InP-Net: Hybrid Autoencoder-Transformer with Masked Attention for Image Inpainting

P.V.V.S. Aditya
M.Tech AI
IIT Jodhpur, Dept. CSE
m24csa018@iitj.ac.in

G. Mohith Nukesh
M.Tech AI
IIT Jodhpur, Dept. CSE
m24csa037@iitj.ac.in

S. L. Shanmukha
M.Tech AR VR
IIT Jodhpur, School of AIDE
m24air013@iitj.ac.in

Abstract

In this work, we propose InP-Net, a hybrid architecture that integrates Autoencoders and Vision Transformers (ViTs) to address the task of image inpainting. The novelty lies in combining the global contextual understanding of transformers with the spatial reconstruction capabilities of autoencoders. Our model employs masked encoding strategies, including random and structured masks such as center and corner-based masking (via a center-dict). We reduce computational complexity by selectively processing only visible patches while maintaining high contextual richness. A major enhancement over traditional approaches is the flexibility in masking ratio, which can be set from 0% to 90% in our framework, unlike the fixed 75% masking ratio in the original Facebook MAE. This not only enhances training generalization across mask types but also makes our model more robust during inference, where setting the mask ratio to 0 enables the model to auto-reconstruct lost regions directly from the input image.

We further incorporate skip connections to enhance gradient flow and representation learning, alongside a lightweight decoder for high-fidelity image reconstruction. Using self-supervised training, our method avoids dependency on labeled datasets. Empirical results demonstrate that InP-Net outperforms baseline MAE models in terms of reconstruction quality and visual realism, making it a scalable and efficient solution for image inpainting and restoration.

1 Introduction

The ability to automatically recover missing or occluded regions in images is a longstanding challenge in computer vision. This task, known as image inpainting, has applications ranging from old photo restoration and object removal to scene completion and image editing. As image resolutions increase and real-world occlusions become more complex, there is a growing demand for models that can perform high-fidelity inpainting while being efficient and robust.

Conventional inpainting models, particularly those based on convolutional neural networks (CNNs), have made significant progress by learning spatial priors and texture continuity. However, their reliance on local receptive fields often limits their ability to understand and reconstruct semantically meaningful content across large image regions. To overcome this, researchers have increasingly turned to self-attention mechanisms and transformers, which naturally capture long-range dependencies. While these methods offer improved contextual awareness, they also come with higher computational costs and are often less effective at reconstructing low-level visual details.

Recent transformer-based inpainting approaches, such as Masked Autoencoders (MAEs), learn to reconstruct masked image patches using self-supervised learning. Although effective in learning global representations, these models typically employ fixed masking strategies and large transformer encoders, which may not generalize well across diverse occlusion patterns and can be inefficient for practical deployment.

In this work, we aim to bridge the gap between global reasoning and local detail preservation by designing a hybrid framework that combines the best of both worlds transformers for high-level semantics and autoencoders for efficient spatial reconstruction. Our motivation stems from the observation that while transformers provide powerful contextual embeddings, their output can benefit from the refinement capabilities of decoder-based architectures with residual connections and spatial priors.

By addressing challenges in scalability, flexibility, and visual quality, our goal is to develop an architecture that not only performs robustly across a wide range of mask patterns and ratios but also adapts well during inference without requiring labels or supervision. This sets the foundation for our proposed InP-Net, which we explore in detail in the subsequent sections.

2 Literature Review

The field of image inpainting has evolved significantly over the past two decades, moving from traditional diffusion-based techniques to advanced deep learning models capable of understanding semantic content and generating visually realistic completions.

Convolutional Autoencoders (CAEs) have long been a foundational architecture for tasks like image reconstruction, denoising, and compression. These models typically consist of an encoder that compresses the input into a latent representation and a decoder that reconstructs the original image from this embedding. While CAEs are effective at capturing low-level spatial features and textures, they are inherently limited by their local receptive fields. This constraint often leads to blurry reconstructions when dealing with large or complex missing regions, as the models lack a global understanding of the scene.

With the introduction of Vision Transformers (ViTs) [Dosovitskiy et al., 2020], attention mechanisms have become a powerful tool for learning global dependencies in visual data. The Masked Autoencoder (MAE) framework proposed by He et al. (2021) leverages ViTs to perform self-supervised representation learning by randomly masking image patches and training the model to reconstruct the missing regions. This approach enables models to learn high-level semantic features from vast amounts of unlabeled data. However, MAEs have certain limitations: they often reconstruct only the masked patches (not the full image), rely on a large transformer backbone, and require a relatively heavy decoder to achieve detailed outputs. Moreover, the default 75% fixed random masking strategy can hinder generalization when encountering different types of occlusions during inference.

Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] have been widely used in image inpainting to enhance visual realism. Models like Context Encoder [Pathak et al., 2016] and EdgeConnect [Nazeri et al., 2019] use adversarial training to encourage the generator to produce plausible image completions. While GANs are effective in generating sharp and photorealistic images, they come with challenges such as training instability, mode collapse, and difficulty in maintaining global semantic consistency, especially when the missing region is large or complex.

More recently, diffusion-based models like DDPM [Ho et al., 2020] have gained traction for image generation and inpainting. These models gradually refine noise into a coherent image through a sequence of denoising steps. Although they offer impressive image quality, diffusion models are typically slow to sample and computationally expensive due to their iterative nature, which limits their scalability for real-time or resource-constrained applications.

In an effort to combine the local precision of CNNs with the global reasoning of transformers, several hybrid architectures have been proposed. For instance, TransUNet [Chen et al., 2021] combines convolutional encoders with transformer bottlenecks for medical image segmentation. Similarly, models like Uformer [Wang et al., 2022] explore hierarchical transformer-based encoders within a U-Net structure. However, many of these hybrid models are either too large for practical deployment or lack architectural coherence when switching between convolutional and attention-based modules. Moreover, they often rely on supervised learning, limiting their adaptability across domains.

2.1 Summary and Motivation

The existing literature highlights a clear trade-off between efficiency and performance in image inpainting. CNN-based models offer speed and spatial fidelity but lack semantic understanding.

On the other hand, transformer-based models provide powerful contextual learning at the cost of computational overhead and architectural complexity. Few existing solutions manage to blend these paradigms effectively within a lightweight, scalable, and self-supervised framework.

InP-Net builds upon these insights by integrating autoencoder structures with masked transformer-based encoders in a principled manner. It introduces flexible masking strategies (random, structured), skip connections for effective gradient propagation, and a lightweight decoder designed for high-fidelity reconstruction. Unlike prior MAE-based methods, InP-Net allows dynamic masking ratios (including 0% at inference), enhancing adaptability without sacrificing efficiency.

3 Problem Statement

Image inpainting is a challenging task that requires a model to infer and restore missing or corrupted regions in an image in a perceptually convincing manner. While transformer-based models particularly Masked Autoencoders (MAEs) have demonstrated strong capabilities in modeling global semantic context, their application to full-resolution images often results in high computational costs and inefficiencies. Moreover, many existing approaches rely on fixed masking ratios and random masking patterns, which limit their generalization across diverse occlusion scenarios during inference.

A further limitation of transformer-based inpainting is the lack of spatial inductive biases, making it difficult for such models to preserve local textures and fine-grained visual details. On the other hand, convolutional models, although effective at capturing local patterns, struggle to model long-range dependencies and semantic consistency over large missing regions.

There is a growing need for a hybrid model that can leverage the strengths of both convolutional and transformer-based architectures, while addressing their individual shortcomings. Specifically, the problem we address in this work is to design an image inpainting framework that can:

- **Reconstruct masked regions with high fidelity** while maintaining semantic consistency across the image.
- **Preserve local textures and structures**, especially in regions near the missing patches.
- **Scale efficiently to high-resolution images** without incurring prohibitive computational costs.
- **Support dynamic and structured masking strategies**, enabling flexible training and robust inference across a wide range of occlusion types and ratios.

Our goal is to build a model that not only produces visually realistic inpainted images but also generalizes well to real-world conditions where occlusions may be irregular, structured, or unpredictable in size and shape.

4 Proposed Method

We propose **InP-Net**, a hybrid architecture that combines the spatial reconstruction capabilities of convolutional autoencoders with the global contextual reasoning strength of Vision Transformers (ViTs). This section details both the baseline model (MAE by Facebook AI) and our proposed improvements, followed by an overview of our architectural variants.

4.1 Original MAE (Facebook ViT-MAE)

The Masked Autoencoder (MAE) is a self-supervised learning model that reconstructs missing image patches from a randomly masked input. Key components include:

- **Random Masking:** Roughly 75% of image patches are randomly masked during training.
- **ViT Encoder:** Only visible (unmasked) patches are processed by the Vision Transformer encoder.
- **Transformer Decoder:** A heavy transformer-based decoder reconstructs the full image by predicting the masked patches.

- **Loss Function:** Mean Squared Error (MSE) is used to compute the pixel-wise reconstruction loss between predicted and original image patches.

Formally, the reconstruction loss is:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{x}_i - x_i\|^2,$$

where $\hat{x}_i$ is the reconstructed patch and x_i is the ground truth patch for position i.

4.2 Our Enhancements

To address the limitations of the original MAE (e.g., loss of local features, fixed masking, heavy decoder), we propose the following improvements.

4.2.1 Masking Strategies

We extend the masking methodology beyond random patch selection. In addition to stochastic masking, we define deterministic structured masking using a dictionary of location presets:

```
center_dict = {
    'top_left':     [0, 0],
    'top_right':    [0, w],
    'bottom_left':  [h, 0],
    'bottom_right': [h, w],
    'top_middle':   [0, w/2],
    'center_middle':[h/2, w/2],
    'center_left':  [h/2, 0],
    'center_right': [h/2, w],
    'bottom_center':[h, w/2],
}
```

The masking ratio is configurable in the range **0% to 90%**, providing flexibility during both training and inference. During training, we optionally apply **mask ratio scheduling** or **mask warm-up**, gradually increasing the masking percentage across epochs to stabilize learning.

4.2.2 Patch Embedding and Positional Embedding

Input images of size $H \times W \times C$ are divided into non-overlapping patches of size $p \times p$, resulting in $N = \frac{H \cdot W}{p^2}$ total patches. Each patch is flattened and linearly projected into a feature embedding of dimension D. Let $\mathbf{x}_i \in \mathbb{R}^{p^2 \cdot C}$ be the vectorized representation of the i-th patch. The patch embedding is computed as:

$$\mathbf{z}_i = \mathbf{W}_e \cdot \mathbf{x}_i + \mathbf{b}_e,$$

where $\mathbf{W}_e \in \mathbb{R}^{D \times (p^2 C)}$ is the learnable patch embedding matrix. To encode spatial information lost during patch flattening, learnable positional embeddings $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N \times D}$ are added to each patch embedding:

$$\mathbf{z}_i^{(0)} = \mathbf{z}_i + \mathbf{E}_{\text{pos},i}.$$

This combined representation $\mathbf{z}_i^{(0)}$ is then passed as input to the transformer encoder. The positional embedding allows the model to preserve the spatial layout of the image patches, which is crucial for capturing contextual relationships during the inpainting process.

4.2.3 Transformer Encoder

The encoded patch tokens are passed through a standard Vision Transformer (ViT) consisting of L transformer blocks. Each block includes Multi-head Self-Attention (MSA) and Feed-Forward Network (FFN) layers with LayerNorm and residual connections.

Self-attention is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

where Q, K, V are projections of the input embeddings. The transformer encoder captures long-range dependencies and contextual patterns among visible patches.

4.2.4 Skip Connections

Inspired by U-Net and encoder-decoder segmentation architectures, we route intermediate transformer features to the decoder using skip connections. These lateral connections enhance:

- **Gradient flow**, aiding deep network training.
- **Feature fusion**, by combining hierarchical representations.
- **Texture recovery**, especially beneficial for reconstructing local details.

4.2.5 Lightweight Decoder

Instead of using a heavy transformer decoder, we employ a shallow convolutional decoder comprising:

- Transposed convolution (deconvolution) layers for upsampling.
- Batch Normalization and ReLU activations.
- Residual blocks to refine texture and reduce checkerboard artifacts.

This decoder design reduces inference latency while maintaining high visual fidelity in reconstructed images.

4.2.6 Training Losses

We utilize the following loss functions:

- **Reconstruction Loss (MSE):** For training with only pixel-level supervision.
- **Adversarial Loss (GAN Loss):** When using a discriminator D, we add adversarial training to improve perceptual quality.

The total loss is given by:

$$\mathcal{L}_{\text{Total}} = \lambda_1 \mathcal{L}_{\text{MSE}} + \lambda_2 \mathcal{L}_{\text{GAN}},$$

where λ_1 and λ_2 are tunable weights to balance reconstruction and realism objectives.

4.3 Model Variants

To systematically evaluate our design choices, we implement several variants of InP-Net:

1. **Model + MAE (Random mask)+ Recon Loss - Original Model**
2. **Model + MAE (Random mask) + GAN Loss**
3. **Model + MAE (Dynamic mask) + Recon Loss**
4. **Model + MAE (Dynamic mask) + GAN Loss**
5. **Model + MAE (Dynamic mask) + Skip + Recon Loss**
6. **Model + MAE (Dynamic mask) + Skip + GAN Loss**

Each variant is realized through modular configuration in the `inference()` routine, using flags for masking, skip connections, and loss types.

4.4 Flexible Inference Modes

Unlike traditional MAE models that always assume a fixed mask ratio, our design enables flexible inference with any mask ratio. Notably, when the ratio is set to 0, the model performs **full image reconstruction**, acting like a feed-forward autoencoder without masked supervision. This makes InP-Net applicable to real-world scenarios such as image restoration, enhancement, or denoising, not just inpainting.

4.5 Architecture Overview

The full InP-Net architecture consists of the following stages:

- **Patch Embedding:** Convert image into patch tokens.
- **Transformer Encoder:** Extract global context from unmasked tokens.
- **Skip Connections:** Connect mid-layer encoder outputs to decoder stages.
- **Lightweight Decoder:** Reconstruct image via upsampling and refinement.

The modularity of our architecture (as illustrated in **Figure 1**) allows for rapid experimentation, ablation studies, and extensibility to tasks beyond inpainting.

5 Experimental Setup

This section provides a comprehensive overview of the experimental setup used for training and evaluating InP-Net. We focus on the dataset, the training process, implementation environment, and the model's evaluation strategy.

5.1 Dataset Description

For the image inpainting task, we use the **Places365** dataset. It is a large-scale dataset designed for scene recognition but is well-suited for image inpainting due to the diversity and richness of its images. The dataset contains 1.8 million images from 365 different scene categories, including outdoor scenes, interiors, and more.

Dataset Characteristics:

- **Total Images**: 1.8 million images from 365 scene categories.
- **Image Resolution**: Most images are 256x256, but the dataset provides images in multiple resolutions.
- **Categories**: The dataset covers various classes, including:
  - **Urban**: City streets, buildings, and plazas.
  - **Natural**: Forests, mountains, beaches, etc.
  - **Indoor**: Rooms, kitchens, hallways, etc.
  - **Landscapes**: Open fields, water bodies, etc.

Preprocessing:

- **Resize**: Images are resized to 224x224 pixels for uniformity across the dataset, which is common for Vision Transformer models.
- **Normalization**: Images are normalized to a range of [0, 1] by dividing pixel values by 255.
- **Augmentation**: Standard augmentations are applied during training, including random horizontal flips, rotations, and cropping to increase model generalization.

Classes in Places365: Here's a brief look at some example scene categories (classes) in the dataset:

- **Bedroom**
- **Living Room**

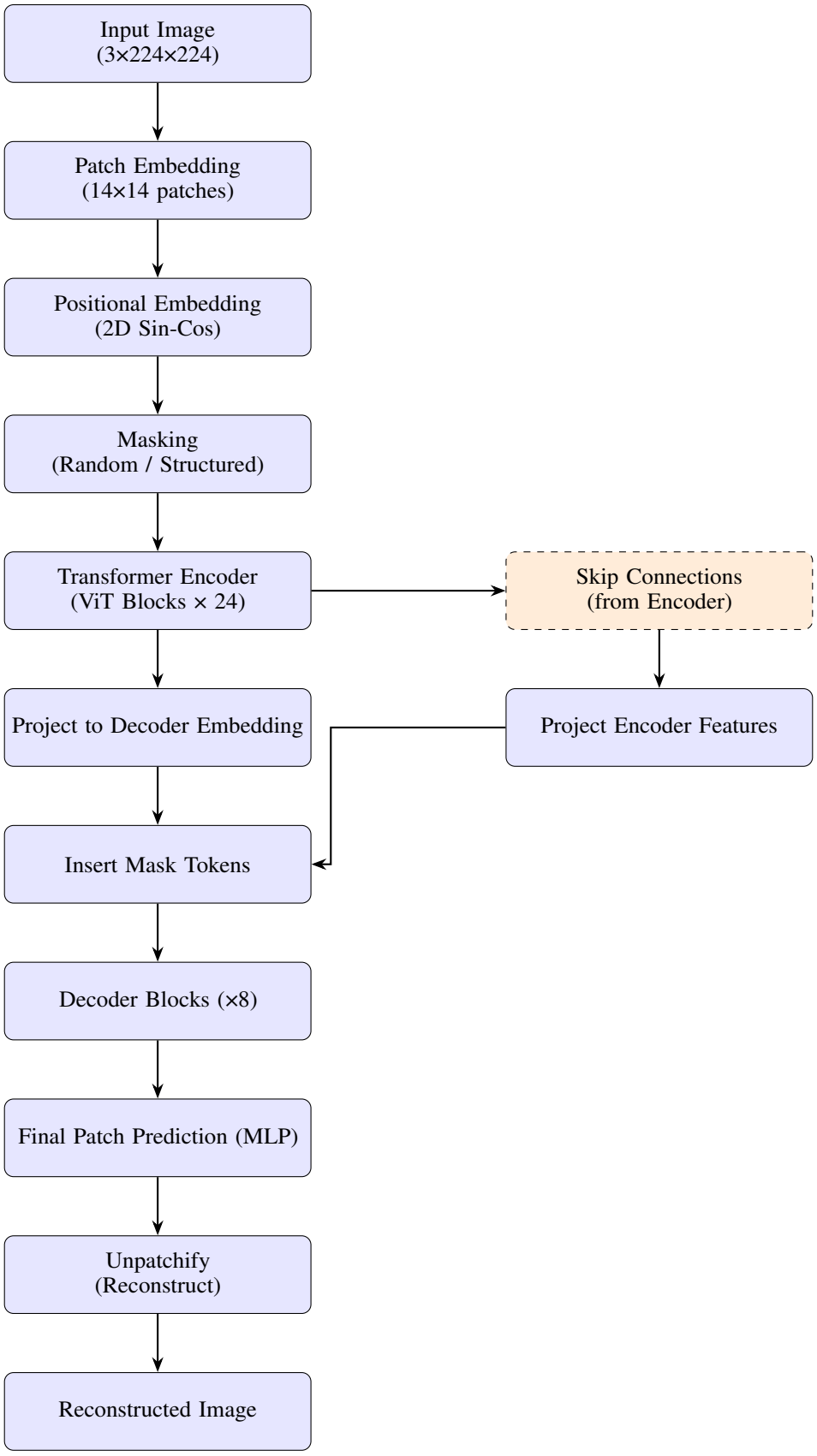


Figure 1: InP-Net architecture

- **Office**
- **Store**
- **Restaurant**
- **Beach**
- **Mountain**
- **Forest**
- **Kitchen**
- **Train Station**

These categories help the model to learn a broad range of scene types, which can be useful when dealing with varied types of images in real-world applications.

In the context of inpainting, each image serves as a source for training the model to infer missing or corrupted regions based on the surrounding context. By using the large variety of scenes, we ensure that our model is robust across different image contexts and environments.

5.2 Training Details

The training process for InP-Net involves several critical components:

Optimizer:

- **Optimizer**: AdamW (Adaptive Moment Estimation with weight decay).
- **Learning Rate**: Set initially to 1e-3, with a cosine annealing schedule for gradual reduction during training.
- **Batch Size**: We use a batch size of 32 images for efficient training.
- **Epochs**: The training is run for 30 epochs to ensure convergence.
- **Weight Decay**: A weight decay of 1e-4 is applied to prevent overfitting.

Loss Functions:

- **Reconstruction Loss (MSE)**: For pixel-level similarity, we use Mean Squared Error to measure the difference between the reconstructed and the ground truth images.
- **GAN Loss**: For variants with GAN, we use the adversarial loss to promote realistic texture and finer details.
- **Combined Loss**: For certain variants, we combine both reconstruction loss and GAN loss to balance pixel-level fidelity with texture realism.

Hardware: The model was trained on a Google Colab instance with GPU (NVIDIA Tesla K80/ P100/V100), leveraging the free GPU resources provided by Google Colab for fast training. Each epoch takes approximately 20 minutes on this setup.

Training Time: Each epoch takes approximately 10 minutes, and the total training time depends on the number of epochs and the batch size.

Training Strategy:

- **Early Stopping**: Training halts if validation performance stops improving for 5 consecutive epochs.
- **Data Augmentation**: To increase robustness, we perform random image augmentations like horizontal flipping and random cropping.
- **Masking Ratios**: We experiment with varying masking ratios, from 0% to 90% masking of patches, and compare performance across these settings. The mask ratio influences the reconstruction quality and generalization ability.

5.3 Implementation Environment

The implementation of InP-Net is based on the following setup:

- **Framework**: PyTorch 1.9.0, utilizing the torchvision library for pre-trained models and image transformations.
- **Libraries**:
  - **PyTorch**: For defining the neural network architecture, loss functions, and optimizer.
  - **torchvision**: Used for image transformations (e.g., resizing, normalization).
  - **Matplotlib/Seaborn**: For visualizing results and loss curves.
  - **CUDA**: GPU acceleration with CUDA 11.2 for faster training.
  - **TensorBoard**: For visualizing training and validation metrics.
- **Custom Modules**: Custom code is written for:
  - Mask generation based on structured or random masking.
  - Skip connections from intermediate layers of the Vision Transformer (ViT) to the decoder.
  - Lightweight convolutional decoder.

Custom Network Modules:

- **Vision Transformer (ViT)**: Used for encoding visible patches. A pre-trained ViT backbone is fine-tuned for inpainting tasks.
- **Skip Connections**: Features from intermediate ViT layers are passed directly to the decoder to aid in gradient flow and capture finer spatial features.
- **Lightweight Decoder**: We use a shallow CNN decoder to reduce computational complexity while maintaining high reconstruction quality.
- **Masking Strategies**: Our framework supports both **random masking** (patch-level) and **structured masking** (e.g., block-based or region-specific), enabling more realistic inpainting. Additionally, we introduce **flexible mask ratios** ranging from 0% to 90%, making the model adaptable to a wide range of real-world occlusion scenarios.

6 Results

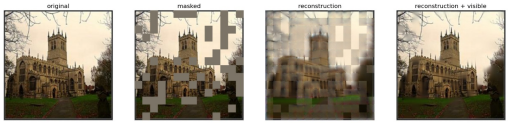
In this section, we present the experimental results for the InP-Net model across various variants and masking techniques. We compare the performance of the original model with the proposed modifications, including the use of different masking strategies, skip connections, and GAN losses.

7 Conclusions

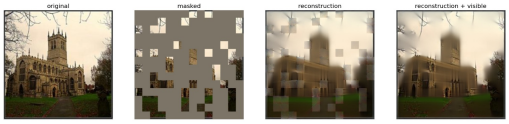
InP-Net demonstrates the effectiveness of combining **autoencoders** with **vision transformers** and **masked attention** for image inpainting tasks. Our hybrid model, leveraging both global (transformer-based) and local (convolutional) features, strikes a balance between computational efficiency and high-quality output. Key enhancements, such as **structured masking** and **skip connections**, significantly improve the model's ability to reconstruct missing regions with fine details and realism.

We observed that the introduction of **structured masking** (e.g., center, corners, and sides) provides better context for the reconstruction of various image regions, improving performance over purely random masking. Additionally, the use of **skip connections** from intermediate transformer layers to the decoder aids in preserving local features, such as texture and spatial relations, leading to more coherent reconstructions. Furthermore, the flexible **masking ratio**, adjustable from 0% to 90%, enhances the model's robustness by enabling it to handle a range of missing information scenarios during both training and inference.

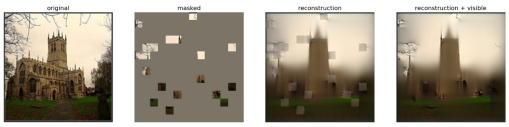
Overall, InP-Net is able to generalize well across different masking strategies and image types, demonstrating its potential for real-world applications such as inpainting for image restoration, data compression, and content-aware editing.



(a) 25% masking ratio

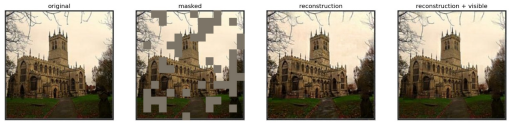


(b) 75% masking ratio

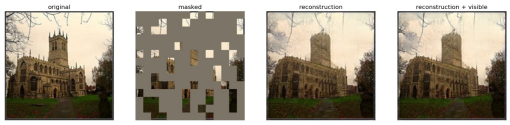


(c) 90% masking ratio

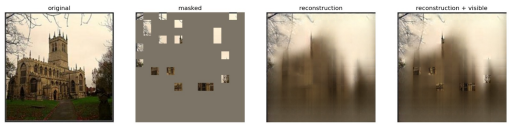
Figure 2: **Original Model reconstructions across varying mask ratios.**



(d) 25% masking ratio

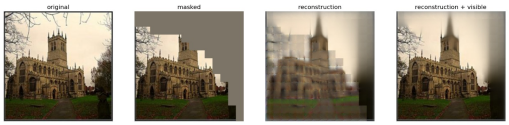


(e) 75% masking ratio

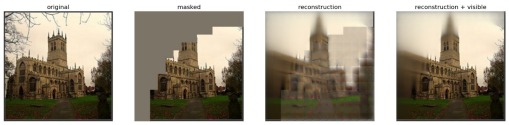


(f) 90% masking ratio

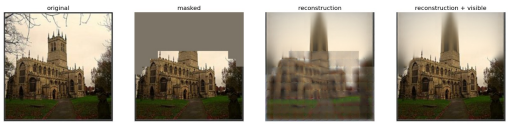
Figure 3: **Model with GAN Loss reconstructions across varying mask ratios.**



(a) bottom left masking

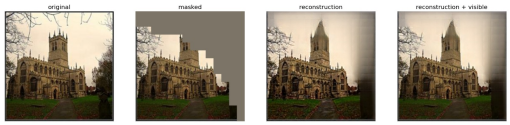


(b) bottom right masking



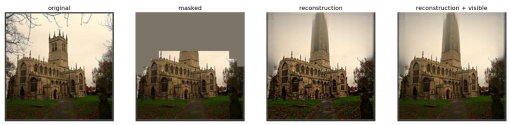
(c) bottom centre masking

Figure 4: **Model with Dynamic masks and Recon. Loss for 45% ratio.**



(d) bottom left masking

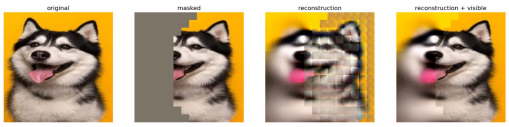
(e) bottom right masking



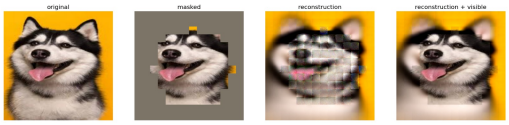
(f) bottom centre masking

Figure 5: **Model with Dynamic masks and GAN Loss for 45% ratio.**

(a) Centre left masking

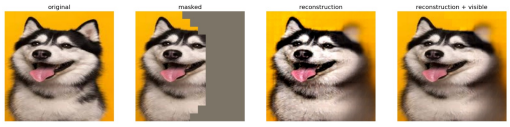


(b) Centre right masking

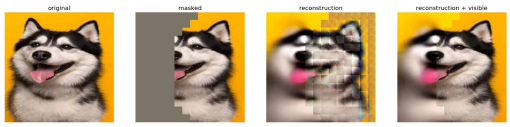


(c) centre masking

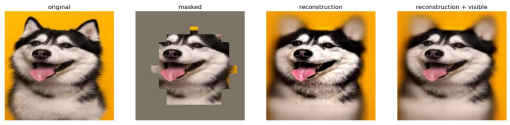
Figure 6: **Model with Dynamic masks + Recon. Loss and skip connections for 60% ratio.**



(d) centre left masking

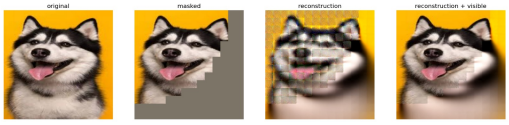


(e) centre right masking

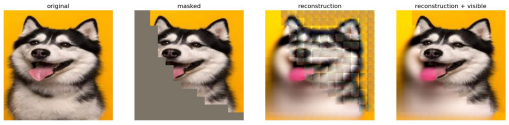


(f) centre masking

Figure 7: **Model with Dynamic masks + GAN Loss and skip connections for 60% ratio.**



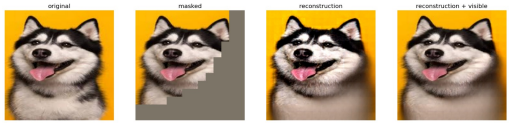
(a) Top left masking



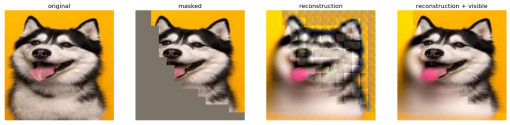
(b) Top right masking

(c) Top centre masking

Figure 8: **Model with Dynamic masks + Recon. Loss and skip connections for 60% ratio.**



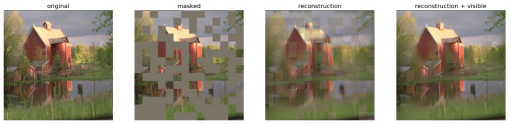
(d) Top left masking



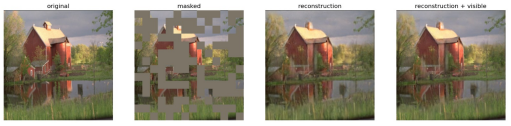
(e) Top right masking

(f) Top masking

Figure 9: **Model with Dynamic masks and GAN Loss and skip connections for 60% ratio.**

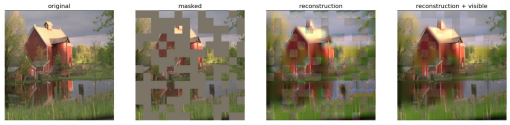


(a) Recon. Loss

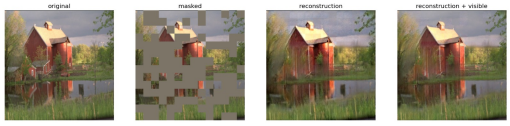


(b) GAN Loss

Figure 10: **Model with Random masks for 60% ratio of inference image**



(d) Recon. Loss



(e) GAN Loss

Figure 11: **Model with Random masks+skip connections for 60% ratio of inference image**

The self-supervised nature of the model enables it to scale efficiently without the need for extensive labeled data, making it particularly suited for large-scale, real-world deployments. The model's lightweight design, facilitated by the use of shallow convolutional decoders and efficient transformers, also makes it practical for real-time applications.

7.1 Future Work

While InP-Net has shown promising results, there are several avenues for future improvement and exploration:

- **Multi-modal Inputs**: Incorporating additional modalities, such as depth information, semantic segmentation maps, or edge maps, could help the model better understand the structure of the scene and improve inpainting quality, especially for challenging cases involving large missing regions or complex textures.
- **Transfer Learning**: Investigating the use of pre-trained models on other tasks or datasets, such as style transfer or super-resolution, may enhance the model's generalization capabilities, especially when working with highly diverse real-world images.
- **Enhanced Decoder Architectures**: While the shallow convolutional decoder offers good performance with lower computational cost, exploring deeper or more complex decoder architectures, such as U-Net or Transformer-based decoders, could further improve reconstruction fidelity, especially for high-resolution images.
- **Real-Time Applications**: Testing the model's performance on edge devices or in real-time applications such as augmented reality (AR) or video inpainting, where computational resources are limited, could open new possibilities for the deployment of InP-Net in practical scenarios.

In conclusion, InP-Net provides a solid foundation for image inpainting tasks, combining cutting-edge transformer-based architectures with practical enhancements to improve performance across various masking strategies and real-world scenarios. The versatility of the model and its ability to handle dynamic masking ratios make it a robust solution for a wide range of applications, and the future work outlined here can help further push the boundaries of what is achievable in this domain.