Facial Inpainting with Generator and VGG Feature Extraction

...

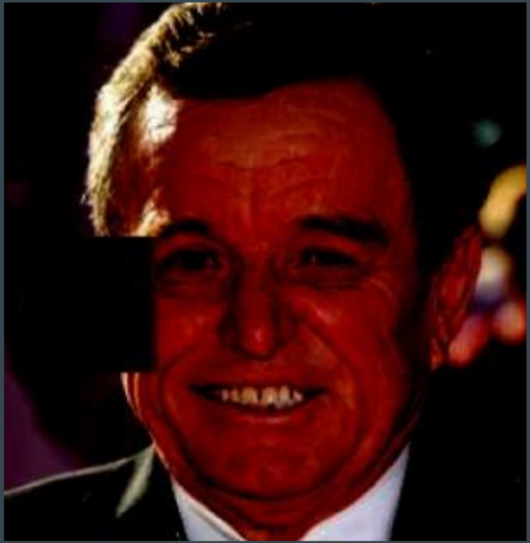
Presented by:

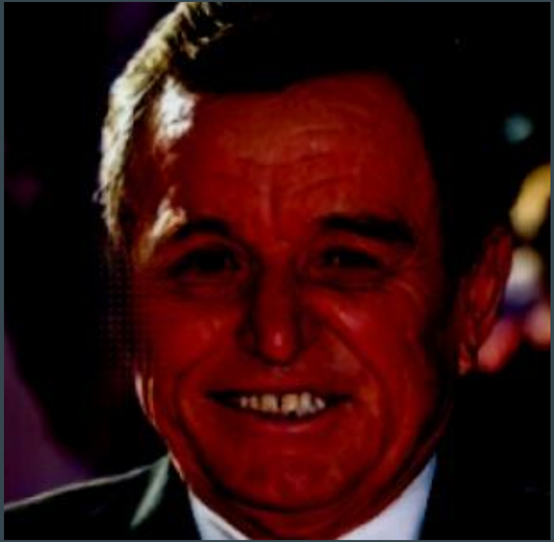
Salma Yousry	7353
Rahma Abdulhamid	7358

28/12/2024

Introduction to Facial Inpainting

Objective: Restore missing parts of facial images using a deep learning model and data mining techniques.





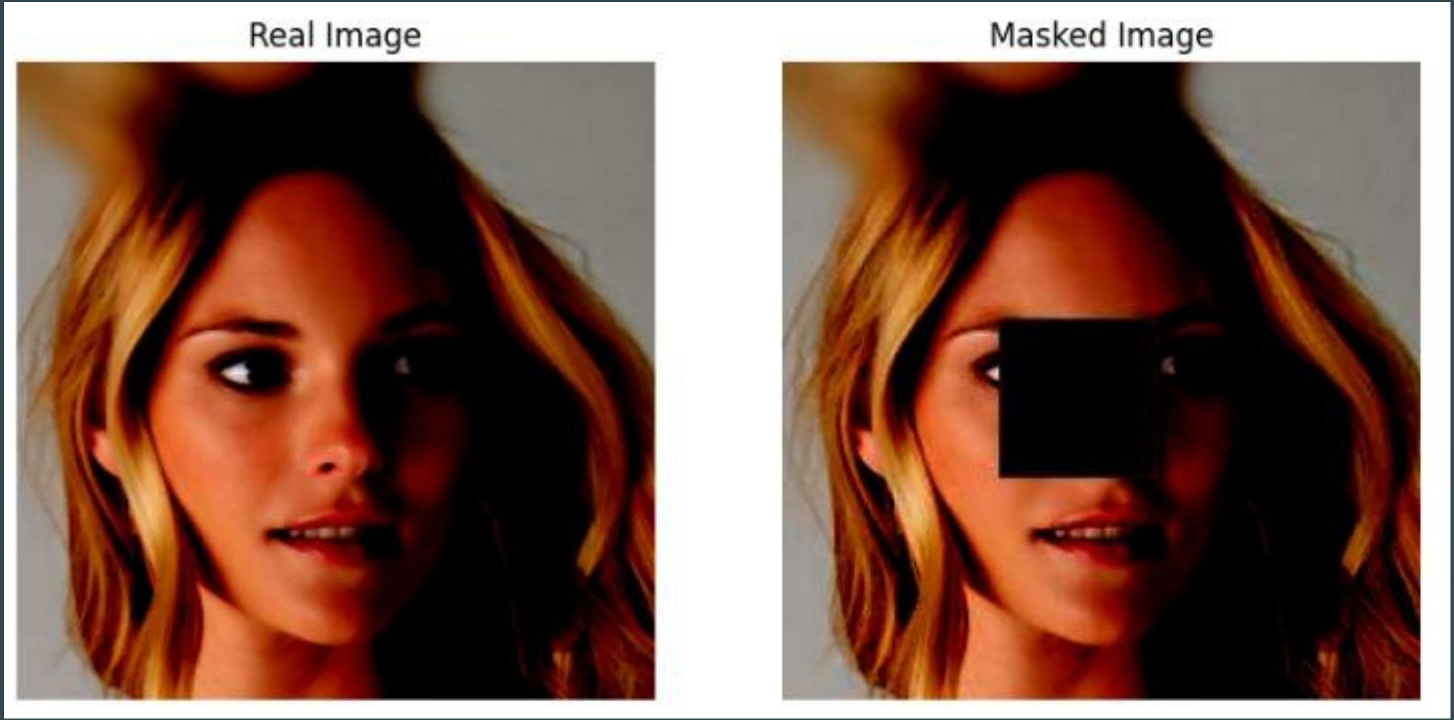
Dataset: CelebA-HQ dataset used for both training and validation (High-resolution celebrity faces).

With 28000 training images , divided into male and female categories.

And 2000 images used for the testset, with the same division as the training set.

Approach: Utilize a Generative Network architecture with feature extraction based on VGG19 for perceptual loss.

Comparison before and after Preprocessing and adding Masks



Overview of the Model Architecture

Generator: A deep convolutional network that generates the inpainted image.

VGG Feature Extractor: Used to compute perceptual loss based on pre-trained features.

Generator Architecture

The Generator model is a type of neural network .It creates or inpaints missing parts of an image using an encoder-decoder architecture, with residual blocks to refine its features. In this case, it is used for facial inpainting, meaning that it generates or fills in missing parts of a facial image

1-Encoder:

Uses multiple convolution blocks to progressively reduce the spatial dimensions while increasing the feature depth.(DownSampling)

Generator Architecture

It uses three **convolutional blocks**, each performing:

- Convolution (`Conv2d`) to extract features.
- Batch normalization (`BatchNorm2d`) to stabilize training by normalizing the activations.
- **ReLU activation** (by default) to introduce non-linearity, which helps the model learn complex patterns.

Generator Architecture

2- Residual Blocks (Feature Refinement):

- The **residual blocks** are a set of layers that further refine the features learned by the encoder. This helps to avoid vanishing gradients.

3- Decoder (UpSampling):

The decoder part of the model gradually reconstructs the image from the lower-resolution feature maps produced by the encoder and residual blocks.

Generator Architecture

3- Decoder (UpSampling):

It uses **transposed convolutions** (`ConvTranspose2d`), which are the opposite of standard convolutions and serve to upsample the feature maps, gradually increasing the image size.

The final layer uses a **Tanh activation** to produce the output, which typically ranges from -1 to 1.

VGG Feature Extractor

Extract features from different layers of a pre-trained VGG19 network.

Layer Selection:

Specific layers of VGG19 (like layers 1, 6, 11, 20, and 29) are selected for feature extraction to capture high-level semantic features of the image.

Normalization: The input is normalized using the mean and standard deviation of the ImageNet dataset, which helps the model generalize better.

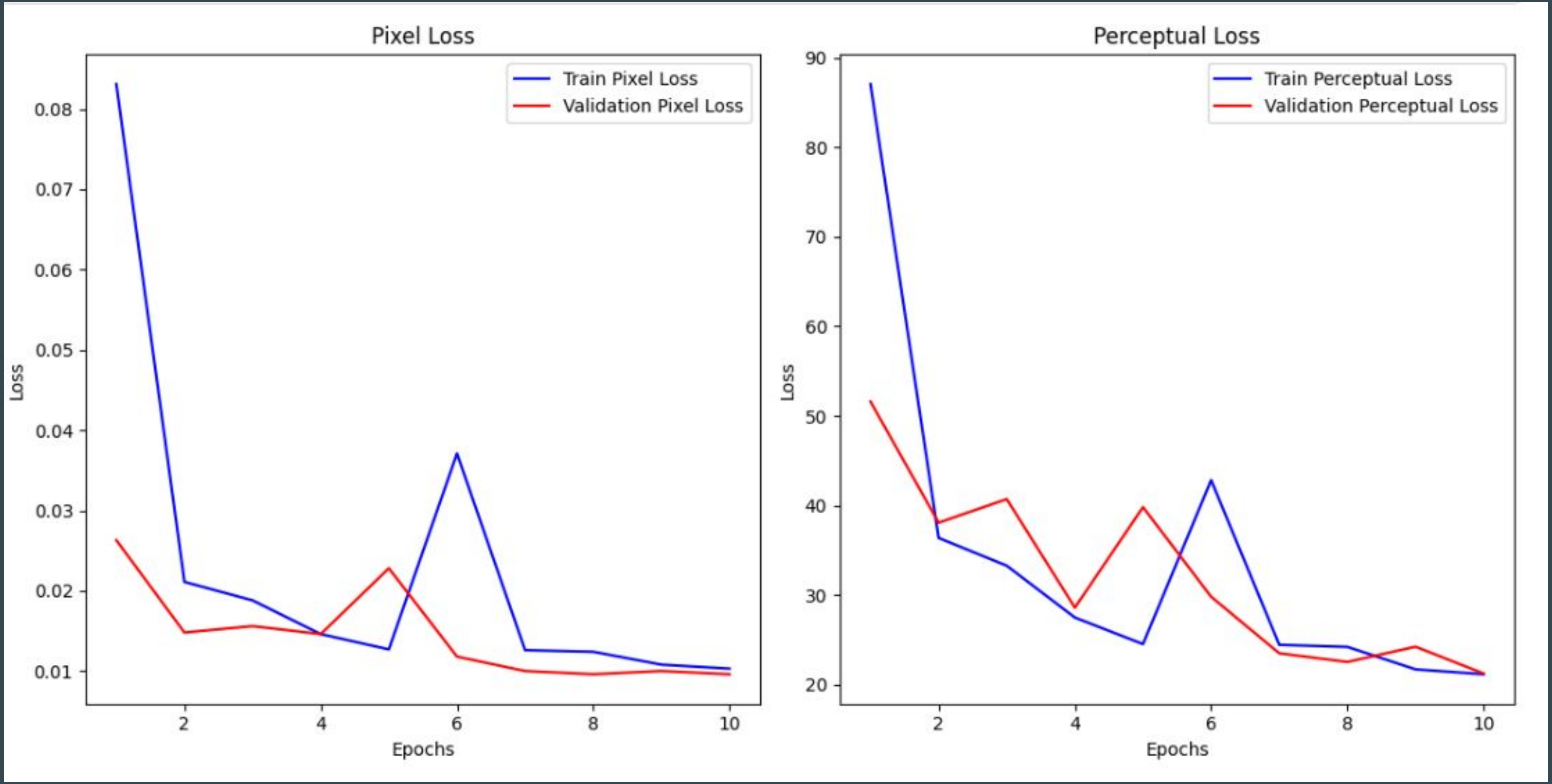
Training the Model

Loss Functions:

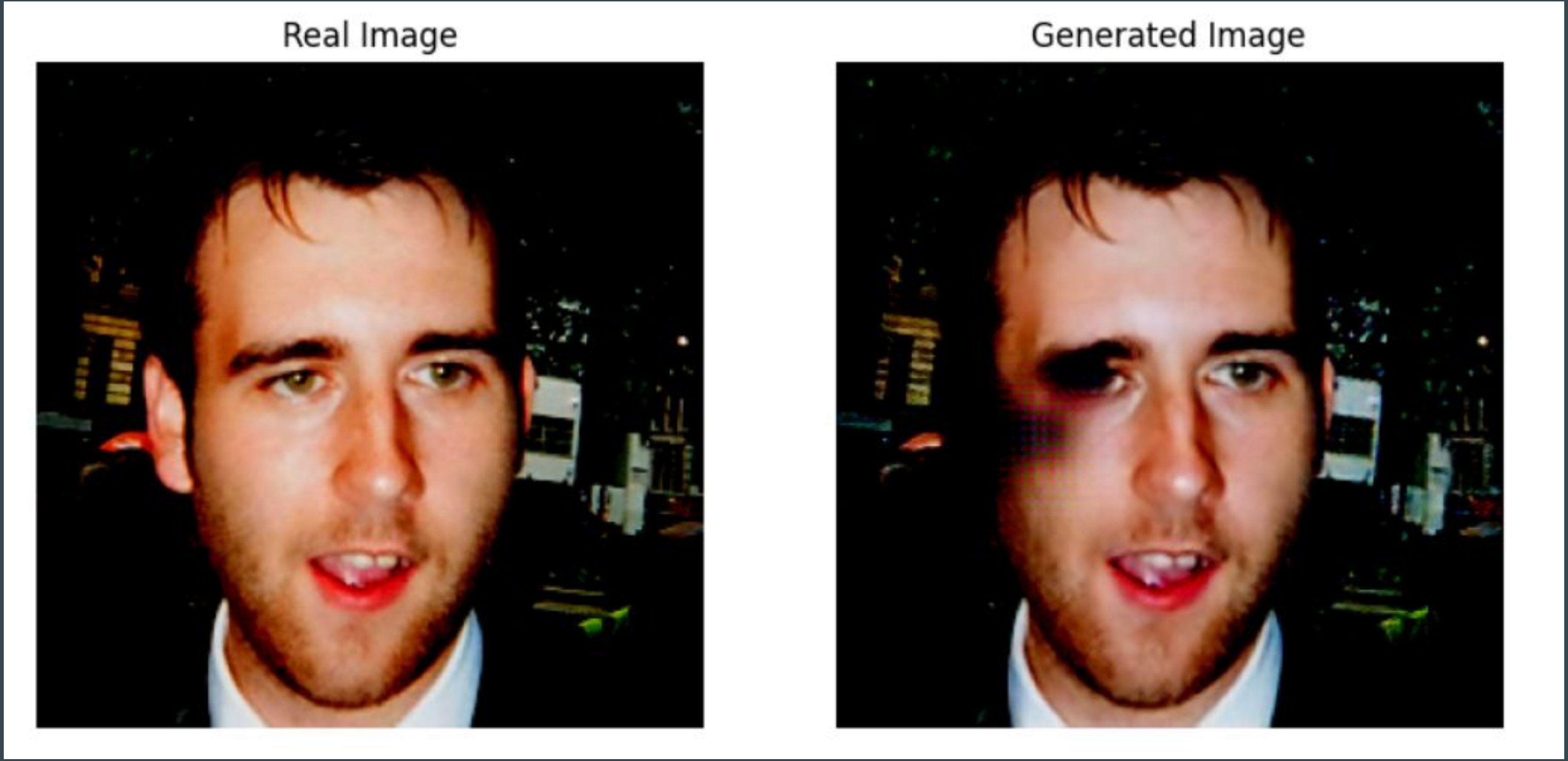
Perceptual Loss: Uses the VGG19 features to calculate the difference between the generated image and the ground truth in feature space.

Pixel-wise Loss: Measures the pixel-level similarity between the generated image and the original image.

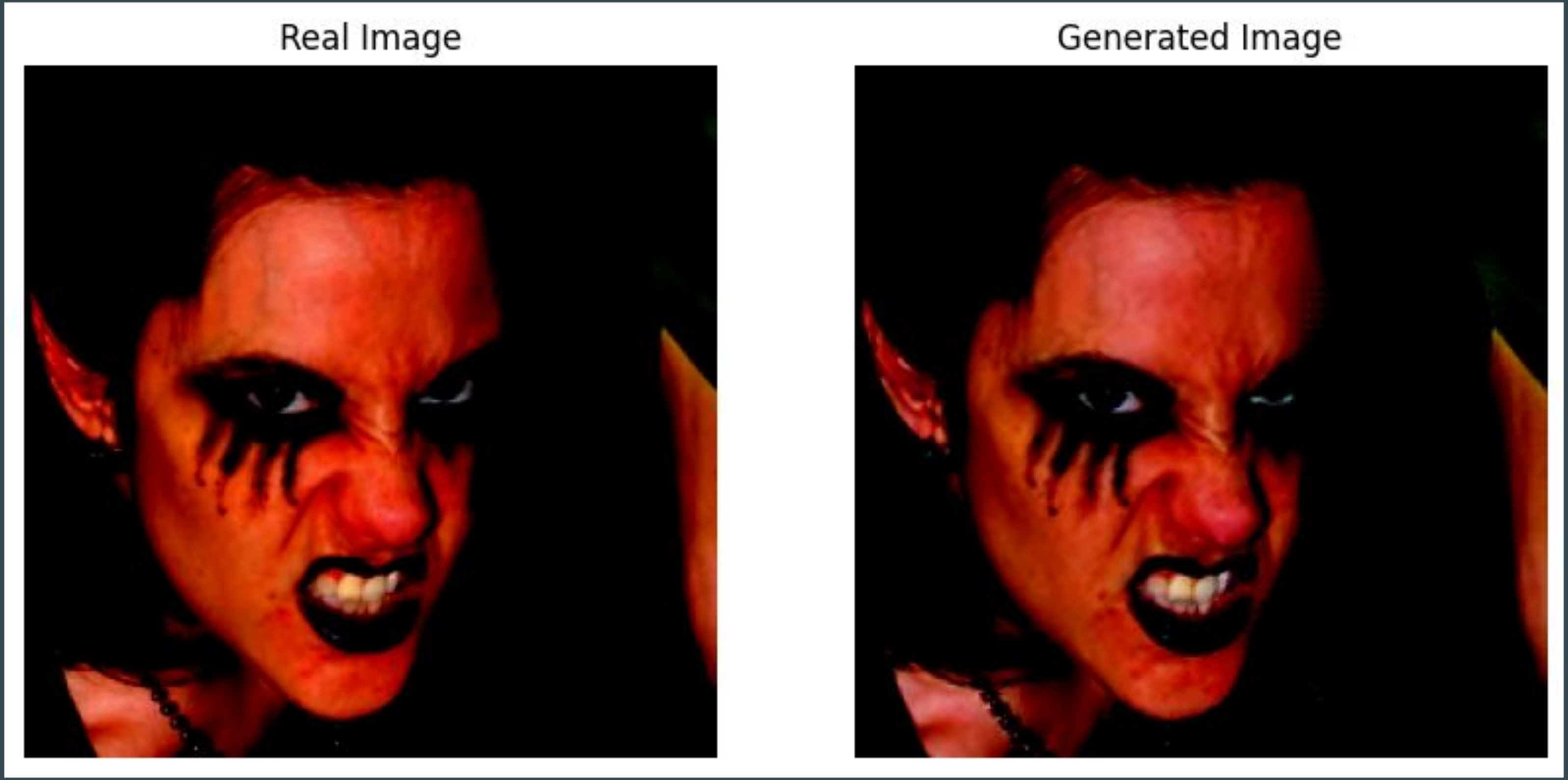
Losses Plots



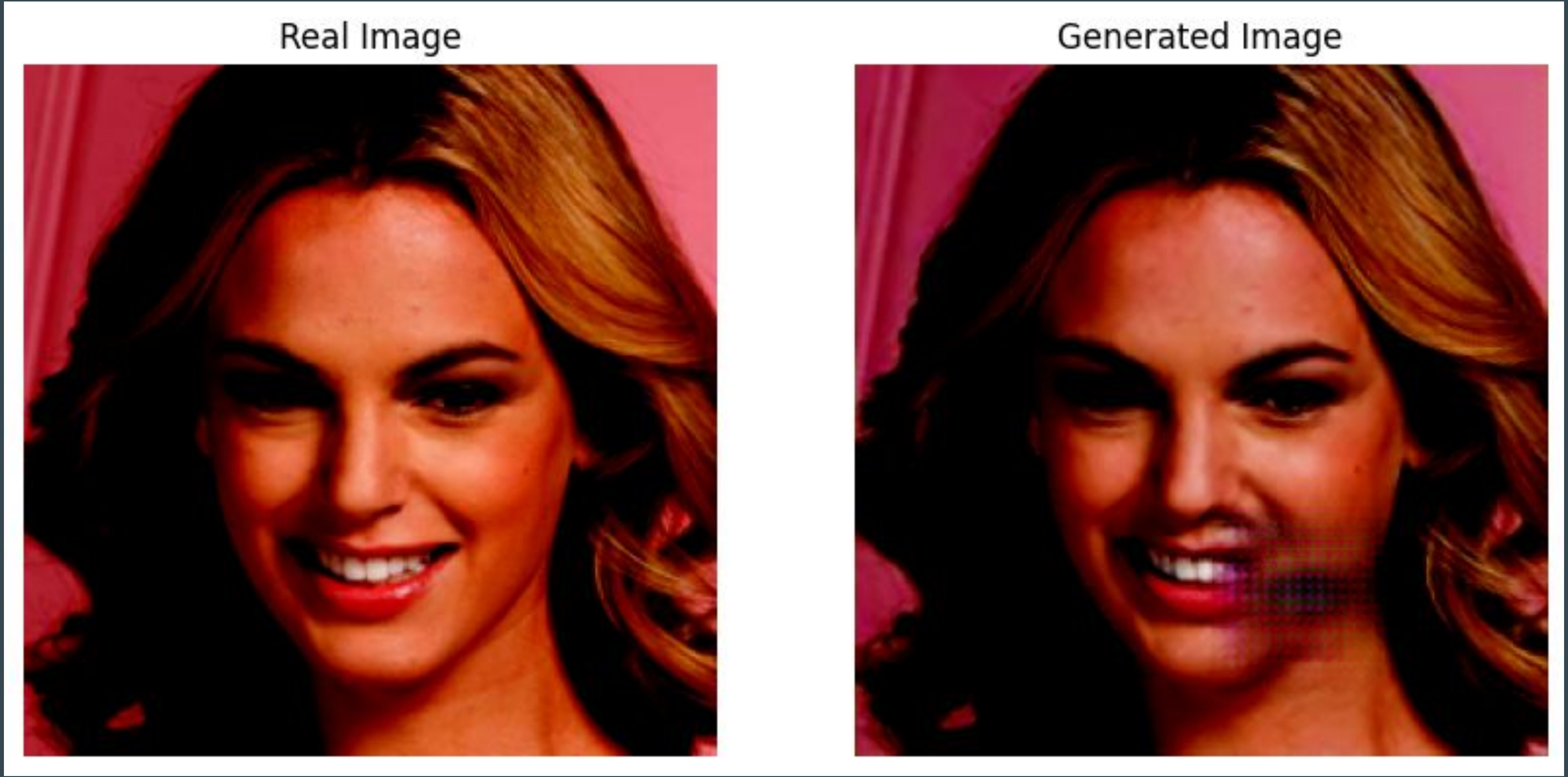
Visual Examples



Visual Examples

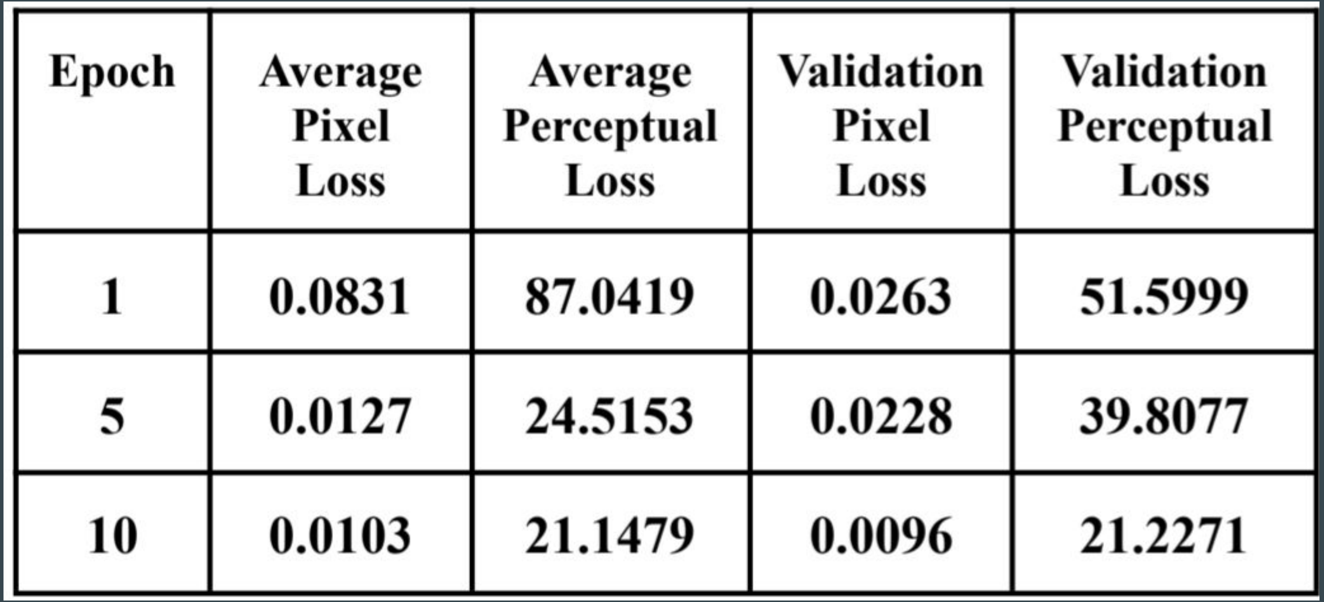


Visual Examples



Performance Metrics and Results

It was concluded that the model results indicated consistent improvement in both pixel and perceptual loss



Epoch	Average Pixel Loss	Average Perceptual Loss	Validation Pixel Loss	Validation Perceptual Loss
1	0.0831	87.0419	0.0263	51.5999
5	0.0127	24.5153	0.0228	39.8077
10	0.0103	21.1479	0.0096	21.2271

Comparisons of different papers

Paper 1: Generative Adversarial Networks

Training alternates between optimizing the discriminator and the generator. Theoretical results show the generator converges to the true data distribution given sufficient capacity and training time.

Comparisons of different papers

Paper 2: Encoders for Image Inpainting

Introduced encoder-decoder pipelines with context encoders for predicting missing regions in images. Combined reconstruction loss (L2) and adversarial loss to balance structural coherence and realism.

Comparisons of different papers

Paper 3: Gated Convolutional Neural Networks GCNN

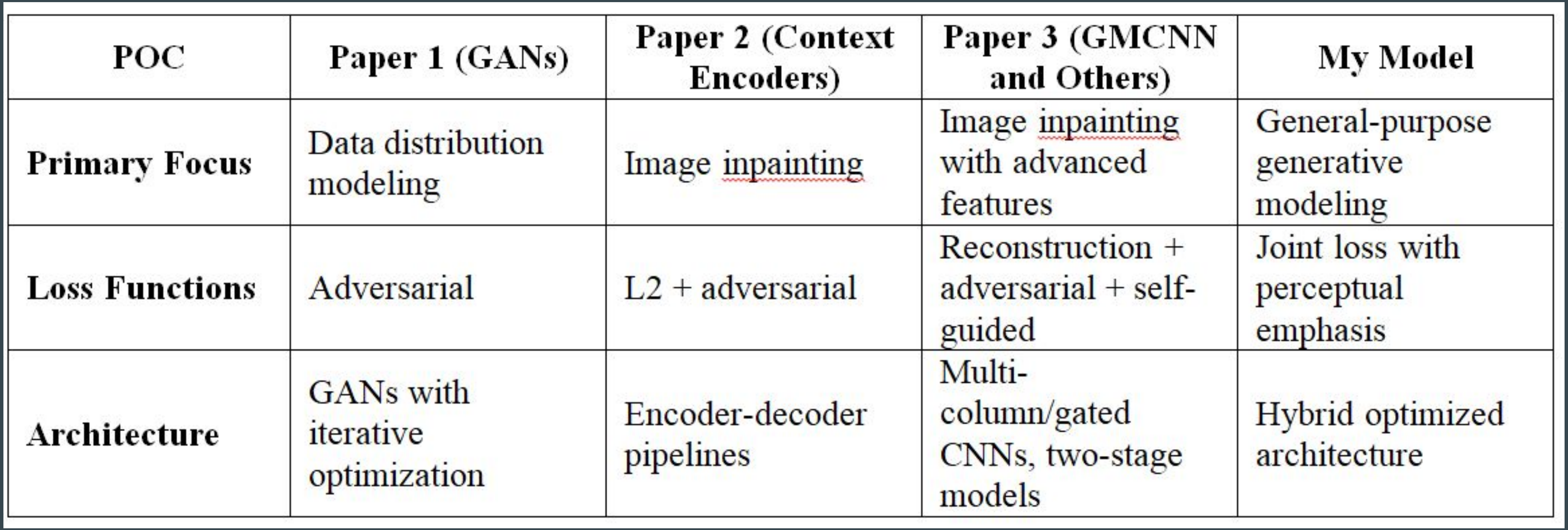
Introduced strategies for improving computational efficiency (e.g., parallel paths for semantic inpainting) and enhancing image texture quality with edge generation or contextual attention mechanisms.

Comparisons of of proposed model with different papers

Our Model

- **Paper 1:** Our model extends it by optimizing the adversarial loss more effectively
- **Paper 2:** Inspired the integration of perceptual loss for better texture and semantic detail. But our model focuses more on image inpainting only
- **Paper 3:** Our model depended on the idea of combining efficient architectures with advanced loss functions to preserve semantic consistency

Comparisons of of proposed model with different papers



POC	Paper 1 (GANs)	Paper 2 (Context Encoders)	Paper 3 (GMCNN and Others)	My Model
Primary Focus	Data distribution modeling	Image inpainting	Image inpainting with advanced features	General-purpose generative modeling
Loss Functions	Adversarial	L2 + adversarial	Reconstruction + adversarial + self-guided	Joint loss with perceptual emphasis
Architecture	GANs with iterative optimization	Encoder-decoder pipelines	Multi-column/gated CNNs, two-stage models	Hybrid optimized architecture

Alternative Approaches

Generative Adversarial Networks (GANs) involve a generator and a discriminator. The generator tries to produce realistic images, while the discriminator tries to distinguish between real and generated images.

This approach encourages the generator to produce images that are indistinguishable from real images according to the discriminator, improving the realism of the generated image.

Conclusion

Our generator-based architecture demonstrates its potential in facial inpainting by producing visually convincing results.

However, addressing limitations like irregular masks and dataset dependency will pave the way for more robust solutions

References

1. Goodfellow, I., et al. "Generative Adversarial Networks." 2014.

2. Pathak, D., et al. "Context Encoders: Feature Learning by Inpainting." CVPR, 2016.

3. Hui, Zheng, et al. "Image fine-grained inpainting." arXiv preprint arXiv:2002.02609 (2020).

Thank You