Clam Contour Reconstruction Based on GAN

1st Wenyang Lyu
Auckland University of Technology
Auckland, New Zealand
sxb7657@autuni.ac.nz

2nd Yanbin Liu
Auckland University of Technology
Auckland, New Zealand
yanbin.liu@aut.ac.nz

3rd Wei Qi Yan
Auckland University of Technology
Auckland, New Zealand
weiqi.yan@aut.ac.nz

***Abstract*—Image inpainting is a crucial technology for restoring the original appearance of obscured images. In this paper, we address the challenging task of clam contour reconstruction by using image inpainting techniques to remove seaweed, pebbles, and other debris from clams to obtain their complete shape. We propose a Generative Adversarial Network (GAN) that includes a modified channel-wise encoder-decoder network as the generator and a convolutional network discriminator. The joint loss function utilized combines weighted Mean Square Error (MSE) between overlapping regions and missing regions for the generator and Binary Cross-Entropy (BCE) for the discriminator. We tested the model on clams obscured by 12.5% to 50% and assessed its performance using Mean Absolute Error (MAE), Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM), achieving promising results compared with the baseline model. This method offers significant technical support for the rapid reconstruction of obscured clams in fisheries, facilitating size measurement and further processing.**

***Index Terms*—Contour reconstruction, encoder, generative adversarial network, image inpainting**

I. Introduction

Traditional [1] image inpainting methods include exemplar-based texture synthesis, structure synthesis, diffusion-based methods, sparse representation methods, and hybrid inpainting. These methods have shown success in generating high-frequency seamless textures but often fail to maintain realistic structures and handle complex masks, resulting in repetitive patterns and unrealistic images. Exemplar-based methods are good for preserving textures but struggle with large damaged areas and can lead to repetitive results. Diffusion-based methods preserve edges well but produce blurry artifacts in large regions. Sparse representation methods work well for facial images but not for natural scenes with complex structures. Hybrid methods combine advantages but are computationally intensive.

Given these limitations, deep learning methods, particularly those using generative neural networks, have emerged, offering better performance by learning high-level semantics for coherent inpainting.

The introduction of deep learning has significantly changed the approach to image inpainting, making earlier methods seem outdated. Traditional techniques like diffusion and patch-matching worked well for small missing areas in simple images [2]. Yet, they often fell short when faced with larger gaps or more complex textures. This limitation led to the adoption of deep learning models, which are better at managing detailed and complicated image challenges, greatly improving the quality of inpainted images.

Deep learning has significantly advanced image inpainting techniques, primarily through Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs). Modified CNN networks like U-Net and Fully Convolutional Network (FCN) have laid the foundation for future inpainting algorithms, both employing an Encoder-Decoder architecture. The encoder extracts features from input images, while the decoder reconstructs the images using these features. Since the introduction of GAN by Goodfellow et al. [3] and their enhancement by Pathak et al. [4] with the Context Encoder, numerous models combining GANs with encoder-decoder or U-Net structures have emerged. These models, leveraging advanced GPU capabilities, address higher resolutions, larger missing regions, and more complex textures, achieving impressive results. This progress has enabled robust image inpainting, essential for applications requiring high-quality image restoration.

In the process of harvesting clams, it is crucial to quickly and automatically determine the size of each clam, typically achieved by identifying the contour of the clam and calculating the maximum Feret Diameter. However, this task becomes challenging when clams are obscured by seaweed, stones, other fish, or debris. To accurately measure the size, it is necessary to first restore the original contour of the clam, an image inpainting task. Considering the stringent requirements of more advanced models for larger datasets, powerful equipment, and extensive training time, as well as our limitations of a smaller training dataset, CPU-only equipment, and limited training time, we chose the classic GAN-based context encoder model from [4] as the baseline. We then made improvements to this model to achieve better results. The effectiveness of this approach will be compared against the results achieved by a baseline model, showcasing our method's capability to handle complex inpainting tasks required for accurate clam measurement.

II. Literature Review

Traditional image inpainting methods, as summarized in [1], include exemplar-based texture synthesis, structure synthesis, diffusion-based methods, sparse representation methods, and hybrid inpainting. These methods have shown success in generating high-frequency seamless textures but often fail to maintain realistic structures and handle complex masks,

resulting in repetitive patterns and unrealistic images. In recent years, diffusion models (DMs) [5], including Stable Diffusion, DALL-E, and Imagen, have shown significant promise in text-to-image generation. These models can be adapted for inpainting by replacing the random noise in the background with a noisy version of the original image during the diffusion reverse process.

Given these limitations, deep learning methods, particularly those using generative neural networks, have emerged, offering better performance by learning high-level semantics for coherent inpainting.

A. Deep Learning Methods

In recent years, significant advancements in image inpainting have been achieved using deep learning techniques, notably Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Graph Convolutional Networks (GCNs).

GAN-based Models have been extensively used in image inpainting due to their ability to generate high-quality and realistic images. These models consist of a generator, which creates plausible image content, and a discriminator, which evaluates the realism of the generated content. The adversarial nature of GANs encourages the generator to produce highly realistic images that are difficult to distinguish from real ones. Wang et al. (2022) [6] present an innovative approach to improving image inpainting. The proposed architecture features a dual-path design, consisting of a reconstruction path and a semantic path. The reconstruction path is dedicated to generating fine details in the inpainted regions, while the semantic path ensures global coherence and high-level semantic accuracy. The introduction of Auxiliary GAN Inversion is a significant contribution, as it allows the model to leverage pretrained GANs to enhance texture and structural detail preservation. This approach effectively addresses the challenges of maintaining realistic textures and coherent structures in large missing areas, offering a robust solution that outperforms traditional methods by achieving a balance between fine detail generation and global image consistency.

Variational Autoencoders (VAEs) provide a probabilistic framework for image generation and inpainting. They work by encoding the input image into a latent space and then decoding it back to reconstruct the image. VAEs are effective in capturing the underlying data distribution, which helps in generating coherent and contextually accurate inpainted regions. Lin et al. (2023) [7] introduces the Frequency Augmented VAE (FA-VAE), designed to address the issue of image reconstruction quality degradation at high compression rates. The FA-VAE incorporates Frequency Complement Modules (FCM) into its decoder to complement missing frequency information, enhancing the detail in reconstructed images. Additionally, the paper introduces Spectrum Loss (SL) and Dynamic Spectrum Loss (DSL) to guide the FCMs in dynamically learning the most critical frequency mixtures for optimal reconstruction. The architecture also includes a Cross-attention Autoregressive Transformer (CAT) for improved text-to-image synthesis by utilizing fine-grained textual embeddings for better image-text semantic alignment. This approach effectively addresses the challenges of spectral bias in autoencoders and improves reconstruction accuracy by better aligning frequency spectrums between original and reconstructed images. The innovations in this paper lie in the integration of frequency-based modules and dynamic loss functions, which significantly enhance image reconstruction quality compared to existing methods

Graph Convolutional Networks (GCNs) leverage the inherent structural properties of images by representing them as graphs. By modeling the image as a graph, these methods can effectively handle high-resolution inpainting tasks and overcome the limitations of traditional pixel-based techniques. Verma et al. (2024) [8] presents GraphFill, an innovative image inpainting method using graph neural networks (GNNs). The architecture employs a pyramidal graph construction that segments images into superpixels, each represented as graph nodes, allowing the transfer of global context from coarse to fine levels. This approach facilitates efficient inpainting, even for high-resolution images, with reduced computational requirements. GraphFill, validated on Places365 and CelebA-HQ datasets, shows competitive performance with fewer parameters than existing methods. The key innovation is the application of GNNs for image inpainting and the efficient pyramidal graph structure, making it suitable for mobile devices.

B. Dataset

The four most commonly used datasets for image inpainting training are Places2, Paris Street View, CelebA, and CelebA-HQ.

Places2 contains over 10 million images across more than 400 scene categories and is designed for high-level visual tasks like scene recognition, using convolutional neural networks (CNNs) to learn detailed scene features.

Paris Street View includes geographically informative images from 12 cities worldwide, particularly focusing on Paris and its suburbs, with approximately 10,000 images per city. It is valuable for computational geographic tasks, providing a rich resource for examining architectural and geospatial patterns.

CelebA offers 202,599 celebrity facial images, annotated with 40 binary attributes and five landmark locations, making it ideal for facial image synthesis tasks due to its diversity in poses and backgrounds.

CelebA-HQ, an extension of CelebA, consists of 30,000 high-quality images at various resolutions. It enhances image quality through preprocessing steps like artifact removal and super-resolution techniques, making it suitable for advanced image synthesis and manipulation tasks.

C. Quantitative Comparison

When evaluating image inpainting algorithms, it is important to consider both qualitative and quantitative comparisons. Visual inspection of reconstructed images with the same input images and masks provides an intuitive qualitative comparison.

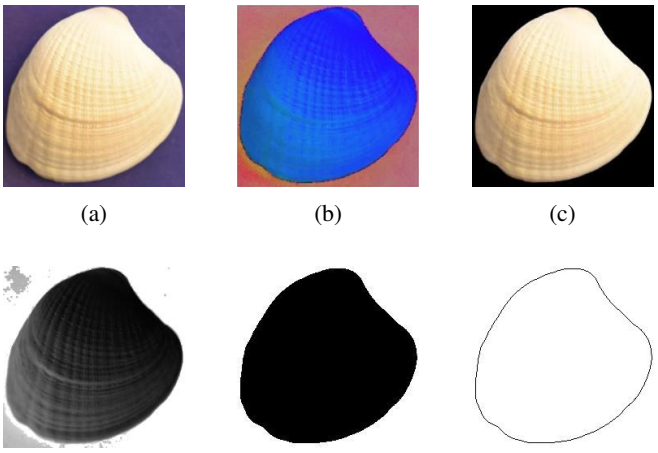
However, quantitative assessments are equally crucial. Recent studies frequently utilize four main metrics for this purpose: Mean Absolute Error (MAE), which measures the average absolute difference between the pixel values of the original and reconstructed images, with lower values indicating better performance; Mean Squared Error (MSE), which computes the average squared difference between pixel values, with lower values signifying superior results; Peak Signal-to-Noise Ratio (PSNR), which evaluates the ratio of the maximum possible signal power to the power of corrupting noise, expressed in decibels (dB), with higher values reflecting better image quality; and Structural Similarity Index Measure (SSIM) [9], which assesses the similarity between the original and reconstructed images in terms of luminance, contrast, and structure, with a range from 0 to 1, where higher values denote higher quality inpainting.

III. Methodology

In this study, we aim to develop a robust image inpainting method to reconstruct the contours of clams damaged by random masks. We first pre-process and augment the images in the dataset, and then train a enerative Adversarial Network (GAN) composed of a channel-wise encoder-decoder network and a convolutional network discriminator. Finally, we use test data to perform quantitative and qualitative comparisons with the context encoder baseline model [4].

A. dataset

We start with a dataset of 478 clam images. Since our ultimate goal is to measure the size of clams, we only need to restore the complete contours of the clams that are partially occluded. Therefore, we pre-process the clam images to extract their contour information, and in subsequent training, we use white masks to randomly cover parts of the clams and attempt to restore the original contours.



(a) (b) (c)

(d) (e) (f)

Fig. 1: Image pre-processing steps. (a) clam boundary detection and auto-cropping; (b) HUE channel filtering; (c) background removing; (d) gray-scale converting; (e) largest blob detection; (f) contours extraction

The image pre-processing steps are as follows (as shown in Fig. 1):

a) **Clam boundary detection and auto-cropping**: Automatically detect the boundaries of the clams and crop the images accordingly.
b) **HUE channel filtering**: Filter the images based on the HUE channel to enhance the distinction between the clam and the background.
c) **Background removing**: Remove the background to isolate the clam in the image.
d) **Gray-scale converting**: Convert the images to gray-scale to simplify the contour extraction process.
e) **Largest blob detection**: Detect the largest blob in the image, which corresponds to the clam.
f) **Contours extraction**: Extract the contours of the clam for further processing.

To increase training samples, we apply data augmentation techniques such as random scaling, flipping, and rotation to simulate various orientations and sizes of clams. This expands the training set to 8,000 images and the test set to 2,000, ensuring meaningful variability. Random masks, ranging from 28x224 to 112x224 pixels, mimic clams obscured by debris. These masked images are used as inputs for our inpainting model.

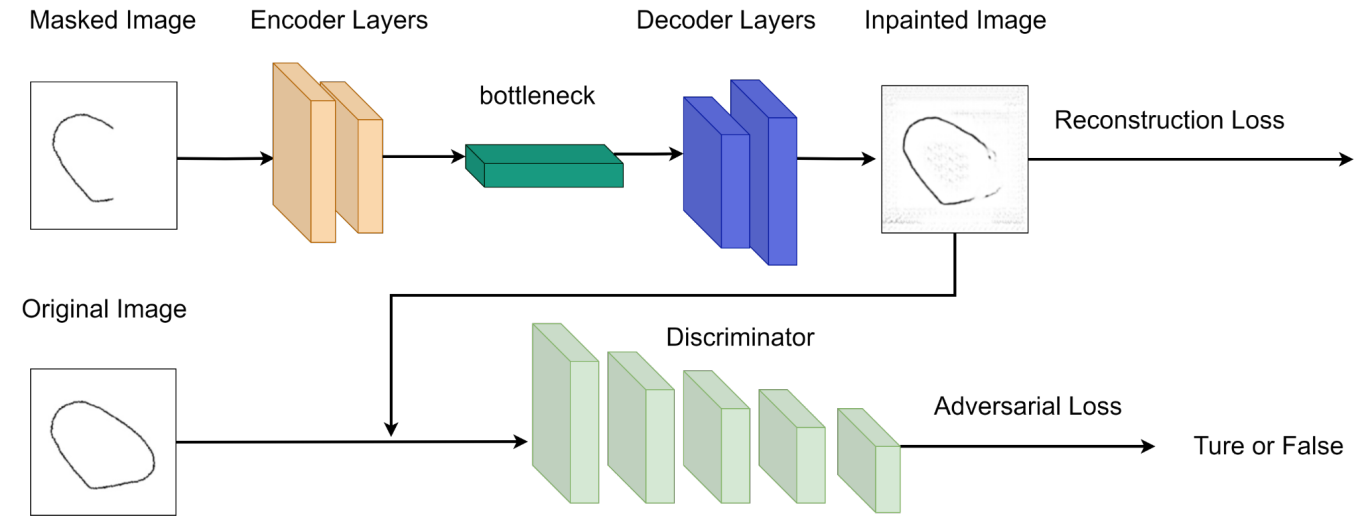
B. architecture

The GAN operates by training two networks simultaneously, as illustrated in Fig. 2a: a generator that creates inpainted images and a discriminator that differentiates between real and generated images. The generator aims to produce images indistinguishable from real ones, while the discriminator works to accurately classify images as real or fake. This adversarial process continues until the generator produces highly realistic images.

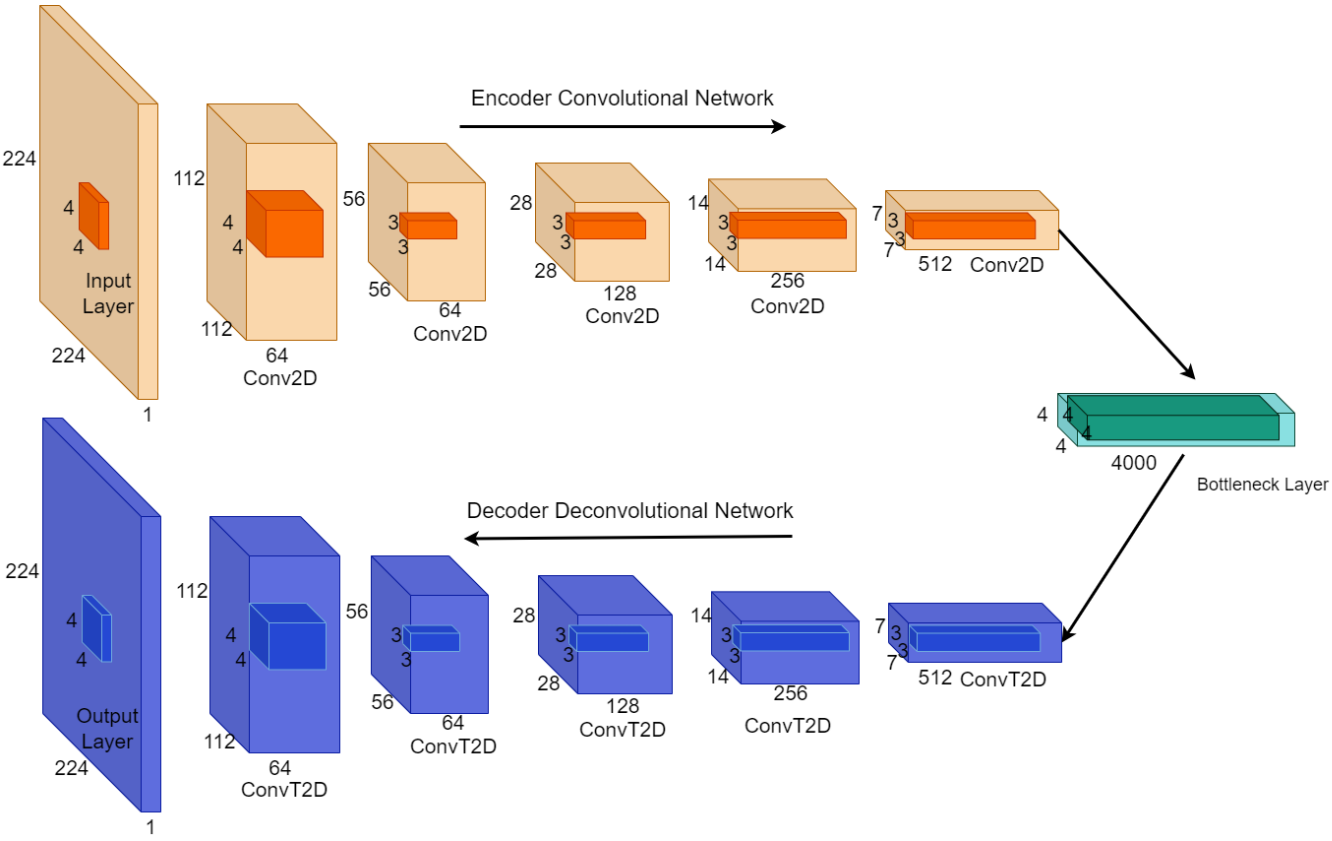
The core of our approach is a modified channel-wise encoder-decoder network used as the generator, as shown in Fig. 2b. This network is designed to capture and reconstruct the detailed structures of clams contour obscured by debris. The encoder part of the network compresses the input image into a latent representation, which the decoder then uses to generate the inpainted image. This channel-wise design helps the model focus on different aspects of the image independently, enhancing the quality of the reconstruction.

The encoder compresses the input image into a latent representation using Conv2D layers. It starts with 64 filters of size 4x4, halving spatial dimensions, and then doubles the filters (128, 256, 512) with 3x3 kernels, further reducing dimensions, followed by LeakyReLU activation and batch normalization. The bottleneck layer compresses the representation to [64, 4000, 4, 4].

The decoder reconstructs the image using ConvTranspose2D layers, starting from [64, 512, 7, 7], halving the filters and doubling spatial dimensions at each step, restoring to [64, 1, 224, 224]. Each layer includes ReLU activation and batch normalization, with Tanh activation in the final layers. This



(a)



(b)

Fig. 2: Proposed architecture. (a) GAN based architecture (b) proposed channel-wise encoder-decoder network.

architecture effectively captures and reconstructs detailed features, focusing on contour information.

The discriminator network, detailed in Table I, consists of a series of convolutional layers, each followed by LeakyReLU activations and batch normalization layers. These components help stabilize the training process and ensure faster convergence. The final layer of the discriminator is a Sigmoid activation function, which outputs a probability score indicating the likelihood that the input image is real.

TABLE I: Discriminator Network Architecture

Layer	Filters	Kernel	Stride	Activation
Conv2D	64	4	2	LeakyReLU
Conv2D	128	4	2	BatchNorm, LeakyReLU
Conv2D	256	4	2	BatchNorm, LeakyReLU
Conv2D	512	4	2	BatchNorm, LeakyReLU
Conv2D	1	4	1	Sigmoid

C. Joint Loss

To train the networks, we use a joint loss function that combines reconstruction and adversarial losses. The reconstruction loss includes L2 losses for overlapping and inpainted regions:

$$\mathcal{L}_{\text{L2, overlap}} = \|\hat{M}_{\text{overlap}} \odot (I - G(I, \hat{M}))\|_2 \quad (1)$$

$$\mathcal{L}_{\text{L2, inpaint}} = \|\hat{M}_{\text{inpaint}} \odot (I - G(I, \hat{M}))\|_2 \quad (2)$$

where $\hat{M}_{\text{overlap}}$ and $\hat{M}_{\text{inpaint}}$ are masks for the respective regions, I is the input image, $G(I, \hat{M})$ is the generated image, and $\odot$ denotes element-wise multiplication.

The GAN adversarial loss, encouraging the generated images to appear real, is defined as:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}[\log D(I, \hat{M})] + \mathbb{E}[\log(1 - D(G(I, \hat{M}), \hat{M}))] \quad (3)$$

where D is the discriminator. The total joint loss is a weighted sum:

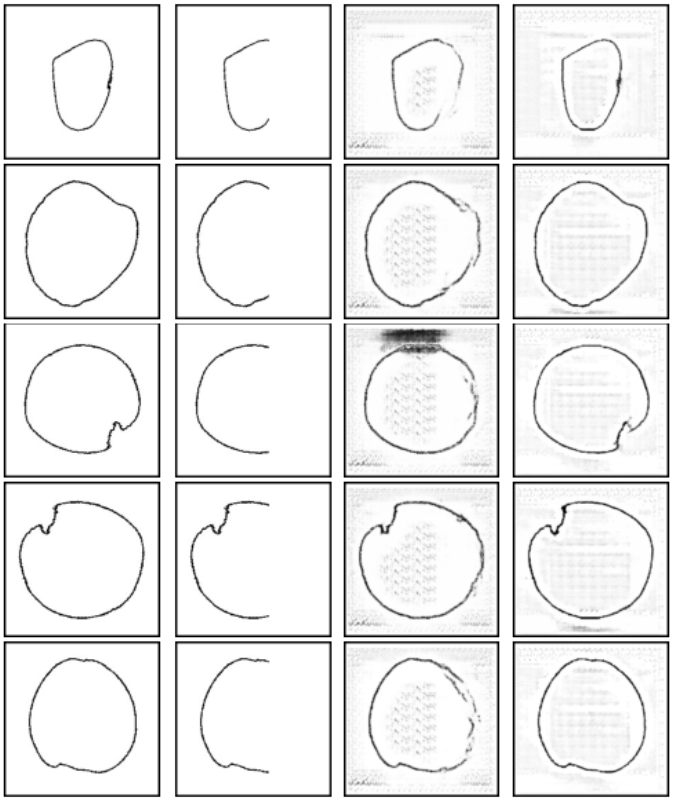
$$\mathcal{L}_{\text{joint}} = \lambda_1 \mathcal{L}_{\text{L2, overlap}} + \lambda_2 \mathcal{L}_{\text{L2, inpaint}} + \lambda_3 \mathcal{L}_{\text{GAN}} \quad (4)$$

Initially, we set λ_1, λ_2, and λ_3 to 0.9, 0.09, and 0.01 for the first 50 epochs, adjusting these values for faster convergence.

IV. Experiment and Result

The training processes both utilize a batch size of 64, with a training dataset consisting of 8,000 images. Each epoch comprises 125 iterations, resulting in a total of 18,750 iterations over 150 epochs. The qualitative comparison is shown in Fig. 3.

The reconstructed images of the baseline model in Fig. 3c exhibit several notable issues: they are significantly blurrier compared to the original images in Fig. 3a or our method Fig. 3d, indicating a lack of high-frequency detail capture. Additionally, there is visible noise and artifacts, suggesting that the baseline model has not effectively learned to cleanly inpaint the missing regions. The contours in these reconstructions are often incomplete or inaccurate, implying difficulty in restoring structural details accurately. Our method outperforms the baseline model, as evident from the improved image quality in Fig. 3d. The issues of the baseline model, potentially due to insufficient training, inadequate model complexity, or an ineffective loss function, are successfully addressed by our method within the same number of iterations, demonstrating our approach's efficiency.



(a) (b) (c) (d)

Fig. 3: Qualitative comparison with the same mask. (a) Original image. (b) masked image; (c) reconstruction images generated by baseline model; (d) reconstruction image generated by our method

For a quantitative comparison, we tested 2000 images with different mask sizes. As shown in Table II, our method outperforms the baseline model across all metrics and mask size tests.

V. Conclusion

This paper deals with the challenge of working with images that have simple contour structures but are available in a very limited dataset. To address this, we augmented the dataset and used a GAN based architecture. By increasing the number of network layers, reducing kernel size, and adjusting weight ratios led to minor improvements in convergence, we speed up convergence and improve restoration outcomes. Alternatively, we test adding penalties for contour integrity and image

TABLE II: Quantitative comparison for various mask sizes

Metric	Mask Size	Baseline Model	Our Method
MSE	112x224	0.0508	**0.0504**
	84x224	0.0464	**0.0431**
	56x224	0.0381	**0.0359**
	28x224	0.0360	**0.0342**
MAE	112x224	0.0597	**0.0538**
	84x224	0.0585	**0.0500**
	56x224	0.0531	**0.0463**
	28x224	0.0527	**0.0454**
PSNR	112x224	23.8910	**24.1940**
	84x224	24.4014	**24.8776**
	56x224	24.3310	**24.9885**
	28x224	24.3310	**24.9885**
SSIM	112x224	0.8165	**0.8407**
	84x224	0.8199	**0.8496**
	56x224	0.8266	**0.8528**
	28x224	0.8266	**0.8524**

cleanliness in the loss function, and also experiment with the Discrete Cosine Transform (DCT) to enhance the resemblance between generated images and the originals. These methods aim to produce images with cleaner and more accurate contours but fail to improve model performance.

Considering the image characteristics, further research is necessary to better capture contour features and achieve stable results with fewer iterations. Advanced techniques and architectures specifically designed for contour detection and image restoration could significantly enhance model performance and efficiency.

REFERENCES

[1] Jireh Jam, Connah Kendrick, Kevin Walker, Vincent Drouard, Jison Gee Sern Hsu, and Moi Hoon Yap. A comprehensive review of past and present image inpainting methods. *Computer Vision and Image Understanding*, 203, 2 2021.

[2] Yu Weng, Shiyu Ding, and Tong Zhou. A survey on improved GAN based image inpainting. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering, ICCECE 2022*, pages 319–322. Institute of Electrical and Electronics Engineers Inc., 2022.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[4] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, Las Vegas, NV, USA, 2016. IEEE.

[5] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22428–22437, June 2023.

[6] Wentao Wang, Li Niu, Jianfu Zhang, Xue Yang, and Liqing Zhang. Dual-path image inpainting with auxiliary GAN inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11421–11430, June 2022.

[7] Xinmiao Lin, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Yu Kong. Catch missing details: Image reconstruction with frequency augmented variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1736–1745, June 2023.

[8] S. Verma, A. Sharma, R. Sheshadri, and S. Raman. Graphfill: Deep image inpainting using graphs. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4984–4994, 2024.

[9] L. Haritha and C. A. Prajith. Image inpainting using deep learning techniques: A review. In *2023 International Conference on Control, Communication and Computing, ICCC 2023*. Institute of Electrical and Electronics Engineers Inc., 2023.