Image Inpainting Techniques for Mobile and High-Resolution Environments

2025년도 동계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
Haeuk Lee

1) SARGSYAN, Andranik, et al. Mi-gan: A simple baseline for image inpainting on mobile devices. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023. p. 7335-7345.
2) LIU, Weihuang, et al. Coordfill: Efficient high-resolution image inpainting via parameterized coordinate querying. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2023. p. 1746-1754.

Outline

- MI-GAN: A Simple Baseline for Image Inpainting on Mobile Devices[1)]
  - ICCV 2023

- CoordFill: Efficient High-Resolution Image Inpainting via Parameterized Coordinate Querying[2)]
  - AAAI 2023 Oral

- MI-GAN: A Simple Baseline for Image Inpainting on Mobile Devices[1)]
  - ICCV 2023

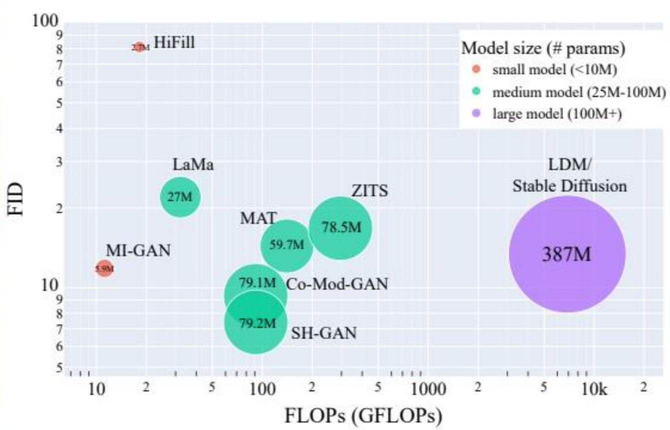
Contribution

- Introduced MI-GAN, the first lightweight generative image inpainting model specifically optimized for mobile devices
  - Designed for efficient computation and deployment on low-end hardware
- Developed a unique combination of techniques for high-quality inpainting
  - Adversarial Training
    - Ensures visually plausible results without artifacts
  - Model Re-parameterization
    - Improves output quality while maintaining efficiency
  - Knowledge Distillation
    - Enhances generative ability by learning from larger networks
- Achieved competitive or superior inpainting performance compared to state-of-the-art models
  - Demonstrated significant improvements in speed and size
  - Human evaluators preferred MI-GAN over commercial mobile applications

1) SARGSYAN, Andranik, et al. Mi-gan: A simple baseline for image inpainting on mobile devices. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023. p. 7335-7345.

Introduction

- Image inpainting: Restores missing regions in an image to produce realistic outputs
- Mobile apps like Photoshop Express, Picsart, Snapseed offer object removal tools.
- Limitations of current state-of-the-art inpainting models
  - Heavy computation unsuitable for mobile devices
  - Dependence on server-side processing (internet, latency issues)
- Proposed solution: MI-GAN
  - A lightweight, high-quality inpainting model optimized for mobile devices



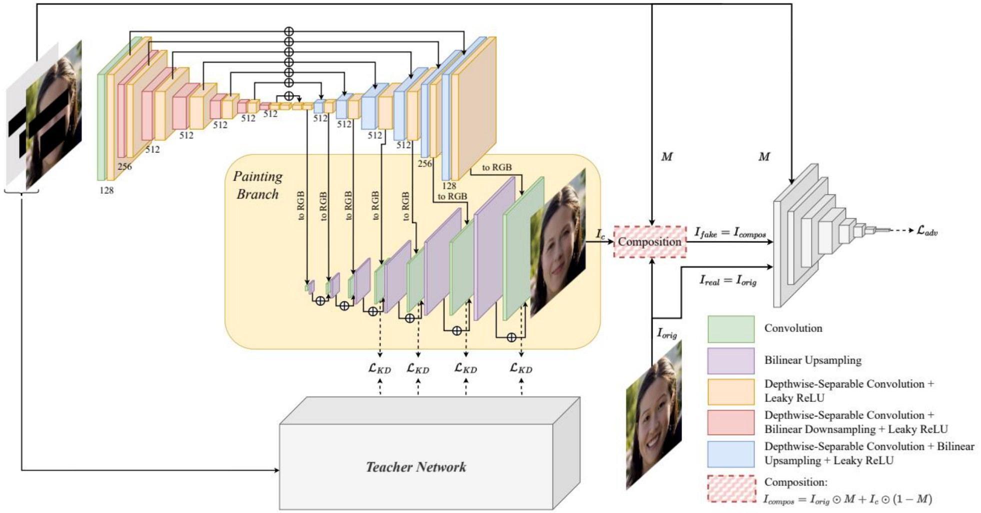
Related Work

- Traditional methods
  - Diffusion-based (e.g., curvature-driven diffusions)
  - Exemplar-based (e.g., PatchMatch, texture synthesis)
  - Lacked semantic understanding
- Deep learning advancements
  - GAN-based approaches (e.g., Co-Mod-GAN, SH-GAN)
  - Transformer-based models (e.g., MAT, ZITS)
  - Diffusion models for inpainting (e.g., Palette, LDM)
- Gaps: None of these models are designed for efficient deployment on mobile devices

1) SARGSYAN, Andranik, et al. Mi-gan: A simple baseline for image inpainting on mobile devices. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023. p. 7335-7345.

Method Overview

- MI-GAN integrates three key techniques
  - Adversarial Training: Ensures visually plausible results without artifacts
  - Model Re-parameterization: Enhances model efficiency and quality
  - Knowledge Distillation: Leverages a larger model (Co-Mod-GAN) to improve generative ability
- Combines a main branch and a painting branch
  - Main branch: U-Net-like structure for core inpainting tasks
  - Painting branch: Mimics iterative expert inpainting processes



Architecture Details

- Main Branch
  - U-Net-like structure with depthwise-separable convolutions
  - Bilinear upsampling/downsampling for feature extraction and resolution changes
  - Incorporates random noise for high-frequency detail generation
- Painting Branch
  - Completes missing regions layer-by-layer in the RGB space
  - Combines intermediate outputs to form the final image
  - The inpainting composition equation

-$I_{compos} = I_{orig} \odot M + I_c \odot (1 - M)$

Key Techniques

- Adversarial Training
  - Inspired by StyleGAN discriminator
  - Ensures non-blurry and artifact-free results
  - Adversarial loss for the generator

$$-L_{adv} = \mathbb{E}_{\{x,m \sim P_{\{c,m\}}\}}[\,SoftPlus(-D_W(G_\theta(x,m),m))]$$

- Knowledge Distillation
  - Teacher: Co-Mod-GAN with strong generative ability
  - Transfers intermediate outputs from teacher to MI-GAN
  - Knowledge Distillation Loss

$$-L_{KD} = \sum_{i=0}^{3} |(x_i - x_i^C) \odot (1 - M_i)|$$

- Model Re-parameterization
  - Optimizes convolutional layers for mobile hardware
  - Improves efficiency without sacrificing quality

Quantitative Results

- Results on Places2 (256x256 resolution)
  - MI-GAN achieves competitive FID and LPIPS compared to Co-Mod-GAN and SH-GAN
  - Significantly faster and lighter
    - 8x faster and 13x smaller than Co-Mod-GAN
- Results on FFHQ (face images)
  - MI-GAN performs comparably to Co-Mod-GAN in FID and LPIPS
  - Outperforms LaMa, ZITS, and HiFill

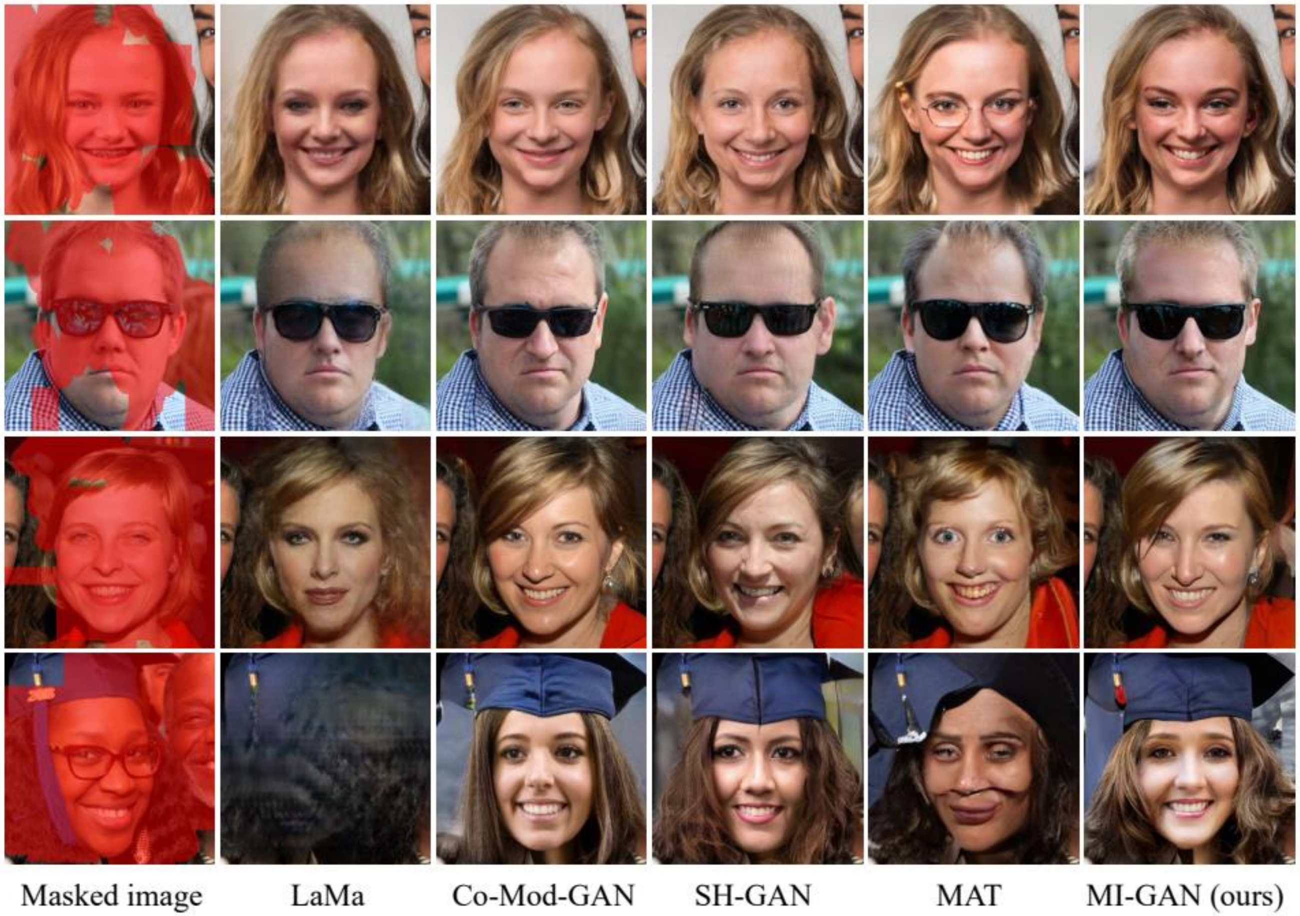
	FFHQ		**Places2**		FLOPS	Params
Method	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	(GFLOPS)	($\times 10^6$)
LaMa	32.71	0.259	22.00	0.378	32.05	27.05
Co-Mod-GAN	4.70	0.257	9.32	0.397	91.21	79.17
SH-GAN	**4.33**	0.254	**7.40**	0.392	91.27	79.21
ZITS	-	-	16.78	**0.356**	295.72	78.49
MAT	7.00	**0.231**	14.38	0.394	140.74	59.78
LDM	-	-	13.40	0.385	6,896.16	387.25
HiFill	-	-	81.27	0.488	18.14	**2.72**
MI-GAN (ours)	4.99	0.257	11.83	0.394	**11.19**	5.95

Quantitative comparison on 256 resolution images

Qualitative Results

Qualitative results on 256 resolution Places 2 samples

Qualitative Results



Qualitative results on FFHQ samples

Qualitative Results

- Real-world testing on devices (256x256 resolution)
  - MI-GAN runs 4x faster on average than Co-Mod-GAN
- Speed comparison on popular devices
  - Example: iPhone 7 – MI-GAN: 1030 ms vs. Co-Mod-GAN: 4475 ms

Device Name	256-resolution MI-GAN speed (ms, mean/std)	256-resolution Co-Mod-GAN speed (ms, mean/std)
iPhone7	1030.25 / 13.37	4475.33 / 39.55
iPhoneX	630.80 / 12.21	2746.00 / 28.84
iPad mini (5th gen)	552.40 / 8.10	2686.17 / 41.41
iPhone14-pro-max	296.00 / 1.35	1374.40 / 84.78
Galaxy Tab S7+	686.17 / 12.36	- / -
Samsung Galaxy S8	1476.40 / 5.98	- / -
vivo Y12	2918.08 / 33.47	- / -

Qualitative results on FFHQ samples

- CoordFill: Efficient High-Resolution Image Inpainting via Parameterized Coordinate Querying[1)]
  - AAAI 2023 Oral

Contribution

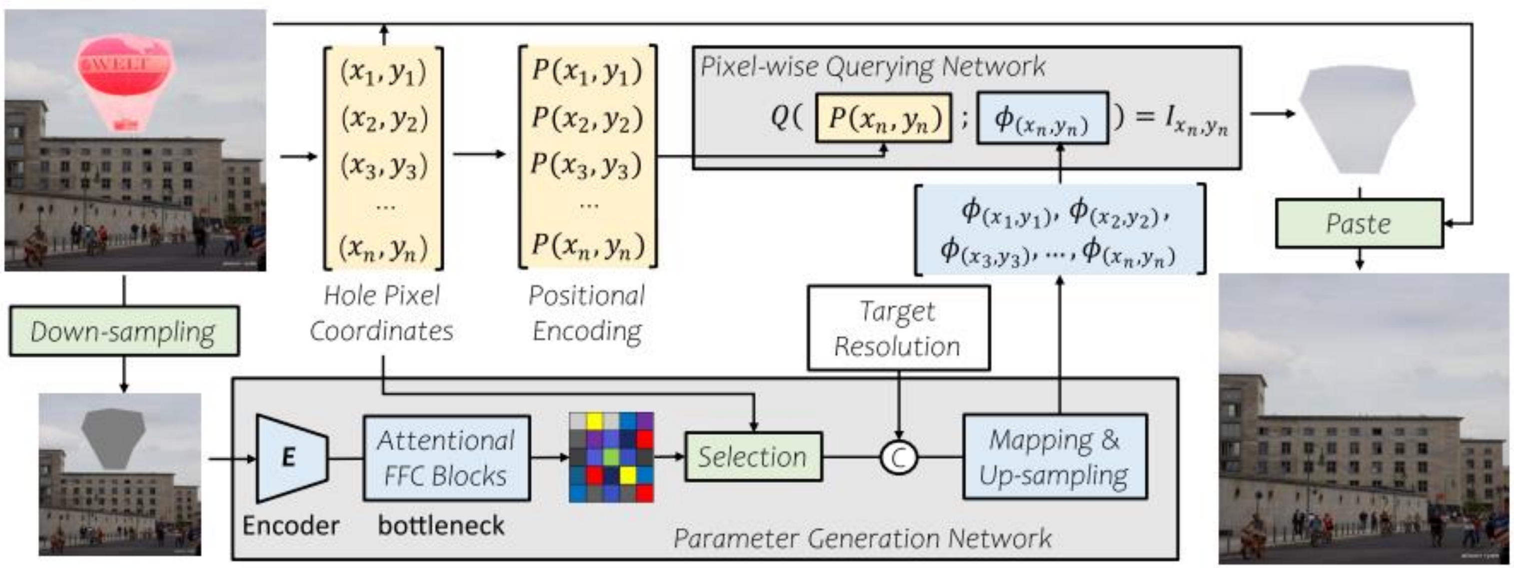
- Proposed CoordFill, a novel framework for efficient high-resolution image inpainting
  - Utilizes parameterized coordinate querying to address computational inefficiencies
- Designed an Attentional FFC-based block
  - Learns to focus automatically on the masked regions
  - Enhances the spatial understanding required for high-resolution inpainting
- Introduced a pixel-wise querying network
  - Generates pixel values for only the masked regions using positional encoding
  - Significantly reduces unnecessary computation
- Achieved state-of-the-art performance
  - Faster and more efficient than existing methods
  - Demonstrates superior qualitative and quantitative results across multiple datasets

Introduction

- Challenges in high-resolution inpainting
    - Requires large receptive fields for contextual understanding
    - Inefficient computation by processing unnecessary regions
- Existing methods rely on CNNs, GANs, or transformers (e.g., MAT, ZITS)
    - High computational cost, limited scalability for very high resolutions
- CoordFill: Efficiently addresses these challenges with coordinate querying

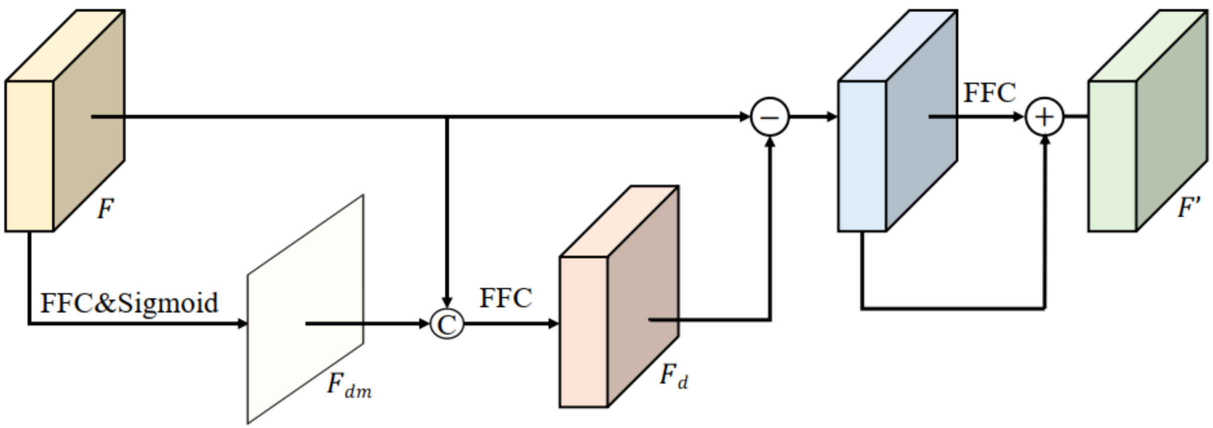
Method Overview

- CoordFill solves high-resolution image inpainting with two key components
  - Parameter Generation Network (PGN)
    - Produces parameters for spatially adaptive reconstruction
  - Pixel-wise Query Network (PQN)
    - Generates pixel values for masked regions using positional encoding
- Key advantages
  - Only processes masked regions, avoiding unnecessary computations
  - Supports arbitrary resolution with continuous coordinate querying



Parameter Generation Network (PGN)

- Downsamples high-resolution input to a lower resolution (e.g., 256×256)
- Uses Attentional FFC blocks to capture spatial and frequency domain features
  - Removes spatial noise and enhances relevant features
  - Formula for spatial attention map
    - $F_{\mathrm{dm}} = \sigma\big(\mathrm{FFC}(F)\big)$
- Generates spatially adaptive parameters for masked regions
  - Incorporates target resolution for flexible upsampling
  - Final parameters guide pixel-wise querying in PQN



The details of the proposed AttFFC Block

Pixel-wise Query Network (PQN)

- Utilizes parameters generated by PGN to reconstruct pixel values
  - Formula for pixel value prediction
    - $y_p = Q(p; \Phi_p)$
  - Q: Multi-layer perceptron (MLP) trained on positional encoding
- Encodes pixel positions using sinusoidal functions for high-frequency details
  - Formula for positional encoding
    - $p = \left(\sin\left(\frac{2\pi p_x}{E_x}\right), \cos\left(\frac{2\pi p_x}{E_x}\right), \sin\left(\frac{2\pi p_y}{E_y}\right), \cos\left(\frac{2\pi p_y}{E_y}\right)\right)$
- Processes only masked regions, reducing computation time significantly

Training Process

- Loss functions
  - Perceptual loss: Measures feature differences in a pre-trained AlexNet

    - $L_{\text{per}} = \sum_k \tau_k \left(E_k(I_o) - E_k(I_{\text{gt}}) \right)$
  - Adversarial loss: Encourages realism in restored regions

    - $L_{\text{adv}} = \mathbb{E}\left[\log\left(1 - D(I_o)\right)\right] + \mathbb{E}\left[\log D\left(I_{\text{gt}}\right)\right]$
  - Feature matching loss: Stabilizes GAN training

    - $L_{\text{fm}} = \sum_i \left(D^i(I_{\text{gt}}), D^i(I_o) \right)$

- Total loss
  - $L_{\text{total}} = \lambda_{\text{per}} L_{\text{per}} + \lambda_{\text{adv}} L_{\text{adv}} + \lambda_{\text{fm}} L_{\text{fm}}$

Quantitative Results

- Results on Places2 (512×512 to 4096×4096 resolutions)
  - CoordFill outperforms HiFill, LaMa, and ZITS in PSNR and SSIM
  - Maintains competitive LPIPS scores
  - Fastest inference time across all resolutions (e.g., 10ms at 512×512)

Resolution	512×512				1024×1024			
	PSNR↑	SSIM↑	LPIPS↓	SPEED↓	PSNR↑	SSIM↑	LPIPS↓	SPEED↓
DeepFillv2 (Yu et al. 2019)	23.973	0.902	0.080	398ms	22.695	0.908	0.092	1002ms
HiFill (Yi et al. 2020)	23.375	0.883	0.097	406ms	23.456	0.894	0.096	423ms
RN (Yu et al. 2020)	22.562	0.880	0.116	17ms	19.587	0.879	0.139	59ms
CR-Fill (Zeng et al. 2021)	24.216	0.893	0.086	46ms	22.881	0.890	0.108	54ms
LaMa (Suvorov et al. 2022)	26.203	**0.914**	**0.067**	27ms	26.154	**0.924**	0.076	142ms
MAT (Li et al. 2022)	24.169	0.900	0.076	71ms	23.751	0.908	0.082	133ms
ZITS (Dong et al. 2022)	26.349	0.911	0.068	183ms	**26.389**	0.913	**0.073**	462ms
CoordFill	**26.365**	<u>0.912</u>	<u>0.068</u>	**10ms**	<u>26.322</u>	<u>0.920</u>	<u>0.075</u>	**14ms**

Resolution	2048×2048				4096×4096			
	PSNR↑	SSIM↑	LPIPS↓	SPEED↓	PSNR↑	SSIM↑	LPIPS↓	SPEED↓
DeepFillv2 (Yu et al. 2019)	-	-	-	-	-	-	-	-
HiFill (Yi et al. 2020)	23.643	0.915	0.087	478ms	23.634	0.933	0.077	662ms
RN (Yu et al. 2020)	18.843	0.908	0.143	240ms	-	-	-	-
CR-Fill (Zeng et al. 2021)	22.056	0.908	0.122	63ms	-	-	-	-
LaMa (Suvorov et al. 2022)	25.688	**0.939**	0.078	598ms	-	-	-	-
MAT (Li et al. 2022)	-	-	-	-	-	-	-	-
ZITS (Dong et al. 2022)	-	-	-	-	-	-	-	-
CoordFill	**26.322**	<u>0.932</u>	**0.077**	**26ms**	**26.175**	**0.943**	**0.075**	**78ms**

Comparison on Places2 Dataset on different resolutions

Qualitative Results

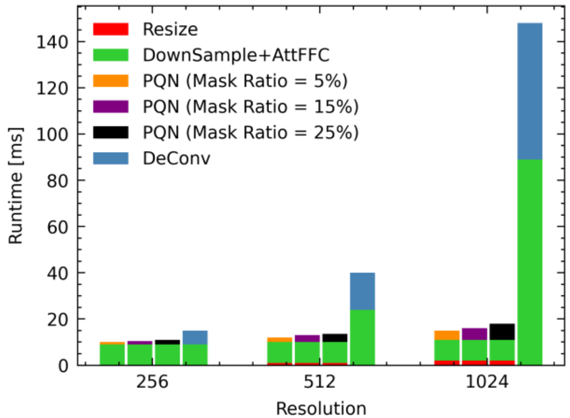
- Visual comparisons with state-of-the-art methods
  - CoordFill produces sharper textures and consistent colors
  - Handles large masked regions better than LaMa and ZITS

(a) Input (b) GT (c) DFillv2 (d) HiFill (e) RN (f) CR-Fill (g) LaMa (h) MAT (i) ZITS (j) CoordFill

Comparison on Places2 Dataset and Unsplash dataset

Efficiency Analysis

- CoordFill is optimized for high-resolution inpainting
  - Handles up to 4096×4096 resolutions without memory issues
  - Processes masked regions only, reducing computational cost
- Speed comparison
  - Faster than LaMa, ZITS, and HiFill at all resolutions
  - Example: 14ms for 1024×1024 compared to LaMa (142ms)
- Flexibility
  - Inference time scales with mask size, enabling efficient partial reconstructions



The speed comparison of the proposed method (three different mask ratios) and the baseline (DownSample + AttFFC + DConv) on the three different resolutions