COGS 181 Final Project

Yue Yin
University of California, San Diego
yuyin@ucsd.edu

Keyi Chen
University of California, San Diego
kec020@ucsd.edu

Abstract

The project focuses on image reconstruction, specifically targeting the restoration of images with a centrally located square mask that obscures part of the image. We propose the development of a context encoder, utilizing L2 loss as the primary mechanism, to effectively reconstruct the missing central portion of the image. This initiative is particularly aimed at assisting individuals in repairing damaged or incomplete images, thereby preserving valuable visual information.

1 Introduction

The project aims to bridge the gap between deep learning theories and their practical applications through the development of a context-based pixel prediction algorithm for image restoration. Central to our approach are context encoders, a specialized variant of convolutional neural networks, tasked with reconstructing missing image segments by leveraging contextual information from the surrounding pixels. These encoders are designed to assimilate and interpret the content of the input image, enabling them to generate a coherent fill for the absent sections. Upon establishing the encoder framework, our exploration will extend to identifying and assessing various factors that could influence the fidelity and precision of the reconstructed areas. Specifically, we intend to investigate the effects of the training dataset's composition and the spatial characteristics of the missing segments. Through these analyses, we aim to refine our methodology and enhance the predictive performance of our system. Our preliminary results demonstrate the model's capability to generate convincing and satisfactory reconstructions, aligning with our initial expectations.

2 Method

2.1 Dataset

We evaluate our methodology on two distinct datasets: one comprising images of resolution 32×32 pixels and the other containing images of resolution 128×128 pixels.

CIFAR-10: The CIFAR-10 dataset consists of 60,000 color images, each of 32×32 pixels, divided into 10 classes, with 6,000 images per class. The dataset is split into 50,000 training images and 10,000 testing images. The ten unique classes include: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck, each being mutually exclusive to ensure no overlap among them. Specifically, the "automobile" category includes sedans and SUVs, excluding pickup trucks, while the "truck" category is confined to large trucks only.

Universal Image Embeddings: Our second dataset, tailored for the Google Universal Image Embedding competition, comprises over 130,000 images resized to 128×128 pixels and spans 11 categories. This dataset is an amalgamation of various sources, encompassing both scraped and meticulously curated images to form a rich collection. The categories feature Apparel from the Deep

Fashion Dataset, Artwork, Cars from the Stanford Cars Dataset, Dishes, and others predominantly sourced from Google. Given the dataset's extensive size and complexity, our evaluation concentrates on the Landmark category, anticipating it to offer higher inpainting accuracy. Thus, we have refined the dataset to 30,000 images within this category for a more focused and feasible analysis.

2.2 Network Architecture for 32×32 pixels images:

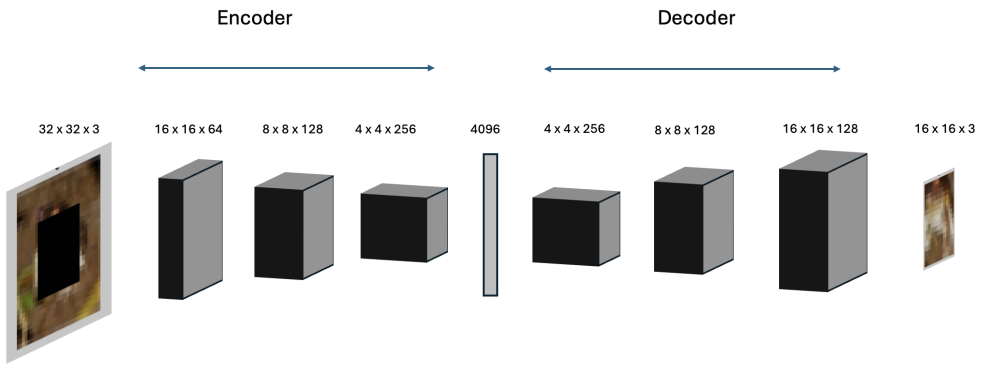

Figure 1: Architecture Visualization for 32 x 32

The proposed encoder-decoder architecture is designed to process images of size 32×32 pixels with 3 color channels (RGB). The model consists of an encoder part, which progressively down-samples the input image to extract and encode salient features, and a decoder part, which up-samples the encoded features to reconstruct the image at a target resolution of 16×16 pixels.

Encoder: The encoding section begins with an input layer accepting images of the specified input shape. It employs a series of convolutional layers, each followed by batch normalization to stabilize learning and accelerate convergence. The convolutional layers use 64, 128, and 256 filters, respectively, with a kernel size of 3×3 and 'ReLU' activation for non-linearity. Each convolutional layer is followed by a 2×2 max-pooling layer with 'same' padding to reduce the spatial dimensions by half, enhancing the model's ability to capture more abstract features at each subsequent layer. The final layer in the encoder outputs a compact, encoded representation of the input image.

Fully Connected Simulation: Post-encoding, the model simulates a fully connected layer by first flattening the encoded feature maps, then passing them through a dense layer with 4096 neurons and 'ReLU' activation. This dense layer acts as a bottleneck, allowing the network to learn a compressed representation of the input data. The output of the dense layer is then reshaped back into a 4-dimensional tensor to be processed by the decoder.

Decoder: The decoder aims to reconstruct the image from the encoded representation. It mirrors the encoder's structure but in reverse, utilizing up-sampling and convolutional layers to progressively increase the spatial resolution of the feature maps. The decoder uses two sets of 3×3 convolutional layers with 256 and 128 filters, respectively, each followed by batch normalization and a 2×2 up-sampling layer to enlarge the feature maps. The final layer of the decoder is a convolutional layer with a kernel size of 3×3, using the 'sigmoid' activation function to output the reconstructed image at the desired target resolution.

The entire model is compiled with the Adam optimizer and mean squared error (MSE) loss function, suitable for regression tasks like image reconstruction. This architecture is expected to perform image inpainting or similar tasks by learning to encode the essential features of the input image and then decode these features to reconstruct or generate the image content.

2.3 Network Architecture for 128×128 pixels images:

This architecture is similar to the previously discussed model, with notable adjustments in the input and output dimensions, as well as in the configuration of the convolutional layers. Designed to accommodate images with an input size of $128 \times 128 \times 3$, the model aims to reconstruct these images

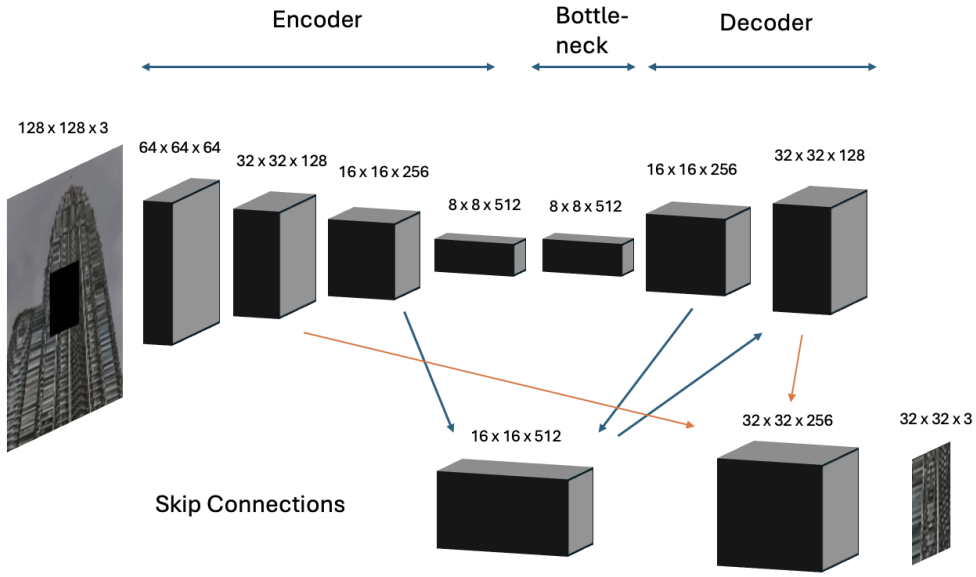

Figure 2: Architecture Visualization for 32 x 32

at a reduced resolution of $32 \times 32 \times 3$. The modified arrangement of convolutional layers further distinguishes this version, tailoring it to the specific requirements of dimensionality reduction and feature preservation within a different spatial context.

Encoder: The encoder segment initiates with an Input layer receiving 128×128 RGB images. It features a series of Convolutional layers, each followed by Batch Normalization to improve training stability and convergence speed. The convolutional layers are configured with 64, 128, 256, and 512 filters respectively, all utilizing a kernel size of $3 \times 3$and 'ReLU' activation for introducing non-linearity. Each Convolutional layer is followed by a 2×2 MaxPooling layer with 'same' padding, systematically reducing the spatial dimensions by half and thereby encoding the input image into a more compact and abstract representation. The final layer of the encoder outputs a condensed feature map of size $8 \times 8 \times 512$, effectively capturing the essential features of the input image.

Bottleneck: At the bottleneck of the network, another Convolutional layer with 512 filters is applied to the encoded feature map, maintaining the spatial dimensions while potentially enabling the model to refine the feature abstraction further.

Decoder: The decoder component, designed to reconstruct the image, begins with an UpSampling layer to increase the spatial dimensions of the feature map from the bottleneck. This is followed by a series of Convolutional layers, each with Batch Normalization, to progressively refine the upsampled features. The architecture incorporates skip connections from the encoder to the decoder, using the Concatenate function, which helps in preserving spatial details by combining feature maps from the encoder with those in the decoder. This approach mitigates the information loss typically associated with deep networks and aids in more accurate reconstruction of the output image. The decoder up-samples and processes the feature maps back to a size of 32×32, followed by a final Convolutional layer that maps the features to the desired output dimension of $32 \times 32 \times 3$, employing a 'sigmoid' activation to ensure the output pixel values are within a normalized range.

Similar to the previous one, the model employs the Adam optimizer and Mean Squared Error (MSE) as the loss function, suitable for the regression nature of the image reconstruction task.

3 Experiments

In our experimental analysis, we conducted training and testing across three distinct scenarios to evaluate the versatility and effectiveness of our model in varying inpainting contexts.

3.1 General Class

In our initial phase of experimentation, the model was tasked with inpainting across a diverse array of image categories, having been trained on the full CIFAR-10 dataset, which encompasses 60,000 images sized at 32×32 pixels, in which we have 50,000 images for training. This broad-based approach aimed to evaluate the model's capability in handling a wide variety of general classes.

Figure 3 below illustrates the progression of training and validation losses throughout the training period. Notably, a sharp decrease in the validation loss is observed within the first two epochs, after which the validation loss stabilizes, indicating a swift adaptation of the model to the general features of the dataset early in the training process.

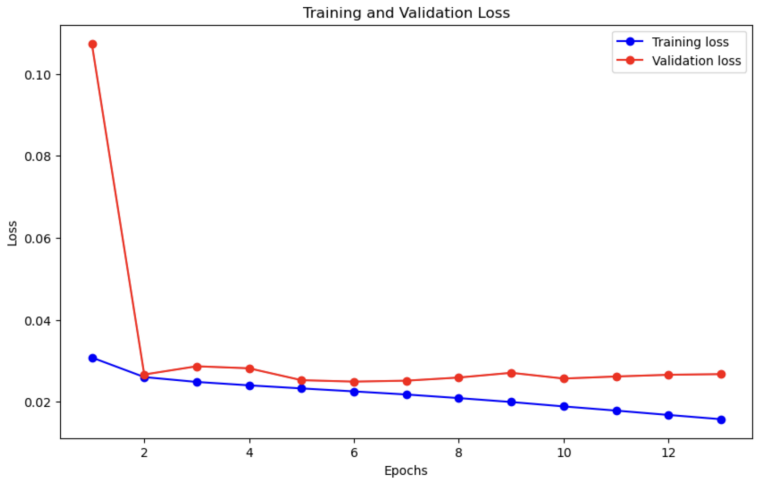

Figure 3: General class training/testing loss history.

Figure 4 and Figure 5 provide a visual comparison between the original 32×32 images (ground truth) and the corresponding inpainted outputs produced from the masked inputs. These figures are divided into two segments: the left segment displays the original and inpainted images from the training dataset, whereas the right segment focuses on the testing dataset. This bifurcation allows for a detailed assessment of the model's inpainting effectiveness, both in scenarios where it has been directly trained on the data (training) and where it encounters unseen data (testing).

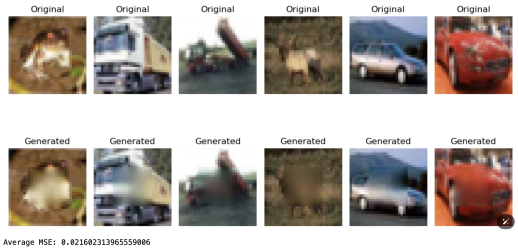

Figure 4: 32×32 general training images.

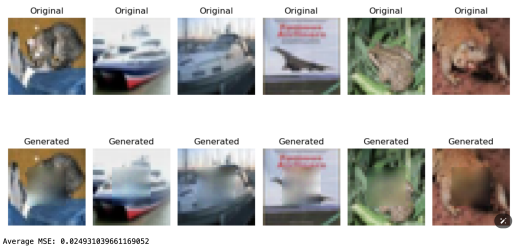

Figure 5: 32×32 general testing images.

3.2 Specific Class

Subsequently, we narrowed our focus to specialized inpainting tasks, training our model exclusively with images from a specific class with the hope of seeing an enhanced performance in targeted scenarios.

Initially, we concentrated on the "truck" category from CIFAR-10, engaging in inpainting experiments with 32×32 images. This subset comprised 5,000 training images and 1,000 testing images, a significantly smaller dataset compared to the general class experiment. This reduction in data volume was instrumental in assessing the model's efficiency and accuracy under constrained conditions.

The loss trends, illustrated in Figure 6, depicted for the truck-class-specific model training show a precipitous decline in training loss, suggesting rapid early learning, which stabilizes as epochs progress. Meanwhile, the validation loss initially spikes, indicating potential overfitting or model instability, but then settles into a steady state, closely paralleling the training loss. This pattern indicates effective learning and generalization, despite early fluctuations in validation loss.

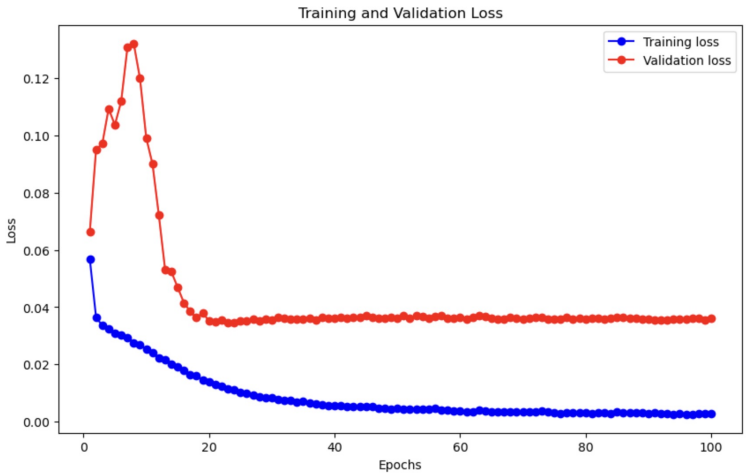

Figure 6: 32×32 specific truck training/testing loss.

Figures 7 and 8 offer a comparative analysis of the original 32×32 truck images against the model's inpainted outputs. The closely matched inpainted images, discernible to the naked eye, underscore the model's adeptness at reconstructing specific class features, indicating its robust performance despite the limited dataset size.

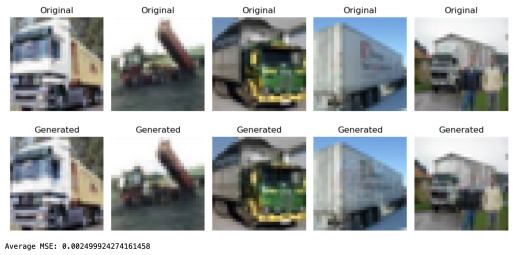

Figure 7: 32×32 truck training images.

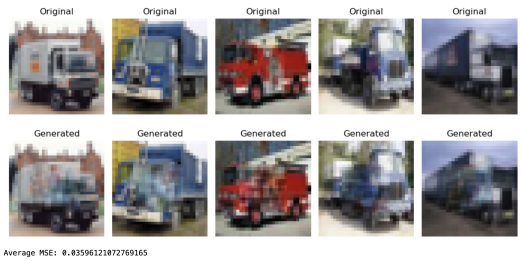

Figure 8: 32×32 truck testing images.

Additionally, we extended our investigation to a more complex dataset by selecting the "landmark" class from Universal Image Embeddings, applying our model to 128×128 images to test its performance on high-resolution inpainting tasks, utilizing a dataset comprised of 24000 training images and 6000 testing images.

The training and validation loss trends for the landmark class, illustrated in Figure 9, reveal a consistent decrease in training loss accompanied by a fluctuating but overall decreasing trend in

validation loss. This indicates the model's learning progression, albeit with varying degrees of adjustment to the validation dataset over time.

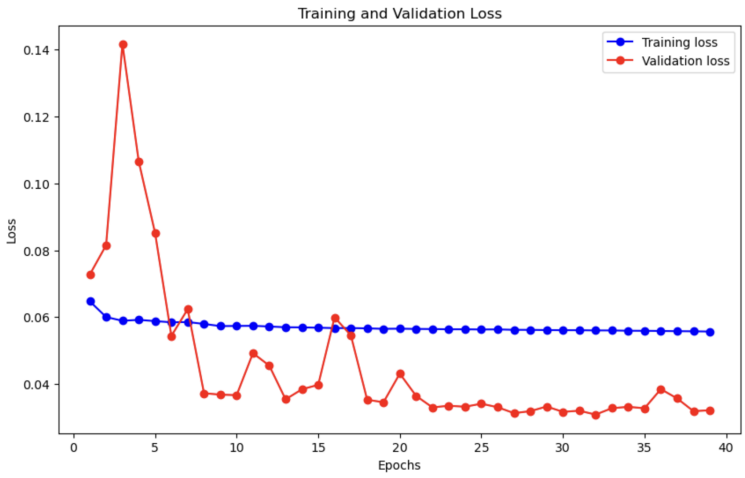

Figure 9: 128 x 128 landmark training/testing loss.

Figures 10 and 11 show the original 128×128 landmark images with the model's inpainted outputs. The outputs, while capturing the general structure, exhibit a noticeable blur within the inpainted regions, reflecting a comparative decline in performance for this dataset size.

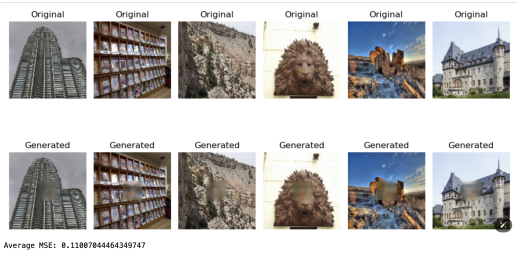

Figure 10: 128 x 128 landmark training images.

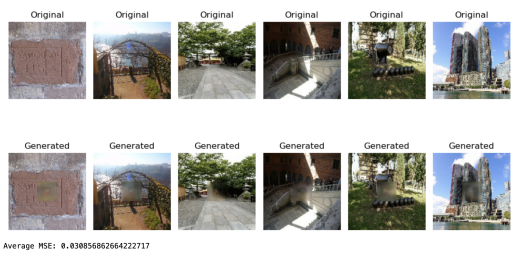

Figure 11: 128×128 landmark testing images.

3.3 General Class & Specific Class Trained Model with the Truck Images

Our experimentation extended to comparing the performance of models trained under different conditions: one with a dataset encompassing all classes within the CIFAR-10 collection and another with a dataset focused exclusively on the truck category. This comparison aimed to ascertain the impact of dataset specificity on the quality of inpainted images, particularly those belonging to the truck class.

Figure 12, presented on the left, illustrates the inpainting results for truck images generated by the model trained on the entire CIFAR-10 dataset, achieving a Mean Squared Error (MSE) of 0.0356. Conversely, Figure 13, displayed on the right, showcases the results from the model trained solely on truck class images from the CIFAR-10 dataset, with a slightly higher MSE of 0.0360. Despite the marginal difference in MSE, a visual assessment of the inpainted images reveals a discernible improvement in quality for the model trained specifically on truck images, which might be attributed to the model's refined understanding and familiarity with the specific features and characteristics of truck images, facilitating a more accurate and detailed inpainting process.

This observation suggests that the focused training on truck images endows the model with a more refined and specialized understanding of truck features, enabling it to perform inpainting tasks with greater precision and detail. The nuanced knowledge acquired from the truck-specific dataset compensates for the slight increase in MSE, highlighting the importance of dataset relevance and

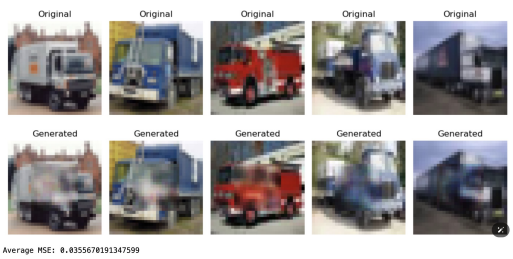

Figure 12: 32 x 32 truck images from general class trained model.

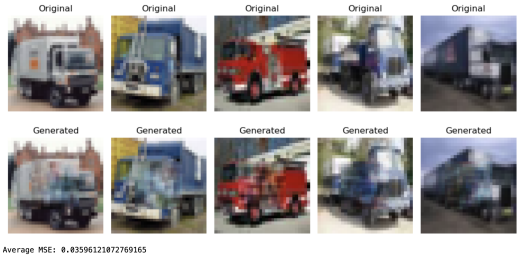

Figure 13: 32×32 truck images from specific truck class trained model.

specificity in enhancing inpainting performance. This finding underscores the potential benefits of tailoring training datasets to the specific subject matter of interest, particularly in applications where high fidelity and detail in inpainting outcomes are paramount.

3.4 Hyper-parameters Tuning

The fine-tuning of hyper-parameters plays a pivotal role in optimizing the performance of machine learning models. In this phase of our experiment, we concentrated on two crucial hyper-parameters: the optimizer and the learning rate. Through systematic testing and evaluation, we aimed to identify the optimal combination that would enhance the efficiency and accuracy of our model across the various inpainting tasks.

Optimizer Selection: We experimented with several popular optimizers, including Stochastic Gradient Descent (SGD), Adam, AdamW, and RMSprop, to determine their impact on the model's performance. Among these, Adam emerged as the slightly superior choice, demonstrating marginally better overall performance compared to the others. This outcome aligns with Adam's well-documented balance of speed and convergence efficiency, making it a robust choice for a wide range of deep learning tasks.

Learning Rate Exploration: The learning rate is a critical parameter that influences the speed and stability of the training process. We explored learning rates beyond the commonly used default of 0.001, testing alternatives such as 0.01 and 0.0001 to gauge their effects on model training and convergence. Our findings indicated that the default learning rate of 0.001 outperformed the others, striking an optimal balance between rapid convergence and the avoidance of overshooting minimal loss landscapes.

Based on these investigations, we adopted Adam as our optimizer with a learning rate of 0.001 for all subsequent model implementations. This combination was identified as the best practice, given its consistent performance advantages across our experiments. By integrating these optimized hyper-parameters, we aimed to further refine our model's capability to accurately perform inpainting tasks, thereby enhancing its overall effectiveness and reliability.

3.5 Metric Evaluation

In our experimental evaluations, we employed the Mean Squared Error (MSE) metric to assess the performance of our model. This evaluation was conducted on two distinct datasets: CIFAR-10 and a collection of Universal Image Embeddings featuring 128×128 resolution landmark images. Specifically, our analysis focused on calculating the MSE solely within the regions covered by the generated masks, comparing these areas directly with the corresponding ground truth mask regions to gauge the accuracy of our model's reconstructions.

MSE Evaluation	32×32 **all classes**	32×32 **truck**	128×128 **landmark**
training	0.0216	0.0025	0.11007
testing	0.02493	0.03596	0.03086

Table 1: Mean Squared Error for 32×32 and 128×128 image datasets

The data in Table 1 reveals that the model trained on 32×32 images across all classes achieves a lower Mean Squared Error (MSE) than the model trained exclusively on 32×32 images of the truck category.

Despite this quantitative outcome, qualitative assessments—based on visual inspection by human evaluators—suggest that the truck-specific model yields more visually convincing reconstructions compared to the general model.

4 Limitation and Future Work

4.1 Limitation

Our study encountered several limitations that point to areas for future exploration and improvement. These limitations primarily revolve around the resolution of inpainted images and the computational resources required for training on high-resolution datasets.

High-Resolution Prediction Challenges: One of the notable challenges we faced was achieving high-resolution predictions in both 32×32 and 128×128 image scenarios. Despite our efforts to enhance image resolution through the introduction of adversarial loss, we observed no significant improvement in this regard. This limitation suggests that the current approach may not effectively capture the finer details necessary for high-resolution inpainting, indicating a potential area for further investigation.

Training Time and Computational Resources: The training process for 128×128 images was particularly time-consuming, highlighting the need for high-performance computational resources. Our limited access to such resources restricted our ability to experiment with larger datasets and more complex model architectures.

4.2 Future Work

Adding Adversarial Loss: Given the inconclusive results with adversarial loss in our current setup, future work could explore different strategies for integrating adversarial components. Adjusting the balance between reconstruction and adversarial loss, experimenting with various adversarial architectures, or implementing more advanced adversarial training techniques may yield better outcomes in enhancing image resolution.

Optimizing Model Architecture: Future efforts could focus on redesigning the model architecture to reduce the number of parameters without compromising the quality of inpainting. Simplifying the network or employing techniques like pruning or knowledge distillation may enable efficient training on less powerful machines while maintaining or improving performance.

Mask Size in High-Resolution Images: For 128×128 images, we experimented with a mask size of 64×64, which proved too large, contributing to more blurred outputs, therefore we ended up to use a 32×32 mask size. Future research could involve optimizing mask size relative to image dimensions to balance the difficulty of the inpainting task with the model's ability to generate coherent and detailed reconstructions.

Hardware, Epochs and Model Depth: Our experiments were constrained by GPU limitations, resulting in fewer training epochs and a reluctance to design deeper architectures. Future studies could secure access to more powerful GPUs, thus being able to increase the number of training epochs, which allows for more thorough learning and exploring deeper, more complex architectures to enhance the model's capacity for high-resolution image inpainting.

In summary, while we have navigated through a series of challenges and identified certain limitations in our current approach, these hurdles pave the way for a broad spectrum of exciting research opportunities. By addressing these limitations through advanced adversarial techniques, architectural optimizations, and leveraging superior computational resources, future work has the potential to significantly advance our current results in image inpainting.

5 Conclusion

In summary, our study embarked on a comprehensive exploration of image inpainting, using the CIFAR-10 and Universal Image Embeddings datasets to examine the effectiveness of our model across a range of scenarios. Our investigation spanned general and specific class inpainting, alongside

meticulous hyper-parameter optimization, revealing insights into the model's ability to perform feature learning and reconstruction for images of varying resolutions.

Despite encountering challenges such as achieving high-resolution predictions and limitations due to computational resources, our research highlighted several avenues for further exploration. The attempt to incorporate adversarial loss, despite not delivering expected results, suggests the need for further refinement to improve image resolution. Additionally, exploring computational efficiency through model architectural adjustments presents a promising direction for facilitating high-resolution inpainting.

Future research holds significant potential to enhance the field of image inpainting through advanced adversarial methods, model optimization, and leveraging improved computational strategies. Addressing the limitations identified in our work, future efforts can lead to substantial advancements in image inpainting, enabling more detailed and realistic reconstructions across diverse imaging scenarios.

Our commitment to this project is demonstrated through our efforts in detailed architectural visualization and tackling the challenges of image inpainting, affirming our belief in the contribution of our work to the field. Looking forward, the insights and foundation established by our study are poised to support further advancements in image inpainting technology.

6 Bonus Point

We believe we deserve the bonus point because we devoted a significant amount of time to creating the architectural visualization figures.

References

[1] Singh, R. (2023) Google Universal Image Embeddings (128x128). Available at: `https://www.kaggle.com/datasets/rhtsingh/google-universal-image-embeddings-128x128`.

[2] Krizhevsky, A. (2009) CIFAR-10 (Canadian Institute for Advanced Research) dataset. Available at: `https://www.cs.toronto.edu/~kriz/cifar.html`.

[3] Dumoulin, V., Shlens, J., and Kudlur, M. (2016) A Learned Representation for Artistic Style. Available at: `https://arxiv.org/pdf/1604.07379.pdf`.