our Partial Convolutions are trained in a completely unsupervised manner. During the training of the Partial Convolution-based model for image inpainting, we explored a variety of loss functions to measure and improve the performance of our network. We implemented a conventional pixel-wise L2 loss, which calculates the mean squared error across all pixels in the image, ensuring a direct pixel-level accuracy between the generated and the target images.

Additionally, we integrated an adversarial loss as proposed by Pathak et al. in their research. This loss function leverages a discriminative network to distinguish between the inpainted and original images, which prompts the generative model to produce outputs that are indistinguishable from the real images. In our project, we use nn.BCEWithLogitsLoss() to enhance the clarity of the results, by adding a competitive dynamic between the generator and discriminator networks to encourage more realistic inpainting.

We also employed a perceptual loss function, which involves the use of a pre-trained VGG network. This loss compares the high-level feature representation of the inpainted output and the original image, as derived by the VGG network, thereby minimizing the perceptual differences between them. This approach ensures that the inpainted output is visually coherent with the surrounding

image regions, maintaining consistency in terms of textures and patterns.

To further bolster the robustness and generalization ability of our Partial Convolution, we introduce Gaussian noise to the input data during training. This stochastic augmentation ensures that the model is not just memorizing the input-output mapping but is learning to handle variations, leading to improved performance on unseen data.

2 Related Work

In the rapidly evolving field of deep learning-based semantic image understanding and object detection[1] [2], our project endeavors to reimplement and enhance the algorithm proposed by Liu et al.[3], focusing on the application of Partial Convolutions of reconstructing the irregular line masks. The methodological backbone of our project is characterized by an innovative utilization of encoder-decoder architectures, coupled with a standard pixel-wise L2 loss, an adversarial loss inspired by Pathak et al.[4], and a perceptual loss to refine the inpainting process for better quality and higher realism in the reconstructed images.

A recent study of High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis [5] underscores the significance of utilizing neural features for creating textures and details in high-resolution images, employing a mix of content, texture, and TV-loss terms. This inspires our project's aim to harness the semantic context and texture fidelity, albeit through the lens of Partial Convolutions, thereby enhancing the quality of inpainted images.

Liu et al. introduced the concept of Partial Convolutions, a novel approach that specifically targets the challenge of inpainting images with irregularly shaped holes. The key insight from their work is the partial convolutional layer's ability to leverage unmasked pixels for generating the inpainting output, thereby ensuring the reconstructed regions are both contextually coherent and seamlessly integrated with the surrounding image data. This methodology underpins our project's approach, where we aim to reimplement and enhance the Partial Convolution algorithm to address the intricacies presented by irregular line masks while preserving the semantic context of the inpainted areas.

Pathak et al., through their development of Context Encoders, further expand the scope of image inpainting by employing a convolutional neural network (CNN) trained to generate the contents of an arbitrary image region conditioned on its surroundings. Their work underscores the potential of CNNs to understand the content of the entire image and produce plausible hypotheses for the missing parts, facilitated by a combination of standard pixel-wise reconstruction loss and adversarial loss. The adversarial component, in particular, contributes to the generation of sharper, more realistic inpainted images by effectively handling the multi-modal nature of the output space. Our project incorporates these insights, experimenting with adversarial loss alongside pixel-wise L2 loss and perceptual loss to enhance the clarity and contextual relevance of the inpainted output.

3 Methodology

We now introduce our neural network that predict the line masked regions. We first define our convolution and mask regions generation, and then discuss our network architecture and loss function.

3.1 Partial Convolution layer

The cornerstone of our methodology is the Partial Convolution layer, an innovative approach that aims to reconstruct the inpainting images with irregularly shaped lines. The partial convolution operates uniquely and make sure the accuracy of our model, as it only considers unmasked pixels during the convolution process. This selective consideration is pivotal for tasks like image inpainting, where certain image regions are missing.

Each Partial Convolution layer comprises two components: an input convolution and a mask convolution. The input convolution applies to the input image multiplied by a binary mask, while the mask convolution transforms the binary mask itself. In addition, the weights of the mask convolution are initialized to 1 and frozen to prevent learning, ensuring that only the area of valid pixels influences the feature learning process. Then, a normalization step follows, compensating for the varying number of valid pixels within the convolution window across the image.

3.2 Network Architecture and Implementation

Implementation Our network utilizes a simplified encoder-decoder architecture which is a customized structure, departing from the more complex UNet-like structures. The encoder comprises stacked Partial Convolution layers that downsample the image into a lower-dimensional feature space while being conscious of the masked regions. This downsampling process is achieved without the influence of the masked pixels, ensuring that only relevant information is propagated through the network. The decoder's role is to reconstruct the input image from the encoded features. This is accomplished through a series of upsampled Partial Convolution layers, where each layer's output is merged with corresponding features from the encoder via skip connections. These connections are instrumental in preserving spatial information that may be lost during encoding.

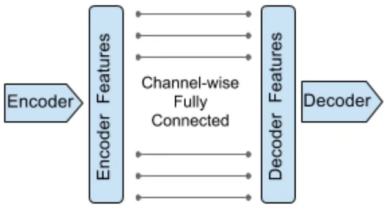


Figure 3: Encoder-Decoder model[4]

Encoder in the inpainting model architecture is composed of sequential stages that refine the features extracted from input images with missing regions. Initially, the image and its corresponding binary mask pass through the first partial convolution layer (p_conv_1), which we defined above, followed by batch normalization and ReLU activation to stabilize and introduce non-linearity to the learning process. Subsequently, the output from the first stage is further downsampled and abstracted by the second partial convolution layer (p_conv_2), with an increased stride to reduce spatial dimensions and again normalized and activated. This double-layered approach ensures that each encoder layer progressively captures more complex and abstract representations necessary for inpainting while preserving contextual information from unmasked parts of the image.

Decoder in the inpainting model architecture is essential for reconstructing the final image from encoded features. It begins by upsampling the input features and corresponding masks to a higher resolution to match the size of earlier features in the network. These upsampled features are then combined with corresponding features from earlier encoder layers through skip connections, which helps in restoring spatial details lost during downsampling. Next, the merged features undergo two sequential partial convolutions, followed by batch normalization and activation with a LeakyReLU to further refine and enhance the feature representation. This process helps in generating a detailed and contextually coherent inpainting of the missing image areas, effectively reversing the encoding process.

3.3 Loss function

In our approach, we utilize a composite loss function that combines multiple criteria to measure the performance of our network:

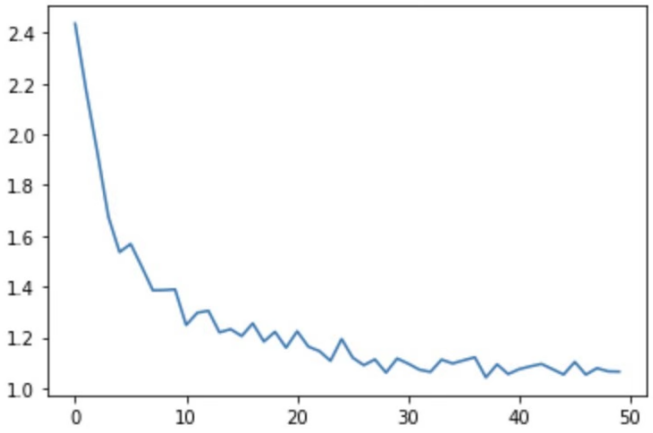
- **Pixel-wise L2 Loss:** This loss computes the mean squared error across all pixels, directly correlating the pixel-level accuracy of the generated image to the target.
- **Adversarial Loss:** Through the nn.BCEWithLogitsLoss() function, we add an adversarial dynamic to our training process. This loss is integral to the adversarial training approach within a Generative Adversarial Network (GAN) for image inpainting. It evaluates the discriminator's ability to distinguish between real and inpainted images while simultaneously encouraging the generator to produce more convincing, realistic images. Thus, it enhances clarity and reducing the likelihood of producing blurred results.
- **Perceptual Loss:** Leveraging the feature extraction capabilities of a pre-trained VGG network, this loss assesses the high-level feature discrepancies between the inpainted output and the original image. This comparison not only promotes the visual coherence of the

inpainted regions with their surroundings but also preserves the textural and contextual integrity.

To further improve the robustness of our model, we introduce Gaussian noise to the input data during the training phase. This augmentation induces a degree of randomness that helps the model generalize better to unseen data and not merely memorize the input-output mappings (see Fig. 3).

3.4 Training process

The training process proceeds as follows. First, Gaussian noise enhancement is performed on each input image to amplify the features, thus allowing our model to be better trained and enhancing the generalization ability of the model. Then the model performs a forward pass where the Partial Convolution layers process the noisy input image and its corresponding mask. After this, we do a combined loss that sums the pixel-wise L2 loss, adversarial loss, and perceptual loss. Lastly, we use the Adam optimizer to update the model parameters based on the gradients calculated during backpropagation. During the training of our partial convolutional model, there is a downward trend in the loss values, which indicates that our model is learning and improving its predictions over time (see Fig. 4).




```
CPU times: user 13min 32s, sys: 1min 50s, total: 15min 23s
Wall time: 15min 33s
```

Figure 4: Loss value

The testing process involves running the trained model on a separate test set and evaluating its performance using the Jaccard coefficient, which quantifies the overlap between the predicted and actual inpainted regions.

Our approach differentiates itself from previous works by the customized encoder-decoder architecture and the integration of adversarial and perceptual losses with pixel-wise L2 loss, a combination not standard in the field. The addition of Gaussian noise during training also represents an innovative step to increase the model's generalizability. These elements collectively contribute to a more robust and contextually sensitive inpainting process, capable of handling the complex irregular line masks often seen in real-world scenarios.

4 Experiments

In this section, we expand on the experiments conducted to evaluate the performance of our image inpainting algorithm. We will describe how different approaches brought the conclusions we achieved.

Figure 5: Randomly Generated Masks

Figure 6: Masked Images

4.1 Datasets

Our choice of datasets was determined based on their size, availability, and the unique features they offer for our training process. We opted to train our network using three distinct datasets to facilitate learning within our architecture. We occasionally truncated the dataset for training constraints and resized the images to either 64x64 or 128x128 to fit into our model.

- Flowers102: The Flowers102 dataset served as our primary training dataset, including 102 classes, each comprising between 40 to 200 images. While the dataset offers a great number of labeled classes, our focus lay more on the similarities among the images rather than the class diversity. This dataset, characterized by relatively similar images compared to datasets like ImageNet, facilitated easier differentiation of common features within our architecture. Furthermore, the textures present in these images involves lesser complexity, enabling us to achieve higher qualitative accuracy in image recreation. Therefore, Flowers102 served as an excellent starting point for our training process.
- CelebA: With its vast collection of over 200,000 celebrity images, CelebA presented a rich reserve of facial expressions, poses, and occlusions. The complexity of facial structures demanded a more precise and robust training approach to achieve relevant and qualitatively successful image inpainting. Due to the massive size of the dataset and our physical and materialistic limitations, we had to truncate this dataset to include a smaller version. Training on CelebA provided our model with exposure to greater textural complexity, determining its ability to handle diverse challenges.
- Describable Texture Dataset (DTD): The Describable Texture Dataset (DTD) provided another valuable addition to our training process. This dataset includes 5640 images from 47 different categories or texture types. Describable Texture Dataset offers a diverse range of textures, enriching our model's understanding of textural differences and aiding in the inpainting of images with various surface patterns.

4.2 Masking Approach

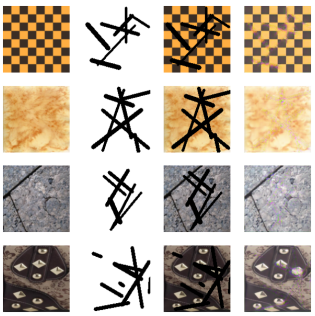
To train our model, we generated custom masks to simulate various damaged imaging scenarios. As proposed by Liu et al., inserting square masks at specific positions in the images does not provide enough regularization. Therefore we recreated the randomized masks that offers more diversity. Our implementation for the masking consists of random lines generated with random line thickness, which could start and end anywhere on the image. We generate the masks when we call the dataset. Therefore, when we initiate the dataloader, we create both the masks and the masked images simultaneously.

4.3 Results

After training our model implementation with different three datasets, we obtained results as presented. Several observations arise from the outcomes of these experiments. We can see that even though we keep the dimensions of the Flowers102 dataset relatively low at 64x64, we can successfully recover our image keeping the semantic context and recreate figures that have qualitatively accurate results compared with the Ground Truth. Conversely, our training with the CelebA dataset, comprising 6400

(a) Flowers102 Dataset

(b) CelebA Dataset



(c) Describable Texture Dataset

Figure 7: Test Set Results

images at dimensions of 128x128, yielded less satisfactory results. Some fillings feel unnatural due to the high complexity of facial features and inability of the network to extract these features. With possibly more GPU memory and faster computation speeds, more data could have been used in the training process of this architecture. This would have resulted in higher image inpainting success, as we would be able to generalize more of the data and provide higher qualitative accuracy compared with the Ground Truth. The results from Describable Texture Dataset provides us better insight about how the training of this truly model works. The same training process was implemented for DTD as for the other trainings. We used the entirety of the dataset with 128x128 images and trained for 40 epochs. The results indicate that this network model can recreate figures that closely resembles each other with much higher qualitative accuracy. Since this dataset includes various textures with more variety, our restored versions of the damages images cannot easily recreate the pattern that are available on the images. We can observe that the restored images do not closely resembles the Ground Truth. This issue can be resolved with more training with much larger dataset, as DTD does not include a lot of images from the same category. An interesting observation from this training results is as the noise of the images get higher, our network yields more natural looking images.

5 Acknowledgements

We benefit from the dataloader implementation proposed by Shivkumar25 on GitHub.

References

[1] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[2] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. *CoRR*, abs/1412.6056, 2014.

[3] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. *CoRR*, abs/1804.07723, 2018.

[4] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016.

[5] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. *CoRR*, abs/1611.09969, 2016.