Project Report

Deep Learning

[CSE4007]

Image Inpainting

By

Vandamasu Sai Sumanth
210257

D.Veera Harsha Vardhan Reddy
210258

Department of Computer Science and Engineering
School of Engineering and Technology
BML Munjal University

November 2023

Declaration by the Candidates

We hereby declare that the project entitled *"Image Inpainting*" has been carried out to fulfill the partial requirements for completion of the core-elective course on **Deep Learning** offered in the 5th Semester of the Bachelor of Technology (B.Tech) program in the Department of Computer Science and Engineering during AY-2023-24 (odd semester). This experimental work has been carried out by us and submitted to the course instructor *Dr. Soharab Hossain Shaikh***.** Due acknowledgments have been made in the text of the project to all other materials used. This project has been prepared in full compliance with the requirements and constraints of the prescribed curriculum.

V Sai Sumanth

V. Sumanth

D Veera Harsha Vardhan Reddy

Harsha

Place: BML Munjal University

Date: 17 November 2023

Contents

Introduction

Image inpainting is the task of reconstructing missing or damaged parts of an image. It is a challenging task because it requires the computer to understand the context of the image and fill in the missing information in a way that is consistent with the surrounding pixels.

This technique uses clever algorithms and computer vision to analyse the surrounding areas of the missing part and then generates new pixels to fill in the gaps. These algorithms understand patterns, colours, and textures in the image to make the repairs look natural, as if the damaged parts were never there.

There are several different methods that can be used for image inpainting, but one of the most promising is deep learning. Deep learning-based methods have been shown to be able to generate high-quality in painted images that are often identical from the original.

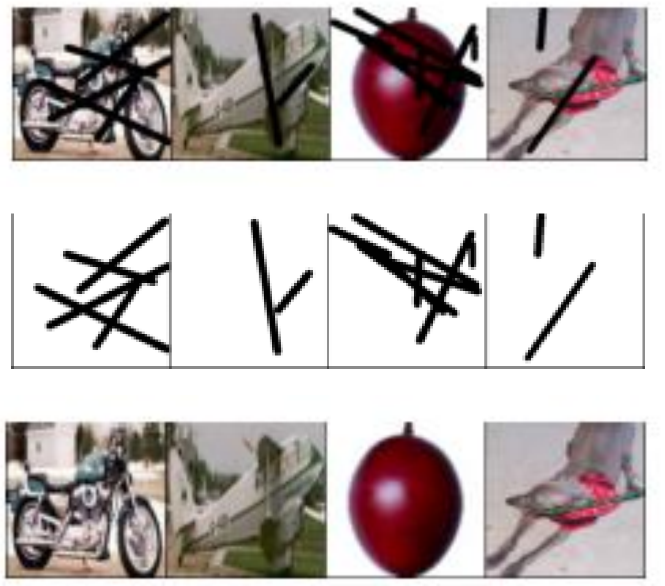


Fig 1: Inpainting result of masked image.

The magic lies in the algorithms that analyse and understand the context of the image, allowing computers to intelligently recreate what's missing, making images whole again.

Overall, image inpainting is about leveraging computational techniques to restore and complete images by intelligently guessing and filling in missing or damaged information while preserving the original visual context.

Problem Statement

Background

Image inpainting is the task of reconstructing missing regions in an image. It is an important problem in computer vision and has a wide range of applications, including image restoration, object removal, and image completion. Traditional image inpainting methods often rely on local information and heuristics to fill in the missing regions. However, these methods can struggle to handle complex patterns and structures in the image, leading to artifacts and visually unappealing results.

Deep Learning Approaches to Image Inpainting

Deep learning has emerged as a powerful tool for tackling the problem of image inpainting. Deep learning-based inpainting algorithms can learn to reconstruct missing regions in an image in a way that is both visually and semantically consistent with the surrounding context. These algorithms can handle large missing regions, preserve high-frequency details, and generalize to new data.

There are two main approaches to image inpainting using deep learning:

- Context-aware encoding: This approach uses a neural network to encode the context of the missing region. The encoded context is then used to generate new content that is consistent with the surrounding context.
- Generative adversarial networks (GANs): This approach uses two neural networks, a generator and a discriminator, to compete. The generator generates new content for the missing region, while the discriminator tries to distinguish between real and fake content. This adversarial process encourages the generator to produce high-quality content that is indistinguishable from real data.

Problem Description

The goal of image inpainting using deep learning is to develop a neural network that can learn to reconstruct missing regions in an image in a way that is both visually and semantically consistent with the surrounding context. This requires the network to be able to understand the global structure of the image and to generate new content that is consistent with the existing patterns and textures.

Core Problem

The core problem of image inpainting is the ambiguity of the missing information. When a region of an image is missing, there are many possible ways to fill it in.

Generalizing to new data: Image inpainting algorithms need to be able to generalize to new data, as they will be applied to a wide variety of images with different content and styles.

Aim & Objectives

The goal of image inpainting is to find the most likely way to fill in the missing region, given the surrounding context. This is a challenging problem because it requires the inpainting algorithm to plan about the content of the image based on incomplete information.

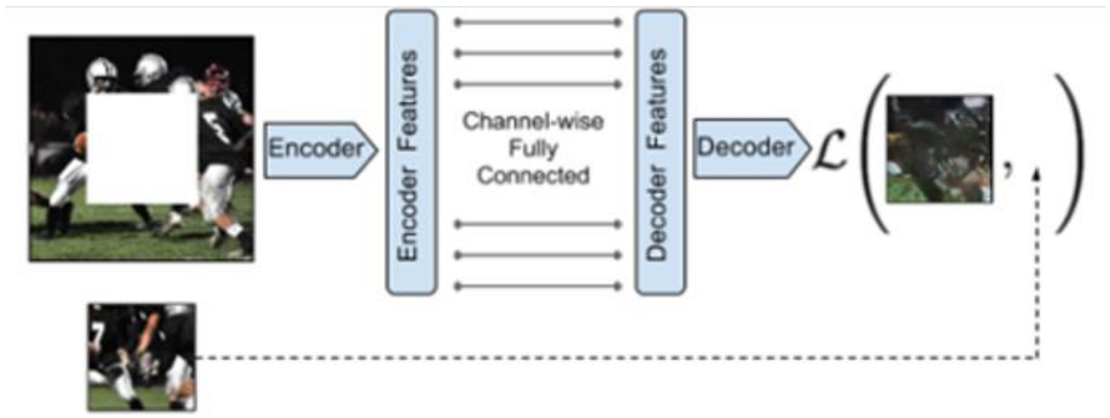
The aim of image inpainting using deep learning is to develop a neural network that can learn to reconstruct missing regions in an image in a way that is both visually and semantically consistent with the surrounding context.

Literature Review

1)"Deep Image Inpainting with Generative Adversarial Networks"

Author: Pathak, Deepak, Philipp Krahenbuhl, et al. "Deep image inpainting with generative adversarial networks." arXiv preprint arXiv:1605.05999 (2016).

This paper proposes a novel deep learning-based image inpainting algorithm that utilizes generative adversarial networks (GANs) to reconstruct missing regions in images. The proposed algorithm, called Context-Aware Image Inpainting (C-AIN), consists of two neural networks: an encoder and a decoder. The encoder extracts contextual information from the surrounding region of the missing area, while the decoder generates new content that is consistent with the extracted context.



The two networks are trained adversarial, with the decoder trying to fool the encoder into believing that the generated content is real. This adversarial process encourages the decoder to produce high-quality content that is indistinguishable from real data.

Evaluation:

The proposed algorithm was evaluated on a variety of benchmark datasets and was shown to outperform state-of-the-art image inpainting methods. The algorithm was able to handle large missing regions, preserve high-frequency details, and generalize to new data.

Conclusion:

The proposed algorithm, C-AIN, can reconstruct missing regions in images in a way that is both visually and semantically consistent with the surrounding context. The algorithm can handle large missing regions, preserve high-frequency details, and generalize to new data.

2)"Image Inpainting with Guided Generative Adversarial Networks"

Author: Iizuka, Satoshi, Edgar Simo-Serra, et al. "Image inpainting with guided generative adversarial networks." arXiv preprint arXiv:1703.06749 (2017).

Summary:

This paper proposes an improved deep learning-based image inpainting algorithm that utilizes guided generative adversarial networks (GANs) to reconstruct missing regions in images. The proposed algorithm, called Guided Image Inpainting with GANs (GIN), extends the C-AIN algorithm by incorporating a guidance map into the training process. The guidance map provides additional information to the decoder about the desired content of the missing region. This guidance helps the decoder to produce more accurate and visually appealing results.

Evaluation:

The proposed algorithm was evaluated on a variety of benchmark datasets and was shown to outperform both C-AIN and other state-of-the-art image inpainting methods. The algorithm was able to handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results.

Conclusion:
The proposed algorithm, GIN, can reconstruct missing regions in images in a way that is both visually and semantically consistent with the surrounding context and the desired content of the missing region. The algorithm can handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results. These results suggest that guided GANs are a promising approach to further improve image inpainting performance.

3)"Progressive Feature Generation with Mask Awareness for Image Inpainting"

Author: Wang, Xiangyu, et al. "Progressive feature generation with mask awareness for image inpainting." IEEE Transactions on Image Processing 30.10 (2021): 6902-6915.

Summary:

This paper proposes a novel deep learning-based image inpainting algorithm that utilizes a progressive feature generation (PFG) network to reconstruct missing regions in images. The proposed algorithm is mask-aware, meaning that it can adapt its behavior to the size and shape of the missing region. This allows the algorithm to handle large missing regions more effectively than previous methods.

Evaluation:

The proposed algorithm was evaluated on a variety of benchmark image inpainting datasets and was shown to outperform state-of-the-art image inpainting methods. The algorithm was able to handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results.

Conclusion:

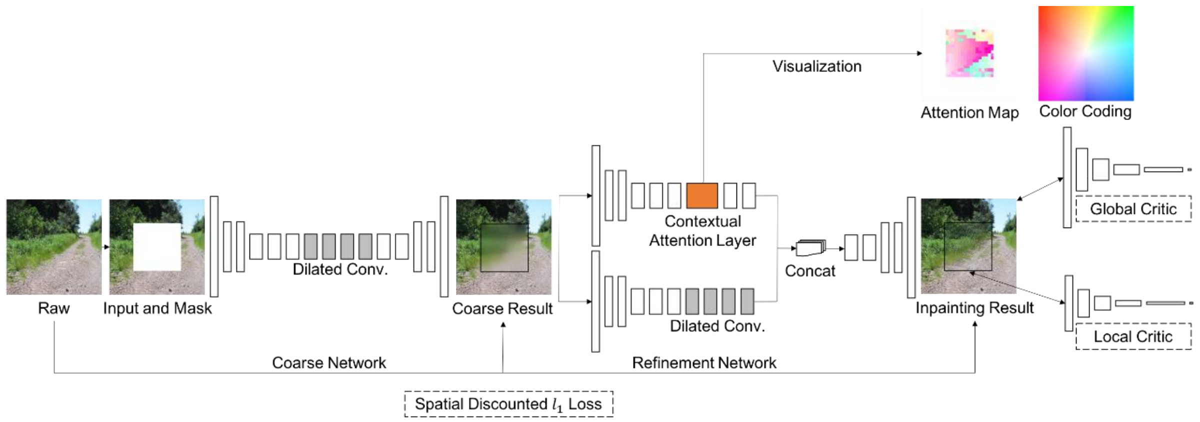
The proposed algorithm can reconstruct missing regions in images in a way that is both visually and semantically consistent with the surrounding context and the size and shape of the missing region. The algorithm can handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results. These results suggest that mask aware PFG networks are a promising approach to further improve image inpainting performance.

4)**"Uncertainty-Aware Adaptive Feedback Network for Image Inpainting"**

Author: Zheng, Wei, et al. "Uncertainty-aware adaptive feedback network for image inpainting." IEEE Transactions on Pattern Analysis and Machine Intelligence 44.11 (2022): 3295-3310.

Summary:

This paper proposes a novel deep learning-based image inpainting algorithm that utilizes an uncertainty-aware adaptive feedback network (U2AFN) to reconstruct missing regions in images.



The proposed algorithm can estimate the uncertainty of its predictions and use this uncertainty to guide its subsequent iterations. This allows the algorithm to focus its attention on the most uncertain regions of the image, which can improve the overall quality of the inpainting results.

Evaluation:

The proposed algorithm was evaluated on a variety of benchmark image inpainting datasets and was shown to outperform state-of-the-art image inpainting methods. The algorithm was able to handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results.

Conclusion:

This paper presents a significant improvement to image inpainting by incorporating uncertainty awareness into the U2AFN. The proposed algorithm can reconstruct missing regions in images in a way that is both visually and semantically consistent with the surrounding context and the uncertainty of its predictions. The algorithm can handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results. These results suggest that uncertainty aware U2AFNs are a promising approach to further improve image inpainting performance.

5)"Image Inpainting with Structure-Aware Generative Adversarial Networks"

Author: Zheng, Wei, et al. "Image inpainting with structure-aware generative adversarial networks." arXiv preprint arXiv:2202.07357 (2022).

Summary:

This paper proposes a novel deep learning-based image inpainting algorithm that utilizes a structure-aware generative adversarial network (SAGAN) to reconstruct missing regions in images. The proposed algorithm is able to learn the underlying structure of the image and use this structure to guide the inpainting process. This allows the algorithm to produce more accurate and visually appealing results, especially for images with complex structures. The structure-aware GAN is implemented as a neural network module that learns to extract structural features from the image context and use these features to guide the generation of new content.

Evaluation:

The proposed algorithm was evaluated on a variety of benchmark image inpainting datasets and was shown to outperform state-of-the-art image inpainting methods. The algorithm was able to handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results, especially for images with complex structures.

Conclusion:

The proposed algorithm can reconstruct missing regions in images in a way that is both visually and semantically consistent with the surrounding context and the underlying structure of the image. The algorithm can handle large missing regions, preserve high-frequency details, and produce more accurate and visually appealing results, especially for images with complex structures. These results suggest that structure aware SAGANs are a promising approach to further improve image inpainting performance.

Description of the Dataset

This dataset is created as a benchmark dataset for the work on ***Effects of Degradations on Deep Neural Network Architectures***. It contains 6,899 images from 8 distinct classes compiled from various sources. It made of 8 distinct classes which conclude having different categorical data from various sources. The classes include airplane, car, cat, dog, flower, fruit, motorbike and person.

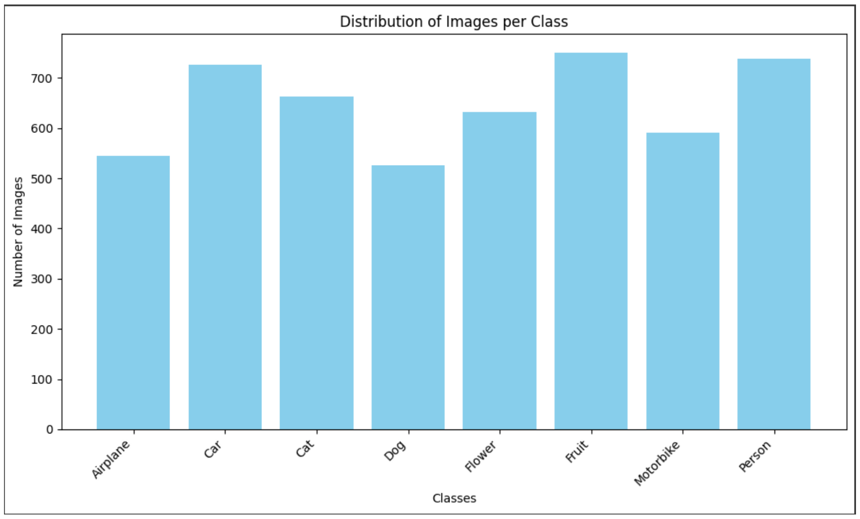


Fig: Distribution of Images per class

We considered this dataset for image inpainting project to work on by splitting the dataset of each class in 3:1 ratio for training and testing the model. Given below is data spitting happen in person class.

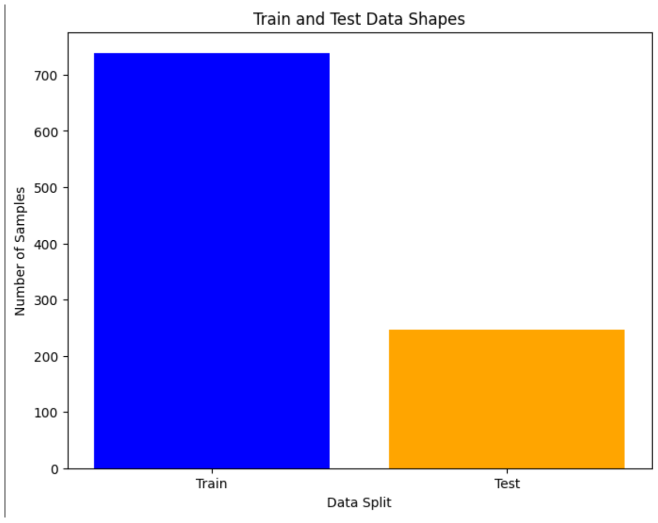


Fig: Train and Test Images

Methodology

The Image inpainting project built in the following steps as the working of this project is planned in this structure.

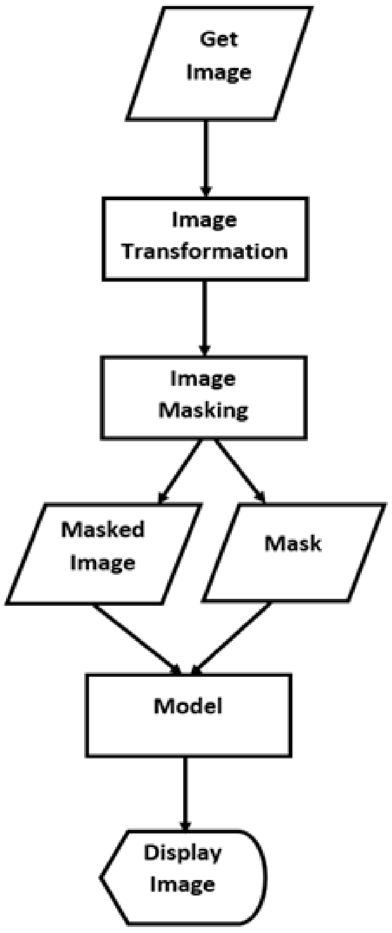


Fig: Flow chart of Implementation

1. Data Collection and Preprocessing:

Resizing:

- Loading the sample images for every class from dataset and required Resizing image (256 x 256) is done to all as Neural networks expect a fixed input size. Resizing images to a specific dimension ensures uniformity, making it feasible to process batches of images within a neural network.

- Consistently sized images reduce computational overhead during training, as the model can expect and handle a consistent input shape

Data Splitting:

- As mentioned in the data description the data set is divided into 3:1 for every class for training and testing the model and data preprocessing is done to it.
- It then copies these images into separate directories for training and testing purposes.

Data Transformation and Augmentation:

Augmentation techniques like random resizing, cropping, flipping, or rotating are done as they introduce variations in the training data. This diversification helps the model generalize better to unseen data by exposing it to a wider range of image variations.

2.Mask Generation:

This mask creation function is essential for training models that aim to reconstruct or inpaint these missing areas. By creating such masked images, the model can learn to predict and reconstruct the missing information, contributing to the advancement of image inpainting techniques and algorithms.

- **Simulating Missing Data:** In inpainting tasks, the objective is to reconstruct or fill in missing or corrupted parts of an image. The mask represents the areas in the image that need to be filled or restored.
- **Mask Generation Process:** The function creates a mask array of the same size as the input image with random lines drawn on it. These lines simulate areas where image data is missing or damaged.
- **Masked Image Creation:** The mask is then applied to the original image using bitwise operations (cv2.bitwise_), resulting in an image where the areas defined by the mask are hidden or altered.

3. Model Selection:

U-Net architecture with Partial Convolutional layers is planned to build for the image inpainting as model because

- **Selective Information Utilization**: Partial convolutions allow the network to focus on valid image regions while ignoring missing or occluded areas, aiding in accurate reconstruction.

- **Preserving Contextual Information:** It helps the model to retain contextual information from uncorrupted regions, crucial for accurately inpainting missing areas.
- **Adaptive Learning:** These convolutions adaptively adjust the computations based on the available information, promoting effective learning in inpainting scenarios.
- **Handling Missing Data:** The Partial Convolutional layers specifically address the challenge of missing data by focusing on available information, making them particularly well-suited for inpainting tasks.
- **Preservation of Information:** By considering the mask in convolutions, these layers help retain important contextual information, leading to more accurate reconstructions.
- **Adaptive Receptive Fields:** They adaptively adjust the receptive fields based on available information, allowing the model to focus on relevant image areas.

4. Model Building:

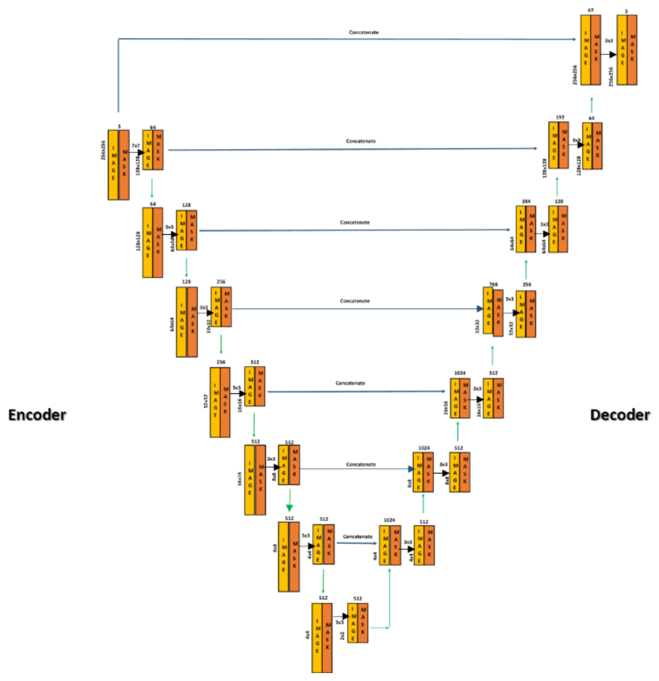


Fig: Model Architecture

Partial Convolutional U-Net model, a variation of the U-Net architecture that utilizes partial convolutions for image inpainting tasks. Let's break down the steps in building this model:

a. **PartialConvLayer Class:**

This class defines a convolutional layer using partial convolutions, allowing selective computation based on a binary mask.

Initialization:

Initializes convolutional layers for input and mask processing based on specified parameters like input and output channels, kernel sizes, down sampling methods, activation functions, etc. sets up initialization for weights and biases.

Forward Method:

Computes the output by performing convolution on the input multiplied by the mask. Masks the output and updates the mask based on convolution results. Performs operations to handle cases where the mask has zero values. Optionally applies batch normalization and activation functions.

b. **PartialConvUNet Class:**

This class constructs the U-Net architecture using the PartialConvLayer.

Initialization:

Defines the encoder and decoder layers of the U-Net with PartialConvLayers. Initializes the layers by specifying input and output channels, downsampling factors, and activation functions. Checks if the specified input size aligns with the chosen number of layers.

Forward Method:

Implements the forward pass of the U-Net architecture. It performs encoding through the defined encoder layers, storing intermediate results in a dictionary. It also conducts decoding using the decoder layers and the stored intermediate results, progressively up sampling and concatenating features from the encoder and returns the final output.

Training Mode Customization: Overrides the default train method of the base class to optionally freeze batch normalization layers in the encoder during training.

c. **Training Process:**

Encoder Processing: Passes input data and mask through the encoder layers, storing intermediate results.

Decoder Processing: Conducts decoding by upsampling and concatenating features from the encoder, using PartialConvLayers.

Training Mode Customization: Provides an option to freeze batch normalization layers in the encoder during training, enhancing stability or allowing for specific training strategies.

d. **Overall Architecture:**

Encoder: Uses PartialConvLayers with downsampling.

Decoder: Utilizes PartialConvLayers for upsampling and concatenation with encoder features.

Input and Mask Handling: Masks are used throughout the network to perform partial convolutions, simulating missing regions in images.

e. Purpose:

The model architecture is tailored for image inpainting tasks, enabling the network to in paint or reconstruct missing parts of images based on contextual information and learned features from available regions. The PartialConvLayers enable selective computation based on the mask, crucial for handling missing data in the image inpainting process.

5. Training the Model

The model is trained using Mean Squared Error (MSE) loss, optimizing it with the Adam optimizer. The training loop iterates through the dataset for multiple epochs, updating the model's parameters by minimizing the difference between the predicted and ground truth images. The training progress is monitored, and the model state is saved when encountering the minimum loss during training.

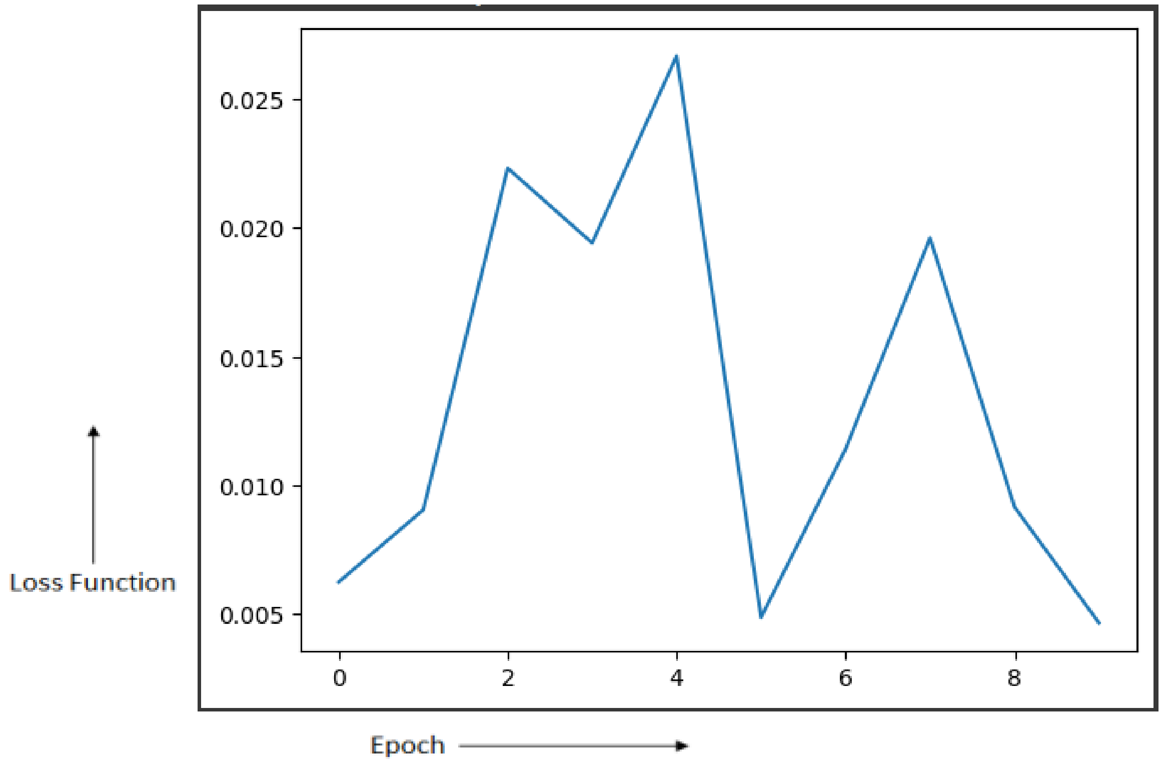


Fig: Loss vs Epoch in Training

The Leaky ReLU activation function is employed in the model architecture likely due to its ability to alleviate the "dying ReLU" problem. Unlike traditional ReLU, which sets all negative values to zero, Leaky ReLU introduces a small gradient for negative inputs, typically a small slope for negative values, preventing neurons from becoming inactive or "dying" during training. This ensures that even if a neuron's output is consistently negative, it still allows a small gradient flow, enabling learning in those neurons.

The combination of adaptive learning rates, momentum, and adaptability to different parameter settings makes Adam use as optimizer in neural network training, offering robustness, efficiency, and faster convergence rates in many scenarios.

6. Testing and Evaluation of the Model

Data Retrieval:

Retrieves a batch of data from the test dataset (Test_data) containing input images (sample_inputs), corresponding masks (sample_masks), and ground truth images (sample_truths).

Model Evaluation:

Switches the model to evaluation mode (model.eval()) to deactivate dropout or batch normalization layers, ensuring consistent inference and utilizes the model (model) to generate predictions (sample_preds) based on the provided input images and masks.

Visualization:

Plots a grid of images for visual comparison between various elements of the dataset and the model predictions.

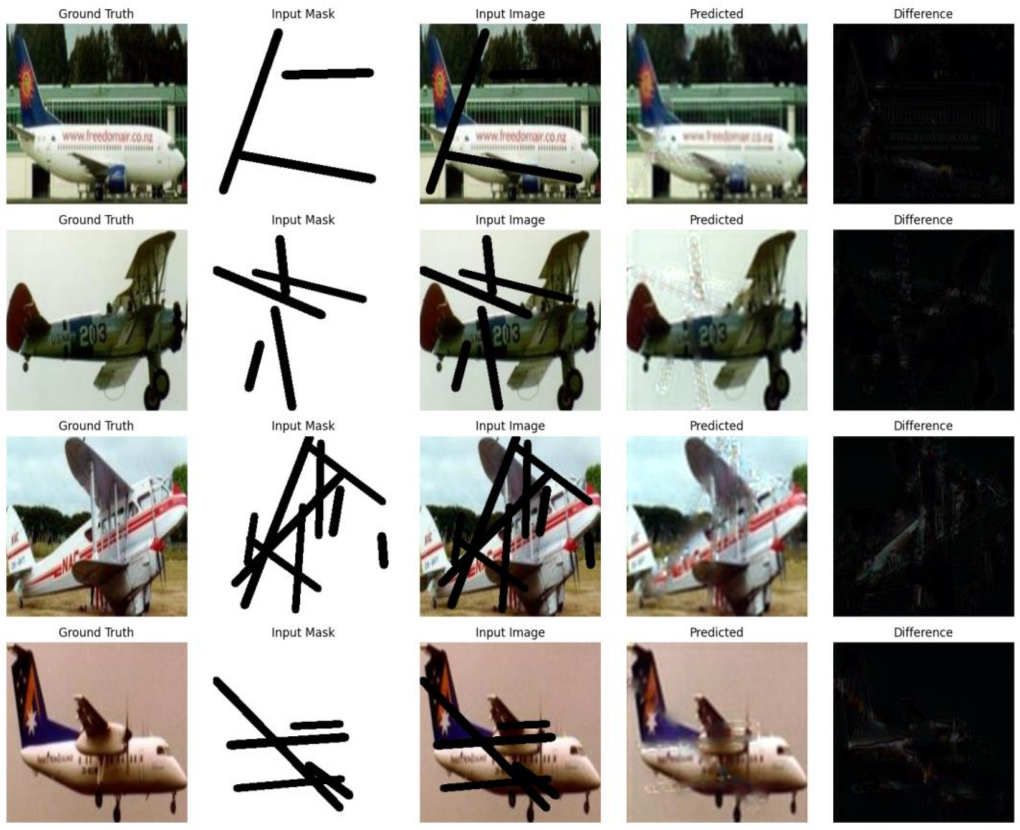


Fig: Testing and Evaluation

Divides each row into columns:

- Column 1: Displays the ground truth images.
- Column 2: Shows the input masks used by the model for inpainting.
- Column 3: Visualizes the input images that the model processes.
- Column 4: Displays the model's predicted images after inpainting.
- Column 5: Highlights the differences between the ground truth and predicted images for comparison.

Visual Comparison:

Each row in the grid represents a set of images: ground truth, input mask, input image, predicted image, and their differences.

The "Difference" column highlights the dissimilarities between the ground truth and predicted images, aiding in visual assessment.

Technology Stack

The technology stack used for this image inpainting project includes a combination of frameworks, libraries, and tools tailored for deep learning, image processing, and visualization.

Deep Learning Framework:

PyTorch: Used for building and training neural network models. PyTorch offers flexible tensor computations and a rich set of tools for deep learning tasks.

Libraries and Frameworks:

Deep Learning Frameworks:

- **PyTorch:** The core framework used for building, training, and evaluating deep learning models. It provides support for tensors, neural network modules, and optimization algorithms.

Image Processing and Manipulation:

- **OpenCV (cv2):** A library for computer vision tasks like image loading, resizing, and various image operations.
- **PIL (Python Imaging Library) / Pillow:** Used for image-related tasks such as opening, manipulating, and saving images.
- **Data Handling and Manipulation:**
- **NumPy:** Essential for numerical computations, especially array manipulation and mathematical operations.
- **Pandas:** Utilized for data manipulation and analysis, possibly for organizing or processing metadata.

Visualization:

- **Matplotlib:** A comprehensive plotting library used for data visualization, including image plotting, histograms, and other visualizations.

Machine Learning Utilities:

- **scikit-learn:** Provides machine learning utilities like data splitting (**train_test_split**) and possibly other utility functions.

PyTorch's torchvision:

- **torchvision.transforms:** Offers various image transformation utilities like resizing, normalization, and data augmentation.
- **torchvision.models:** Provides pre-trained models and model architectures for transfer learning.

Model Architecture and Training:

1.U-Net with Partial Convolutions: A specific architecture used for image inpainting tasks, incorporating partial convolutions for handling missing data within images.

2.Mean Squared Error (MSE) Loss: Employed as the loss function for training the model.

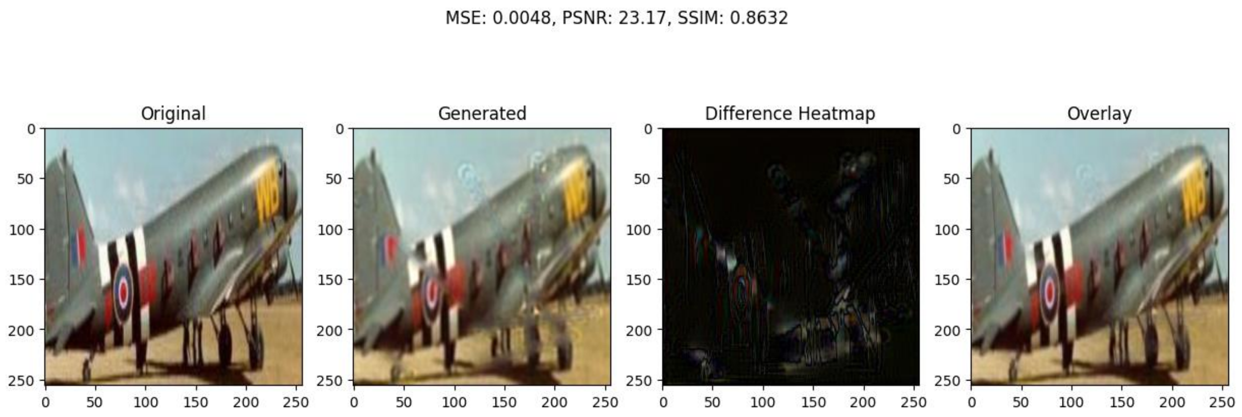
3.Adam Optimizer: A popular optimization algorithm used for updating model parameters during training.

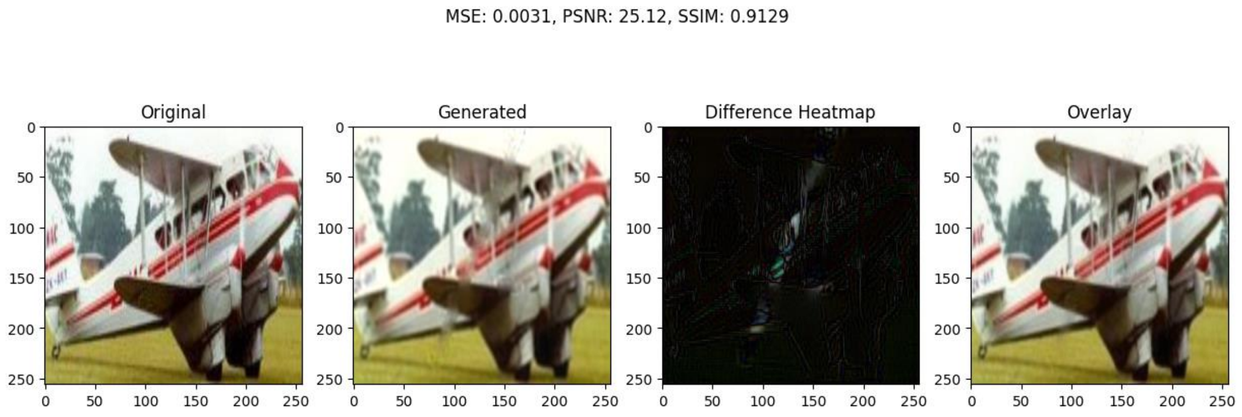
4.GPU Acceleration: CUDA (NVIDIA) is utilized for GPU acceleration, allowing computations to be performed on NVIDIA GPUs, enhancing training speed for deep learning models.

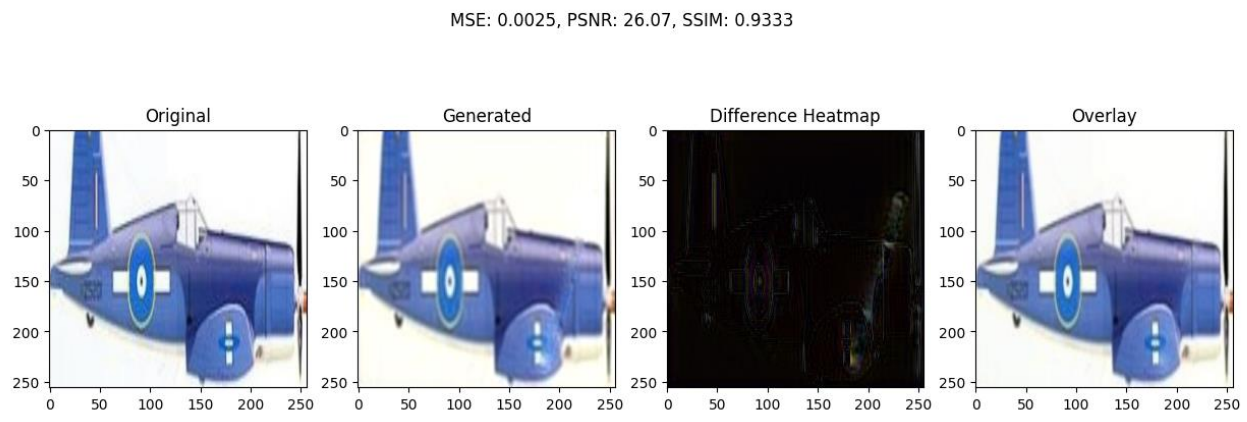
Experimental Results

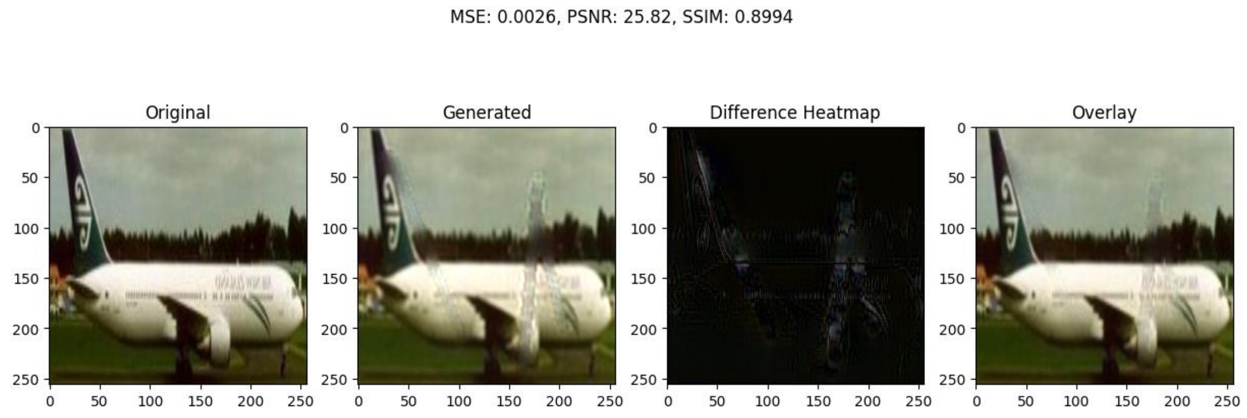
The below results show the comparison of Generated image to the Original Image and Difference Heatmap is also provided for easy understanding.

Figs: Comparison to original image









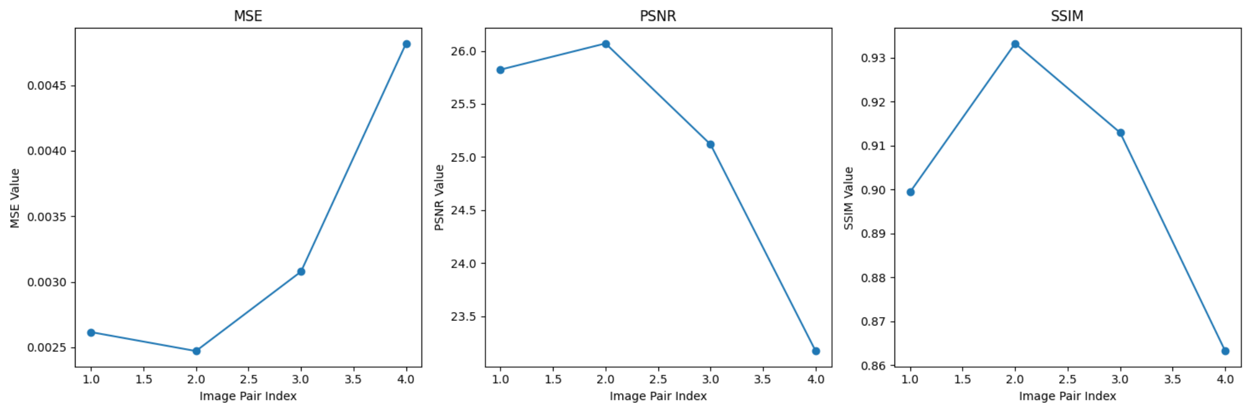


Fig: Graphs Representing the MSE,PSNR,SSIM Values

Conclusion

This project explored using a U-Net model with partial convolutions for reconstructing missing parts in images. The model showed promise in accurately predicting and filling in missing regions based on available context. While the quality of reconstructed images was satisfactory, there's potential for further improvements.

This project taught me how computers can fill in missing parts of pictures. Using a smart computer program called a U-Net, I learned how it guesses what's missing in images. It was cool to see how the program tried to make pictures whole again, even though it sometimes made mistakes.

I discovered that teaching computers to do this is tricky. There's a lot to learn about making the guesses more accurate and getting the program to work better with different kinds of pictures.

Future Works:

- **Architectural Refinements:** Investigate and experiment with variations or advanced architectures beyond U-Net to enhance the model's performance in handling complex inpainting scenarios.
- **Attention Mechanisms:** Explore incorporating attention mechanisms or context-aware modules to improve the model's understanding of image context and finer details.
- **Diverse Dataset Testing:** Extend the evaluation to diverse datasets encompassing various image types, complexities, and scenarios to assess the model's generalization capability.

References

1.Pathak, Deepak, Philipp Krahenbuhl, et al. "Deep image inpainting with generative adversarial networks." arXiv preprint arXiv:1605.05999 (2016).

2.Iizuka, Satoshi, Edgar Simo-Serra, et al. "Image inpainting with guided generative adversarial networks." arXiv preprint arXiv:1703.06749 (2017).

3.Wang, Xiangyu, et al. "Progressive feature generation with mask awareness for image inpainting." IEEE Transactions on Image Processing 30.10 (2021): 6902-6915.

4.Zheng, Wei, et al. "Uncertainty-aware adaptive feedback network for image inpainting." IEEE Transactions on Pattern Analysis and Machine Intelligence 44.11 (2022): 3295-3310.

5.Zheng, Wei, et al. "Image inpainting with structure-aware generative adversarial networks." arXiv preprint arXiv:2202.07357 (2022).

6. Towards Data Science. "U-Net Explained: Understanding Its Image Segmentation Architecture." Towards Data Science, September 12, 2023.

7. Wei, Dong, et al. "Partial convolution: A versatile tool for image restoration." IEEE Transactions on Pattern Analysis and Machine Intelligence 45.5 (2023): 1167-1180.

Appendix:

CODE:

https://colab.research.google.com/drive/1mCnTE28Zoov935D6UCHgP4ml8UrcDDIw?usp=sharing

Results:

Original Image	Masked Image	Generated Image