Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

КВАЛІФІКАЦІЙНА РОБОТА

освітній ступінь – бакалавр

НА ТЕМУ: «АЛГОРИТМИ ВІДНОВЛЕННЯ ФОТОРЕАЛІСТИЧНИХ ЗОБРАЖЕННЬ/ PHOTO-REALISTIC IMAGE RESTORATION ALGORITHMS»

Виконав: студент 4-го року навчання
освітньої програми «Прикладна математика»,
спеціальності 113 Прикладна математика

Засядько Матвій Олегович

Керівник: Крюкова Г. В.
Доцент, кандидат
фізико-математичних наук

Рецензент:

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК_____
(підпис)

«_____» _____20___р.

Київ - 2025

Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

ЗАТВЕРДЖУЮ

Зав.кафедри математики,

доцент, кандидат фіз.-мат наук

_____*Чорней Р.К.*

(підпис)

“____”_____2024

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

для кваліфікаційної роботи

студенту 4-го курсу, факультету інформатики

Засядьку Матвію Олеговичу

Тема: «Алгоритми відновлення фотореалістичних зображень/ Photo-realistic image restoration algorithms»

Зміст кваліфікаційної роботи:

Introduction

1. Overview of existing image restoration algorithms
2. Development of a custom algorithm
3. Experimental evaluation and results

Conclusions

References

Дата видачі “____”_____2025 Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Графік підготовки кваліфікаційної роботи

Графік узгоджено “_____”_____2024р.

№ з/п	Перелік робіт	Термін виконання етапу	Підпис наукового керівника	Дата ознайомлення наукового керівника	Примітка
1.	Отримання теми кваліфі-каційної роботи.	25.09.2024			
2.	Ознайомлення з темою кваліфікаційної роботи.	10.10.2024			
3.	Розробка плану та структури роботи.	15.10.2024			
4.	Огляд наукової Літературою.	10.11.2024			
5.	Дослідження основних методів відновлення зображень.	30.12.2024			
6.	Розробка власного алгоритму відновлення зображень.	29.03.2025			
7.	Верифікація розробленого алгоритму на основі кількісних метрик.	05.04.2025			

№ з/п	Перелік робіт	Термін виконання етапу	Підпис наукового керівника	Дата ознайомлення наукового керівника	Примітка
8.	Робота над текстовим оформленням кваліфікаційної роботи	25.04.2025			
9.	Попередній аналіз кваліфікаційної роботи. Виправлення помилок.	20.05.2025			
10.	Попередній захист кваліфікаційної роботи.	23.05.2025			
11.	Захист кваліфікаційної роботи	06.06.2025			

Науковий керівник _____
(ПІБ)

Виконавець кваліфікаційної роботи _____
(ПІБ)

TABLE OF CONTENTS

ANNOTATION 7

INTRODUCTION 8

1. OVERVIEW OF EXISTING IMAGE RESTORATION ALGORITHMS 11

1.1 Image restoration with StyleGAN 11

1.2 Robust StyleGAN inversion 12
1.2.1 Robust Optimization 13
1.2.2 Robust Loss Function 13
1.2.3 Complete Algorithm 13

1.3 Posterior-Mean Rectified Flow 14
1.3.1 Distortion and Perceptual Index 14
1.3.2 Optimal Estimators for the Squared Error Distortion 15
1.3.3 Flow Matching and Rectified Flow 16
1.3.4 Posterior-Mean Rectified Flow (PMRF) 17

1.4 DiffIR: Efficient Diffusion Model for Image Restoration 19
1.4.1 Diffusion Models 19
1.4.2 Method Overview 20
1.4.3 Pretrain DiffIR 20
1.4.4 Diffusion Models for Image Restoration 22
1.4.5 Summary 23

2. DEVELOPMENT OF A CUSTOM ALGORITHM 24

2.1 Dataset and Preprocessing 24

2.2 Edge-Aware Input Design 25

2.3 Model Architecture 25
2.3.1 DeblurEncoder 25
2.3.2 Generator 27
2.3.3 Latent Encoder 28

2.3.4 Overall Architecture and Training Objective....29

2.4 Loss Function....31
2.4.1 L1 Pixel Loss....31
2.4.2 LPIPS (Perceptual) Loss....31
2.4.3 SSIM-Based Structural Loss....32
2.4.4 Total Variation (TV) Loss....32
2.4.5 Latent Consistency Loss....33
2.4.6 Total Objective....33

2.5 Optimization Strategy....34

3. EXPERIMENTAL EVALUATION AND RESULTS....35

3.1 Evaluation Setup....35

3.2 Evaluation Metrics....36
3.2.1 Mean Absolute Error (L1)....36
3.2.2 Mean Squared Error (MSE)....36
3.2.3 Peak Signal-to-Noise Ratio (PSNR)....36
3.2.4 Structural Similarity Index Measure (SSIM)....36
3.2.5 Learned Perceptual Image Patch Similarity (LPIPS)....37

3.3 Evaluation Results....37

3.4 Qualitative Results....38

3.5 Advantages of the work....39

CONCLUSIONS....41

REFERENCES....43

ANNOTATION

In this work, a new algorithm to reconstruct the facial images from degraded inputs is proposed with the visual high-definition reconstruction as its goal. The approach utilizes edge map information in a generative adversarial network (GAN) framework to be able to restore more delicate local structures and semantic content. The architecture is consisting of three parts: a DeblurEncoder which takes a blurred face image and its corresponding edge map, a Generator which recovers high resolution, and a Latent Encoder which supervises in latent space using the consistency loss terms. Training is performed end-to-end all the while using a combined loss function that includes L1 loss, LPIPS perceptual loss, SSIM-based structural similarity loss, total variation loss, and a latent alignment term. Our approach was evaluated on the CelebABlur dataset and achieved comparable results in terms of numerical evaluation and visual quality. The study also compares with some recent state-of-the-art methods such as StyleGAN-based latent optimization, Posterior-Mean Rectified Flow and DiffIR. An advantages of this method are the combination of edge-information and latent-space constraints, which results in the improved quality of generated images, and that all three model components are trained simultaneously, what provides more consistent learning across the latent and pixel spaces enhancing both visual fidelity and structural coherence.

Key words: image restoration, edge maps, GAN, latent supervision, LPIPS, SSIM, CelebABlur, DeblurEncoder, Generator, diffusion models.

INTRODUCTION

Image restoration is a fundamental problem in computer vision that aims to recover high-quality images from degraded inputs caused by motion blur, noise, low resolution, or compression artifacts. The relevance of this task is growing rapidly, as it is applicable to numerous real-world domains such as medical imaging, digital photography, surveillance, and historical media recovery.

Recent advances in deep learning—particularly Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), and Diffusion Models—have led to major breakthroughs in restoration quality. However, most methods rely solely on degraded images and struggle to recover sharp structures when input information is insufficient or ambiguous.

To address this limitation, this research proposes a novel GAN-based image restoration architecture that incorporates edge maps as auxiliary input. This edge information helps the network focus on critical details such as contours and high-frequency transitions, enhancing perceptual and structural fidelity. The model consists of three jointly trained modules: a DeblurEncoder (based on ResNet18) that processes a 4-channel input (RGB + edge map) into a latent vector, a Generator that reconstructs the image from this vector, and a Latent Encoder that maps the clean image to a reference latent code used for supervision. The architecture ensures consistency both in pixel and latent space.

The proposed approach draws theoretical context from modern generative restoration techniques. In particular, three advanced methods were reviewed and analyzed: StyleGAN-based latent optimization [2], Posterior-Mean Rectified Flow [3], and DiffIR [4]. While all of them offer valuable insights into perceptual and semantic reconstruction, this work is primarily inspired by GAN-based latent-space strategies.

The object of the research is the problem of restoring photo-realistic images with fine structural detail based on severely degraded inputs using generative models.

The subject of the research is a custom GAN-based architecture that enhances restoration through edge-aware encoding and latent supervision.

The purpose of the research is to develop and evaluate a novel model for facial image restoration that leverages edge maps and latent-space alignment to produce perceptually convincing and structurally accurate outputs.

To complete this goal, the following research tasks are addressed:

1. Analysis of modern restoration methods and identification of their advantages and limitations;
2. Implementation of a new architecture that uses edge information and encoder-decoder structure;
3. Selection and computation of relevant loss functions for training: L1, LPIPS, SSIM, total variation, and latent consistency loss;
4. Training of the model on the CelebABlur dataset with edge maps extracted using the Canny operator [1];
5. Evaluation of the proposed model using both quantitative metrics (PSNR, SSIM, LPIPS, etc.) and qualitative visual inspection;
6. Summarization of findings and formulation of future directions for extending the model's capabilities.

The methods of the study include: literature review of state-of-the-art GAN and diffusion-based restoration models; architectural design and implementation in PyTorch; application of composite loss functions; experimental training on facial image datasets; visual and metric-based evaluation; and comparison with published baselines.

The informational basis of the study consists of scientific publications on deep generative modeling, image restoration benchmarks, the CelebABlur dataset, and academic sources describing loss functions and optimization strategies. Key references include foundational works on LPIPS, SSIM, StyleGAN, PMRF, DiffIR, and edge detection methods (e.g., Canny).

The novelty of the study lies in the combination of three techniques: (1) integration of edge map as a structural prior, (2) use of latent supervision to maintain semantic consistency, and (3) simultaneous training of all model components from scratch. This

combination improves image sharpness and perceptual alignment in a way not covered by existing methods.

The practical significance of the results is that this architecture can be applied to real-world scenarios where preserving edge sharpness and perceptual detail is critical – such as in restoring degraded photographs or improving surveillance imagery. The training pipeline and model design may also be adapted to other restoration tasks or domains.

This paper is structured as follows: Chapter 1 analyzes state-of-the-art methods for image restoration. Chapter 2 presents the development and training of the proposed model. Chapter 3 reports experimental results and evaluations.

1. OVERVIEW OF EXISTING IMAGE RESTORATION ALGORITHMS

To provide a solid foundation for understanding recent advancements in photo-realistic image restoration, we begin by reviewing several representative approaches in the field.

1.1 Image restoration with StyleGAN

This approach utilizes a pre-trained StyleGAN generator G to approximate the x_{clean} image from its degraded observation y. The idea is to search for a latent code $w \in \mathcal{W}$ such that the generated image $G(w)$ closely resembles the unknown ground truth image y_{clean}, as measured by some perceptual loss function l. The optimization problem is formulated as follows:

$$w = \arg\min_{\tilde{w} \in \mathcal{W}} \mathrm{l}(G(\tilde{w}),\, y_{\text{clean}}) \tag{1}$$

In practical restoration tasks, however, the ground truth image is not available. Instead, we are given only the degraded image $y = f(y_{\text{clean}})$ where f is a degradation function that may be non-injective or non-differentiable. Assuming the existence of a differentiable approximation $\hat{f} \approx f$, the optimization problem becomes:

$$w = \arg\min_{\tilde{w} \in \mathcal{W}} \mathrm{l}(\hat{f}\big(G(\tilde{w})\big),\, y) \tag{2}$$

This formulation can be extended to a sequence of degradations $\{f_i\}_{i=1}^{k}$ yielding a more general objective:

$$w = \arg\min_{\tilde{w} \in \mathcal{W}} \mathrm{l}\left(\left[\hat{f_k} \circ \cdots \circ \hat{f_1}\right]\big(G(\tilde{w})\big),\, y\right) \tag{3}$$

While this naive formulation often yields images with high visual realism, the restored output may exhibit low fidelity to the degraded input. To address this, authors introduced two key techniques:

1. Latent extension, which expands the search space from $w \in \mathcal{W}$ to $w^{+} \in \mathcal{W}^{+}$ thereby increasing representational capacity.
2. The use of more sophisticated optimizers such as Adam [5].

The method introduces a three-phase latent extension scheme, progressively refining the latent space:

Phase 1: A global latent code w generates a coarse restoration x

Phase 2: A layer-wise extension $w^+ \in \mathcal{W}^+$ produces a more detailed output x^+

Phase 3: A filter-wise latent tensor $w^{++} \in \mathcal{W}^{++}$ enables the final prediction x^+

This hierarchical approach balances realism and fidelity without relying heavily on explicit regularization [2].

1.2 Robust StyleGAN inversion

For better and more stable restoration results, the work [2] introduces a strong version of StyleGAN inversion, which customizes for image restoration. Their method revisits and improves each component of the traditional StyleGAN pipeline: latent code representation, optimization strategy, and loss formulation. The authors designed a three-phase latent extension scheme. Each phase incrementally expands the latent representation, initialized by the result of the previous phase.

Phase I: The method begins with a global latent code $w \in R^{512}$ shared across all generator layers. This latent vector is initialized with the empirical mean of the training latent space $E_{\widetilde{w} \in \mathbb{W}}[\widetilde{w}]$ and the style modulation vector at layer l, denoted $s_i^l = A_l(w)$, is computed via the affine transformation $A_l(w)$.

Phase II: The latent representation is extended layer-wise to a matrix $w^+ \in R^{N_L \times 512}$, where N_L is the number of layers in the StyleGAN2 generator [6]. Each row in w^+ is initialized from the global code w, and the corresponding style vector becomes $s_i^l = A_l(w_l^+)$.

Phase III: The most granular expansion is filter-wise, with a tensor $w^{++} \in R^{N_F \times N_L \times 512}$, assigning unique latent vectors to each convolutional filter. The initialization is inherited from w^+, and the modulation becomes $s_i^l = A_l\left(w_{i,l}^{++}\right)$ [2].

1.2.1 Robust Optimization

The authors propose Normalized Gradient Descent (NGD) [7], a variant of stochastic gradient descent where gradients are normalized before each update:

$$\overline{\nabla_w}\, \mathrm{l}(w) = \frac{\nabla_w \mathrm{l}(w)}{\|\nabla_w \mathrm{l}(w)\|_2} \tag{4}$$

where $\nabla_w \mathrm{l}(w)$ is the gradient of the loss function $\mathrm{l}(w)$, $\|\nabla_w \mathrm{l}(w)\|_2$ – is the Euclidian (L2) norm of the gradient, and $\overline{\nabla_w}\, \mathrm{l}(w)$ is the normalized gradient.

This choice is to make the loss scale-invariant, and thus avoid any dependence among various loss magnitudes, as well as not requiring to tune the learning rate between different tasks. The complete optimization process is performed in three stages, with larger step sizes for higher-level latent extension (0.08 → 0.02 → 0.005) [2].

1.2.2 Robust Loss Function

To improve robustness, the authors introduce a multi-resolution loss function:

$$\mathrm{l}_{\mathrm{MR}}(x, y) = \sum_{i=1}^{k} \mathrm{l}_{\mathrm{LPIPS}}\left(\phi\left(x, 2^i\right), \phi\left(y, 2^i\right)\right) \tag{5}$$

Here, $\mathrm{l}_{\mathrm{LPIPS}}$ – is the Learned Perceptual Image Patch Similarity loss [8], $\phi\left(\cdot, 2^i\right)$ denotes average pooling by a factor of 2^i, allowing perceptual comparison at multiple scales. In experiments with 1024^2 resolution images, k is set to 6. The final loss function combines this term with an l_1 pixel loss:

$$\mathrm{l} = \lambda_{L1} \cdot \mathrm{l}_{L1} + \mathrm{l}_{\mathrm{MR}}, \text{ where } \lambda_{L1} = 0.1 \tag{6}$$

This combination promotes both fine-detail reconstruction and global perceptual quality [2].

1.2.3 Complete Algorithm

The overall algorithm consists of three components: a progressive latent code expansion, a normalized optimization approach and a multi-scale perceptual loss. It proceeds in three stages:

In Phase I, a single global latent vector w is optimized to approximate the target image.

Phase II further expands the latent space layer by layer to accommodate more detailed characteristics.

In Phase III, a filter-wise latent tensor w^{++} is used for fine-grained modulation

Each stage recycles and improves the result of the preceding one. Optimization proceeds based on Normalized Gradient Descent and the loss is the pixel accuracy and the perceptual quality of levels. The process is task-indenpent and fixed, with all hyperparameters preset [2].

1.3 Posterior-Mean Rectified Flow

1.3.1 Distortion and Perceptual Index

In photo-realistic image restoration, evaluating the performance of an algorithm typically involves two complementary criteria: distortion and perceptual quality.

The distortion criterion measures how closely the reconstructed image $\hat{X}$ approximates the original clean image X, on average. This is typically formalized as the expected distortion $E\left[\Delta\left(X, \hat{X}\right)\right]$, where $\Delta(x, \hat{x})$ is a distance function between the original and reconstructed images. Common examples include the mean absolute error $\|x - \hat{x}\|_1$, the mean squared error $\|x - \hat{x}\|_2^2$, and perceptual similarity metrics such as LPIPS [6].

However, for PIR tasks, the goal is not only to minimize distortion, but also to ensure that the generated images look realistic to. One of the most reliable ways to assess perceptual quality is through human judgment – how convincing the appear to a person. While this provides trustworthy feedback, it is impractical for model optimization or large-scale evaluation due to its high resource demands.

As a result, researchers rely on perceptual indices that approximate human judgment using statistical measures of similarity between the distribution of ground-truth images p_X

and that of reconstructed images $p_{\hat{X}}$. These perceptual indices are typically expressed as divergence measures $d(p_X, p_{\hat{X}})$, such as the Kullback-Leibler divergence or the Wasserstein distance. Since computing such divergences for high-dimensional image distributions is often intractable, practical approximations are used. These metrics evaluate how closely the statistical features of generated images match those of real images [3].

1.3.2 Optimal Estimators for the Squared Error Distortion

Image restoration involves a fundamental trade-off between distortion and perceptual quality [9]. This relationship is formalized via the distortion-perception function:

$$D(P) = \min_{\hat{p}_{X|Y}} E\left[\Delta\left(X, \hat{X}\right)\right] \quad subject\ to \quad d(p_X, p_{\hat{X}}) \leq P \tag{7}$$

Two key cases of interest are:

1. $D(\infty)$: no perceptual constraint – minimized by the posterior mean $\hat{X}^* = E[\,X \mid Y\,]$ which achieves lowest MSE but produces overly smooth results;

2. $D(0)$: perfect perceptual match – minimized by solving:

$$\min_{\hat{p}_{X|Y}} E\left[\left\|X - \hat{X}\right\|^2\right] \quad subject\ to \quad p_{\hat{X}} = p_X \tag{8}$$

In the work [10] was shown, that this problem can be solved via optimal transport between the posterior mean distribution $p_{\hat{X}^*}$ and the ground truth distribution p_X. The solution involves two steps:

1. Compute the posterior mean $\hat{x}^* = E[\,X \mid Y = y\,]$
2. Sample from the optimal transport plan

$$p_{U|V}(\cdot \mid \hat{x}^*), \ \ where \ \ (U, V) \sim \Pi(p_X, p_{\hat{X}^*})$$

This estimator, denoted $\hat{X}_0$, preserves perceptual quality and improves MSE over posterior sampling [3].

While sampling from the true posterior $p_{X|Y}$ also ensures a perfect perceptual index, it generally leads to higher MSE than $\hat{X}_0$. Important, that sampling from $p_X \mid \hat{X}^*$ offers no

MSE advantage over posterior sampling, reinforcing the optimality of the transport-based solution [3].

1.3.3 Flow Matching and Rectified Flow

Flow matching algorithms are generative models defined via ODE:

$$\frac{dZ_t}{dt} = v(Z_t, t) \tag{9}$$

Here, Z_t represents a forward process interpolating between a source distribution p_{Z_0} (e.g., standard Gaussian noise) and a target distribution p_{Z_1} (natural images). The vector field v defines the direction and speed of flow in latent space $Z_0 \sim p_{Z_0}$ and integrating the ODE forward in time, one obtains samples from the desired target distribution p_{Z_1} [11].

However, since many different vector fields can satisfy the same ODE, a key challenge is selecting one with favorable theoretical and practical properties, such as existence, uniqueness, and stability of the solution.

Rectified Flow [11] is a specific flow matching method that defines the forward process linearly:

$$Z_t = tZ_1 + (1-t)Z_0 \tag{10}$$

This formulation connects Z_0 and Z_1 by straight lines in latent space. Although conceptually simple, it requires access to Z_1 at all time steps $t \in [0,1)$ which is not feasible in generative scenarios. To overcome this, a causal vector field is used:

$$v_{\mathrm{RF}}(Z_t, t) = E[\, Z_1 - Z_0 \mid Z_t \,] \tag{11}$$

Under mild conditions, this field guarantees a unique solution to the ODE and allows for sample generation using only initial data Z_0. Moreover, integrating the ODE with v_{RF} approximates the optimal transport map from p_{Z_0} to p_{Z_1}, especially when the joint distribution p_{Z_0,Z_1} is close to optimal coupling.

To learn v_{RF}, one minimizes the expected squared error:

$$E\left[\int_0^1 \|(Z_1 - Z_0) - v_\theta(Z_t, t)\|^2 \, dt\right] \tag{12}$$

where v_θ is a neural network trained on samples from the joint distribution p_{Z_0,Z_1} [11] [3].

1.3.4 Posterior-Mean Rectified Flow (PMRF)

The Posterior-Mean Rectified Flow algorithm aims to approximate the optimal image restoration estimator $\hat{X}_0$, which achieves minimal mean squared error under a perfect perceptual constraint. The method consists of two training stages:

Stage 1: Posterior Mean Estimation

A regression model f_ω is trained to predict the posterior mean $\hat{X}^* = E[X \mid Y]$ by minimizing the MSE:

$$\omega^* = \arg\min_{\omega} E[\|X - f_\omega(Y)\|^2] \tag{13}$$

where X – ground truth image, Y – degraded image, $\hat{X}^*$ - the expected value of the clean image given the degraded observation, $f_\omega(Y)$ – a regression model parameterized by ω used to approximate $\hat{X}^*$, $\|\cdot\|^2$ denotes MSE, ω^* is a set of parameters that minimize the expected squared error over data distribution.

This stage can be skipped if a reliable off-the-shelf predictor is available.

Stage 2: Vector Field Training via Rectified Flow

A neural network v_θ is trained to learn the rectified flow vector field by minimizing the loss:

$$\theta^* = \arg\min_{\theta} E\left[\int_0^1 \|(X - Z_0) - v_\theta(Z_t, t)\|^2 \, dt\right] \tag{14}$$

where the forward process is defined as:

$$Z_t \coloneqq tX + (1-t)Z_0, \quad Z_0 \coloneqq f_{\omega^*}(Y) + \sigma_s \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{15}$$

Here, Z_0 is the starting point of the rectified flow trajectory, initialized by the predicted posterior mean f_{ω^*} perturbed with Gaussian noise, $\epsilon \sim \mathcal{N}(0, I)$ – is a standard Gaussian noise, the noise parameter σ_s ensures regularity when mapping between low- and

high-dimensional manifolds, though it should remain small to avoid degrading MSE performance [3].

Inference Procedure

The restoration process begins by initializing a noisy estimate of the posterior mean based on the degraded input y.

$$\hat{x}_0 = f_{\omega^*}(y) + \sigma_s \epsilon \tag{16}$$

The rectified flow is then solved using Euler integration:

$$\hat{x}_{i+1} = \hat{x}_i + \frac{1}{K} \cdot v_{\theta^*}(\hat{x}_i, t_i), \quad t_i = \frac{i}{K}, \quad i = 0, \ldots, K-1 \tag{17}$$

The final output $\hat{x}_K$ serves as the restored image [3].

Theoretical Guarantees

Under the assumption that $\sigma_s = 0$ and the ODE solution exists and is unique, the PMRF output $\hat{Z}_1$ satisfies:

- $p_{\hat{Z}_1} = p_X$ (perfect perceptual index)
- MSE is no worse than that of posterior sampling,
- And strictly better when the conditional variance $\mathrm{Var}\left(X - \hat{X}^*\right) \mid Z_t$ is non-degenerate for almost all $Z_t \in \mathrm{supp}\left(p_{Z_t}\right)$

$\mathrm{supp}\left(p_{Z_t}\right)$ denotes the support of the probability density function p_{Z_t}, i.e., the set of all points Z_t in the input space for which the probability density is non-zero:

$$\mathrm{supp}\left(p_{Z_t}\right) = \left\{ z \in R^{H \times W \times C} \,\middle|\, p_{Z_t}(z) > 0 \right\} \tag{18}$$

These properties make PMRF a principled alternative to posterior sampling for perceptual image restoration [3].

1.4 DiffIR: Efficient Diffusion Model for Image Restoration

1.4.1 Diffusion Models

Diffusion models (DMs) [12] are probabilistic models designed to add step by step noise to structured inputs and then learn to reverse this noising process to recover the original data. In the context of image restoration, DMs are used to extract accurate prior representations by modeling the image distribution and leveraging it to guide reconstruction.

During the forward process, a clean image x_0 is gradually noised over T time steps using a Markov chain of Gaussian transitions. At each step t, the transition from x_{t-1} to x_t follows:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}\right) \tag{19}$$

Where x_t – noised image at time-step t, β_t – predefined scale factor, and $\mathcal{N}$ represents the Gaussian distribution.

This can be simplified to express the marginal distribution of x_t directly given x_0, using the accumulated product of noise parameters $\bar{\alpha}_t$:

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{20}$$

where $\alpha_t = 1-\beta_t,\ \bar{\alpha}_t = \prod_{i=0}^{t}\alpha_i$.

In the reverse process, the model samples an initial latent $x_T \sim \mathcal{N}(0, I)$ and denoises it over T steps to recover a clean sample. The reverse transition is also Gaussian, with learned mean and variance:

$$p(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \mu_t(x_t, x_0), \sigma_t^2\mathbf{I}) \tag{21}$$

where mean $\mu_t(x_t, x_0) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \epsilon\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\right)$ and variance $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$.

The key uncertain variable in the reverse process is the noise ϵ contained in x_t, which is estimated by a neural network $\epsilon_\theta(x_t, t)$. To train the model, a clean image x_0 is noised at a random timestep t, and the denoising network is optimized to predict the added

noise. The objective is to minimize the MSE between the true noise and the network output [12]:

$$\nabla_\theta \left\| \epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t} x_0 + \epsilon\sqrt{1-\bar{\alpha}_t}, t\right)\right\|_2^2 \quad (22)$$

This approach enables the model to effectively learn the reverse dynamics of the diffusion process, ultimately allowing it to generate high-fidelity restorations from noisy or degraded observations [4].

1.4.2 Method Overview

DiffIR is an efficient diffusion-based framework for image restoration. It consists of three key compony:

1. Compact IR Prior Extraction Network (CPEN)
2. Dynamic IRformer (DIRformer)
3. Denoising network

The method is trained in two stages. First, CPEN and DIRformer are jointly trained to extract and utilize a compact IR prior (IPR) for guiding restoration. Then, a diffusion model is trained to estimate and refine the IPR using a denoising network [4].

1.4.3 Pretrain DiffIR

In the first training stage, DiffIR learns to generate a compact prior in inner product space for image restoration. This is realized via two jointly trained networks: CPEN and DIRformer.

Compact IR Prior Extraction Network consists of residual and linear blocks and is expected to achieve a low-dimensional yet informative IPR from the concatenated ground-truth and low-quality images. The input is downsampled using a PixelUnshuffle operation, which rearranges spatial information into the channel dimension to downsample them, to obtain the input for $CPEN_{S1}$. Then $CPEN_{S1}$ extract the The IPR $Z \in \mathbb{R}^{4C'}$ as:

$$Z = CPEN_{S1}\left(PixelUnshuffle\left(Concat\left(I_{GT}, I_{LQ}\right)\right)\right) \quad (22)$$

The extracted prior Z is passed to DIRformer, which has a U-Net-like architecture based on dynamic transformer blocks. These blocks include two key components:

1. Dynamic Multi-Head Transposed Attention (DMTA) to understand how different parts of the image relate to each other, even if they are far apart.
2. Dynamic Gated Feed-Forward Network (DGFN) for encoding local features.

Both DMTA and DGFN use the IPR Z as a dynamic modulation signal to refine feature maps. The modulation in DGFN is computed as:

$$F' = W_l^1 Z \odot Norm(F) + W_l^2 Z \tag{23}$$

where $\odot$ indicates element-wise multiplication, $Norm$ denotes layer normalization, $W_l -$ linear layer, F and $F' \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$ are input and output feature maps respectively, and $W_l^1 Z, W_l^2 Z \in \mathbb{R}^{\hat{C}}$ [2].

In DMTA, global attention is computed more efficiently by projecting feature maps into low-dimensional queries, keys, and values using 1×1 and 3×3 convolutions, and then reshaping them to form a transposed-attention map.

The attention output is computed as:

$$\hat{F} = W_c \hat{V} \cdot Softmax\left(\hat{K} \cdot \frac{\hat{Q}}{\gamma}\right) + F, \tag{24}$$

where, $\hat{Q}, \hat{K}$ and $\hat{V}$ represent the query, key, and value matrices after reshaping, γ is a learnable scaling parameter and multi-head attention is applied as in [13] [14].

In DGFN, spatially local information is aggregated using 1×1 and 3×3 convolutions. The network uses a gating mechanism to better control how information passes through, helping it emphasize important features and suppress less useful ones. The output of DGFN is defined by:

$$\hat{F} = GELU((W_d^1 W_c^1 F') \odot W_d^2 W_c^2 F' + F. \tag{25}$$

CPEN and DIRformer are trained jointly to ensure that the IPR contributes effectively to restoration. The reconstruction loss used during this stage is the L1 distance between the restored and ground-truth high-quality images:

$$L_{rec} = \left\| I_{GT} - \hat{I}_{HQ} \right\|_1, \tag{26}$$

where I_{GT} and $\hat{I}_{HQ}$ are ground-truth and restored high-quality images, $\| \cdot \|_1$ denotes L_1 norm [4].

1.4.4 Diffusion Models for Image Restoration

In the second training stage, DiffIR leverages a diffusion model to estimate a compact IR prior. Using the pretrained $CPEN_{S1}$ the IPR $Z \in \mathbb{R}^{4C'}$ is first extracted. Then, authors use a Gaussian-based diffusion process to gradually transforme Z it into a noisy version Z_T, this process is defined as:

$$q(Z_T \mid Z) = \mathcal{N}\left(Z_T; \sqrt{\bar{\alpha}_T} Z, (1 - \bar{\alpha}_T)\mathbf{I}\right) \tag{27}$$

where, T is the total number of diffusion steps, and $\bar{\alpha}_T$ is the product of the noise schedule parameters [4].

Due to the compactness of the IPR representation, DiffIR requires fewer reverse iterations and less computational overhead compared to conventional diffusion models [15]. Traditional approaches typically sample a single timestep $t \in [1, T]$ and optimize the denoising network for that step only. Moreover, they do not jointly train the denoising network and the restoration decoder, which can lead to error accumulation [4].

In contrast, DiffIR initiates the reverse process from Z_T and performs the full sequence of denoising iterations, using:

$$\hat{Z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\hat{Z}_t - \epsilon \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\right) \tag{28}$$

Here, ϵ is the noise component, which is estimated at each step using a denoising network and $CPEN_{S2}$. The model does not include the variance term, this simplification was found to improve performance and stability [4].

To condition the denoising on the degraded image, a conditional vector $D \in \mathbb{R}^{4C'}$ is extracted using $CPEN_{S2}$:

$$D = CPEN_{S2}\left(PixelUnshuffle\left(I_{LQ}\right)\right), \tag{29}$$

The conditional vector D, noisy estimate $\hat{Z}_t$, and the corresponding timestep t are jointly passed into the denoising model $\epsilon_\theta(\cdot)$, which estimates the noise component ϵ. The predicted noise is then inserted into into the denoising update rule to compute the next estimate $\hat{Z}_{t-1}$.

After T iterations, the final denoised prior $\hat{Z}$ is obtained and passed to the DIRformer for image restoration. A combined training loss is applied to the denoising network, $CPEN_{S2}$, and DIRformer, defined as follows:

$$\mathcal{L}_{diff} = \frac{1}{4C'}\sum_{i=1}^{4C'}\left|\hat{Z}_i - Z(i)\right|, \ \mathcal{L}_{all} = \mathcal{L}_{rec} + \mathcal{L}_{diff}, \tag{30}$$

The total loss consists of the reconstruction loss $\mathcal{L}_{rec}$, and diffusion consistency loss $\mathcal{L}_{diff}$ [4].

During inference, only the reverse diffusion process is used. The conditional vector D is extracted from the LQ image, and a noise sample $\hat{Z}_T \sim \mathcal{N}(0, I)$ is denoised over T steps to recover the refined IPR $\hat{Z}$, which is then used by DIRformer to restore the final high-quality output [4].

1.4.5 Summary

DiffIR introduces an efficient two-stage framework for image restoration based on diffusion models. In the first stage, a compact IR prior is extracted and used to guide restoration. In the second stage, a diffusion model refines this prior through a lightweight denoising process. The method optimizes all components simultaneously, enabling accurate and computationally efficient restoration.

2. DEVELOPMENT OF A CUSTOM ALGORITHM

This part of the work describes a custom-designed algorithm for image restoration based on GAN principles. It differs from regular models by providing a custom method to gain structure information by using the blurred image along with its corresponding edge map as 4 channel input. This fusion helps the network more effectively retain edges and reconstruct fine image details.

A key feature of this model is that all three parts are trained together at the same time: the deblurring encoder, that takes concatenated blurred image and its edge map, the generator that creates the restored image, and latent encoder that processes the ground truth clean image to produce latent vector, which then is compared with one from deblurring encoder. Training everything together instead of separately helps the model learn more effectively. It keeps the internal representations consistent and leads to more accurate results.

2.1 Dataset and Preprocessing

The model is trained and evaluated on the CelebABlur dataset, a synthetically blurred version of the CelebA face dataset. Each sample includes a high-resolution ground-truth image and a corresponding low-quality (blurred) version.

All images are resized to a fixed resolution of 128×128 pixels and are represented as tensors in $\mathbb{R}^{3\times128\times128}$, where each channel corresponds to a normalized RGB value.

To facilitate stable optimization and ensure efficient convergence, image pixel values in the original $[0, 255]$ are mapped to $[-1, 1]$ through the following transformation:

$$x_{\text{norm}} = \frac{x}{255} \cdot 2 - 1 = \frac{x-127.5}{127.5} \tag{31}$$

Normalization is applied independently to each color channel and can be reformulated as:

$$x_{\text{norm}} = \frac{x-\mu}{\sigma}, \text{ with } \mu = 0.5,\ \sigma = 0.5 \tag{32}$$

Here, x denotes the input pixel value, x_{norm} is the result after normalization, μ – the mean of the normalized range, σ – the scaling factor that maps $[0, 1]$ to $[-1, 1]$ [13].

This normalization follows common practice in deep convolutional GAN models for stable training [17].

2.2 Edge-Aware Input Design

An edge map is a binary image that highlights the areas where the pixel intensity changes the most, usually along object edges or structural outlines. In image restoration, this regions often contain fine details that get lost during degradation like blurring. By giving the model this structural information directly, we help it better recover sharp edges and restore small but important structures in the image.

In this work, the edge map is used as an additional input channel, concatenated with the blurred RGB image to form a 4-channel tensor. This allows the model to make predictions based not only on the image itself but also on its structural details. The edge map is generated using the well-established Canny edge detector [1], which optimizes three key criteria: high detection accuracy (maximized signal-to-noise ratio), good localization (minimized deviation from the true edge), and single response per edge [1]. These properties make it a robust and widely adopted choice in both classical and modern vision pipelines.

2.3 Model Architecture

2.3.1 DeblurEncoder

The DeblurEncoder is built upon the ResNet-18 architecture, which utilizes the principle of residual learning [18]. In this approach, instead of directly learning a mapping $H(x)$, the network learns a residual function $F(x) \coloneqq H(x) - x$, reformulating the original transformation as $H(x) = F(x) + x$. This approach helps avoid the training difficulties

that usually appear in very deep networks. Instead of learning the entire transformation from the beginning, the model focuses only on learning the small changes needed to improve the result. [18].

Residual blocks in ResNet consist of two or more convolutional layers and an identity shortcut connection, which allows the input x to be directly added to the output of the block $F(x)$, forming the expression:

$$y = F(x, \{W_i\}) + x \tag{33}$$

where $F(x, \{W_i\})$ denotes the residual function, typically composed of convolutional layers with batch normalization and ReLU activations [19]. When the input and output dimensions differ, a projection shortcut $y = F(x) + W_s x$ is applied using a 1×1 convolution [18].

These identity mappings introduce no additional parameters and have been shown to provide sufficient capacity for stable training, while improving optimization by acting as a form of preconditioning. In convolutional networks, such element-wise additions are performed channel-wise over feature maps, making the method particularly suitable for deep feature encoding tasks like ours.

In our implementation, we adopt the ResNet-18 configuration due to its balance between depth, performance, and computational efficiency. To accommodate our custom input, we modify the initial convolutional layer to accept 4-channel tensors instead of the standard 3-channel RGB images. The fourth channel contains the edge map.

The core of the encoder consists of the standard ResNet-18 residual blocks, which progressively reduce spatial dimensions and increase feature depth. After the final residual layer, a global average pooling is applied, followed by a fully connected layer that projects the features into a 512-dimensional latent vector $z \in \mathbb{R}^{512}$ (Figure 1).

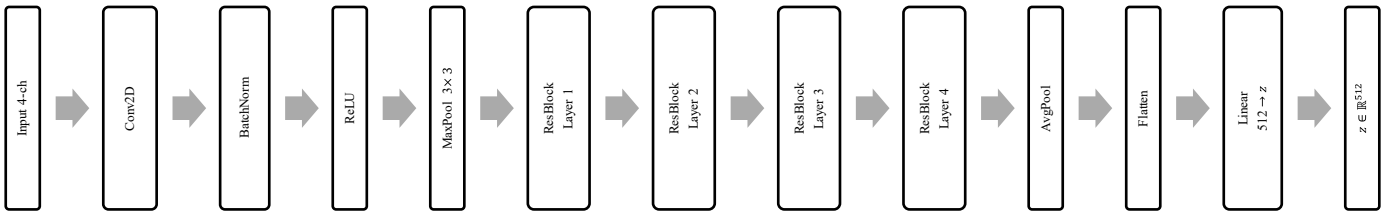

Figure 1: Deblur Encoder architecture

This latent representation captures both semantic and structural properties of the input, and is passed to the generator network for image reconstruction. The combination of edge-aware input and a residual encoder backbone enables the model to focus on fine-grained details and enhance high-frequency content in restored images.

2.3.2 Generator

In constructing the generator component of GAN-based image restoration model, we adopted a design inspired by the Deep Convolutional Generative Adversarial Network (DCGAN) architecture [17]. This architecture is effective in generating high-quality images through a series of transposed convolutional layers, also referred to as deconvolutional layers [20].

The generator begins with a fully connected layer that transforms the latent vector $z \in \mathbb{R}^{512}$ into a 4×4×512 tensor. This tensor is then progressively upsampled through a sequence of transposed convolutional layers, each followed by batch normalization and ReLU activation functions [19], except for the final layer, which uses a Tanh activation to produce the output image. The architecture is as follows (Figure 2):

1. Fully Connected Layer [17]**:** Projects the latent vector to a 4×4×512 tensor.
2. Transposed Convolutional Layers [20]: A series of layers that double the spatial dimensions at each step (i.e., 4×4 → 8×8 → … → 128×128), reducing the number of feature maps accordingly (512 → 256 → 128 → 64 → 3).

3. Batch Normalization and ReLU: Applied after each transposed convolutional layer to stabilize training and introduce non-linearity.
4. Tanh Activation [17]: Used in the final layer to ensure the output pixel values are in the range [−1, 1].

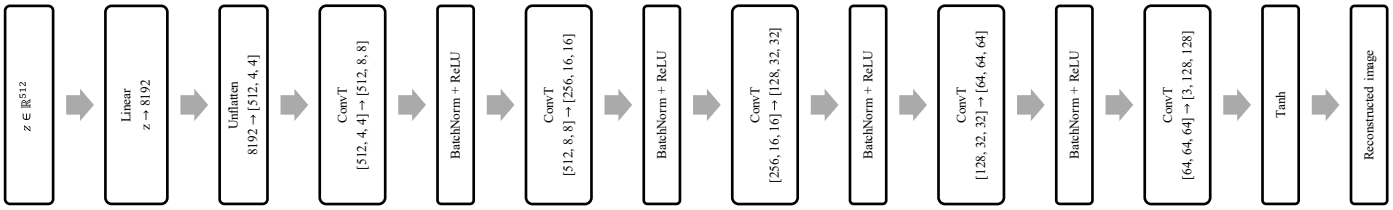

Figure 2: Generator architecture

This design choice aligns with the DCGAN guidelines, which recommend using transposed convolutions for upsampling and avoiding pooling layers to allow the network to learn its own spatial upsampling. The use of batch normalization and ReLU activations further contributes to stable training and improved convergence [17].

2.3.3 Latent Encoder

The Latent Encoder is used to turn a clean, high-quality image into a compact vector that captures its main features. This vector acts like ideal version in the training process. It helps guide the model by making sure that the output from the blurred image looks similar in the latent space to the one from the clean image. This helps the model learn to restore images in a more consistent and meaningful way.

During training, the Latent Encoder receives the clean target image $x_{gt} \in \mathbb{R}^{3\times128\times128}$ and transforms it into a compact latent representation $z_{gt} \in \mathbb{R}^{512}$. This vector is then compared to the corresponding output z_{deblur} generated by the DeblurEncoder, which processes the degraded version of the same image, enriched with edge map input. To enforce alignment in the latent space, we define the latent consistency loss as the mean squared error between these two representations:

$$\mathcal{L}_{latent} = \left\| z_{deblur} - z_{gt} \right\|_2^2 \tag{34}$$

This loss encourages both encoders to embed their respective inputs into a shared latent manifold, ensuring that reconstructions derived from z_{deblur} preserve the high-level structure of the original clean image.

The architecture of the Latent Encoder (Figure 3) follows a progressively deepening convolutional pathway. It consists of five convolutional blocks with increasing feature depth (from 64 to 512 channels), each followed by batch normalization and LeakyReLU [22] activation. After the final convolutional block, the 4×4 feature map is reshaped into a flat vector and transformed through a linear layer to produce a 512-dimensional latent vector. This design provides sufficient representational capacity to capture semantic content while maintaining compatibility with the generator.

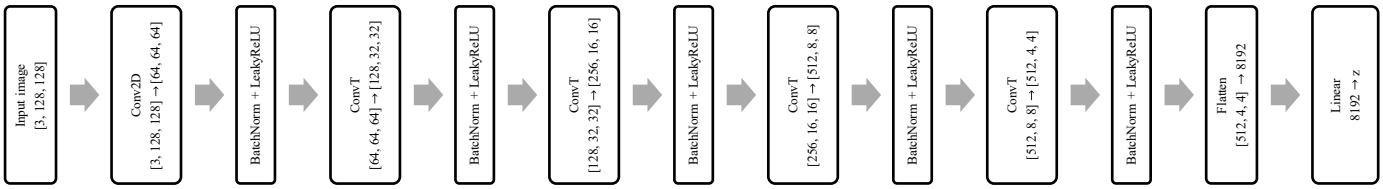

Figure 3: Latent Encoder architecture

Using this additional latent supervision improves the semantic alignment between the clean and reconstructed images. Similar approaches have been effectively employed in unsupervised and semi-supervised generative frameworks [23], [24], where latent space constraints contribute to training stability, better convergence, and more accurate reconstructions

2.3.4 Overall Architecture and Training Objective

The proposed model is built around a three-part architecture consisting of the DeblurEncoder, the Generator, and the Latent Encoder. These components are trained jointly in an end-to-end fashion to reconstruct high-quality images from degraded inputs enriched with edge information.

- The DeblurEncoder takes as input a four-channel tensor composed of a blurred RGB image and its corresponding edge map. It encodes this input into a latent vector $z_{deblur} \in \mathbb{R}^{512}$.
- Based on the latent representation, the Generator reconstructs high-resolution RGB image.
- Simultaneously, the Latent Encoder processes the clean target image to produce a reference latent vector z_{gt}, enabling latent-space supervision through consistency loss.

This architecture ensures that the reconstructed output is not only visually accurate but also structurally coherent in latent space.

Training is performed using a composite loss function that combines multiple objectives: L1 pixel loss, perceptual similarity (LPIPS), structural similarity (SSIM), total variation regularization, and latent consistency loss. All components are optimized jointly (Figure 4) using the Adam optimizer [5]. The model is trained from scratch, and learning rate scheduling is employed to stabilize convergence .

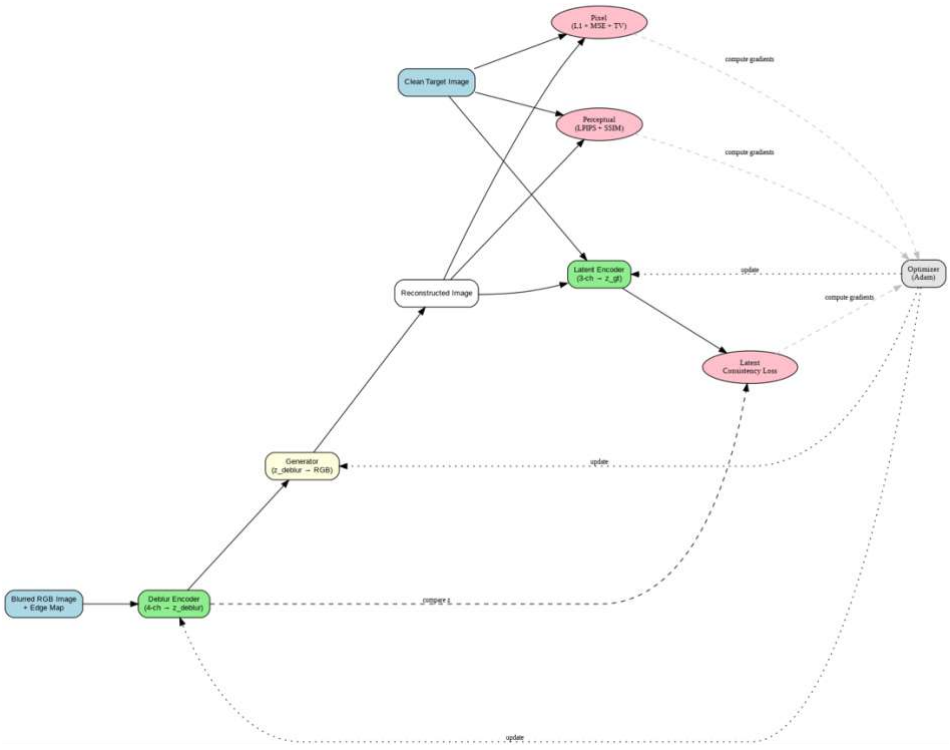

Figure 4: Training process

This coordinated training strategy enables the network to learn both high-level semantic alignment and fine-grained pixel-level details, resulting in reconstructions that exhibit both fidelity and realism.

2.4 Loss Function

The image restoration task addressed in this work benefits from a composite loss function that captures multiple dimensions of quality – from low-level pixel accuracy to high-level perceptual realism and latent consistency. The total loss combines five individual objectives, each contributing uniquely to the optimization process. Below, we describe each component in detail.

2.4.1 L1 Pixel Loss

The L1 loss computes how much the predicted image $\hat{x}$ deviates from ground-truth image x by averaging the absolute pixel-wise difference:

$$\mathcal{L}_{L1} = \|\hat{x} - x\|_1 \tag{35}$$

This loss enforces pixel-level accuracy, encouraging the output to closely match the target image in terms of absolute intensity values [25]. Unlike L2 loss, which overly penalizes large deviations, the L1 loss provides a more robust signal against outliers and helps preserve overall image brightness and structure [26].

2.4.2 LPIPS (Perceptual) Loss

To complement pixel-level accuracy with perceptual similarity, the model uses the LPIPS loss:

$$\mathcal{L}_{LPIPS} = LPIPS(\hat{x}, x) \tag{36}$$

where LPIPS is the Learned Perceptual Image Patch Similarity loss [8].

This metric computes the distance between deep features extracted from a pretrained network (we've used AlexNet), comparing how similar the images are from a human

perceptual standpoint. By focusing on high-level visual characteristics such as texture and object boundaries, this loss helps the model produce images that looks right to the eye, even if minor pixel differences remain [8]. To see definition of LPIPS metric, refer to Section 5.2.

2.4.3 SSIM-Based Structural Loss

The Structural Similarity Index (SSIM) measures how similar two images are by comparing small regions of each image:

$$\mathcal{L}_{SSIM} = 1 - SSIM(\hat{x}, x) \tag{37}$$

Minimizing this loss encourages the preservation of structural elements such as edges and contours – features that are especially important in tasks involving sharp transitions, object boundaries, and textures. In combination with the edge map input, this term enhances edge-aware reconstruction [27]. For the definition of SSIM metric used here, refer to Section 5.2.

2.4.4 Total Variation (TV) Loss

Total variation regularization penalizes rapid changes between adjacent pixels, promoting spatial smoothness and reducing noise:

$$\mathcal{L}_{TV} = \frac{1}{N}\sum_{i,j}\left(\left|\hat{x}_{i+1,j} - \hat{x}_{i,j}\right| + \left|\hat{x}_{i,j+1} - \hat{x}_{i,j}\right|\right) \tag{38}$$

where $\hat{x}$ – the reconstructed image, N – the total number of pixels per channel, $\hat{x}_{i,j}$ – the pixel intensity at position (i, j), $\hat{x}_{i+1,j} - \hat{x}_{i,j}$ – vertical pixel-wise difference, $\hat{x}_{i,j+1} - \hat{x}_{i,j}$ – horizontal pixel-wise difference.

This component is important for suppressing artifacts and encouraging locally consistent textures. It regularizes the output by discouraging high-frequency variations that are not present in natural images [28].

2.4.5 Latent Consistency Loss

To promote semantic alignment in the latent space, we use a latent consistency loss that enforces similarity between the representations of the ground-truth image and its reconstructed version.

Let:

- $x \in \mathbb{R}^{3\times H\times W}$ be the ground-truth clean image,
- $\tilde{x} \in \mathbb{R}^{4\times H\times W}$ be the degraded input (a 3-channel blurred image concatenated with a 1-channel edge map),
- $E_{gt}(\cdot)$ denote the latent encoder (Encoder),
- $E_{blur}(\cdot)$ denote the DeblurEncoder.

Then, the latent representations are computed as follows:

$$z_{gt} = E_{gt}(x), \quad z_{deblur} = E_{blur}(\tilde{x}) \tag{39}$$

The latent consistency loss is defined as the squared L2 distance between these two vectors:

$$\mathcal{L}_{latent} = \left\| z_{deblur} - z_{gt} \right\|_2^2 \tag{40}$$

This term ensures that the latent code extracted from a degraded input is not arbitrarily different from that of its corresponding clean target. In practice, this encourages the deblurring network to produce reconstructions that are not only visually similar but also semantically aligned with the original content at a feature level [23] [24].

Latent-level supervision of this kind has been shown to improve training stability and reconstruction fidelity in adversarial representation learning frameworks such as Adversarial Feature Learning [23] and Adversarially Learned Inference [24], and here it plays a similar role in regularizing the shared representation space.

2.4.6 Total Objective

The final loss is a weighted sum of the five components described above:

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_{L1} + \lambda_2\mathcal{L}_{LPIPS} + \lambda_3\mathcal{L}_{SSIM} + \lambda_4\mathcal{L}_{TV} + \lambda_5\mathcal{L}_{latent} \tag{41}$$

In our implementation, the coefficients were empirically selected as follows:

$$\lambda_1 = 0.5, \quad \lambda_2 = 0.3, \quad \lambda_3 = 0.15, \quad \lambda_4 = 0.05, \quad \lambda_5 = 0.2$$

This combination ensures a balance between pixel accuracy, perceptual realism, edge preservation, smoothness, and latent semantic consistency – all of which are crucial for high-quality photo-realistic image restoration.

2.5 Optimization Strategy

Training is performed using the Adam optimization algorithm [5], initialized with a learning rate of $\eta = 2 \times 10^{-4}$. All model components are optimized jointly by aggregating their parameters into a single update step.

To improve convergence stability and reduce the risk of overfitting, we employ a learning rate scheduler, that halves the learning rate every 15 epochs.

$$\eta_{t+1} = \gamma * \eta_t, \text{ with } \gamma = 0.5 \text{ every 15 epochs} \tag{42}$$

where η_t is the learning rate at epoch t, γ – the decay factor.

To preserve the best-performing model, we implement loss-based checkpointing: the model's parameters are saved whenever a new minimum in average epoch loss is reached. This ensures that the most performant version of the network is retained for evaluation and further experimentation.

This strategy combines adaptive optimization, progressive learning rate decay, and performance-based model selection to achieve robust convergence during training.

3. EXPERIMENTAL EVALUATION AND RESULTS

3.1 Evaluation Setup

The performance of the proposed model was evaluated on a test subset of the CelebABlur dataset. During inference, the model processes blurred RGB images concatenated with their corresponding edge maps, producing a reconstructed RGB output. Evaluation was conducted in a no-gradient mode using PyTorch's torch.no_grad() context to ensure deterministic and efficient computations.

Each prediction was compared against the ground-truth (clean) image using a suite of standard image restoration metrics:

- Peak Signal-to-Noise Ratio (PSNR), measuring signal fidelity in decibels;
- Structural Similarity Index Measure (SSIM), evaluating perceived structural similarity;
- Mean Absolute Error (L1 Loss);
- Mean Squared Error (MSE);
- Learned Perceptual Image Patch Similarity (LPIPS), computed using the pretrained 'alex' backbone.

PSNR and SSIM scores were calculated per sample using NumPy representations of the denormalized images, while L1, MSE, and LPIPS were averaged over mini-batches using their corresponding PyTorch implementations. The evaluation loop aggregates metric values and reports the final mean across all test samples.

To complement numerical evaluation, qualitative results were visualized for the first 100 samples in the test set. For each image, four stages were presented: the blurred input, the corresponding edge map, the reconstructed image, and the original ground-truth image — providing visual evidence of reconstruction fidelity.

3.2 Evaluation Metrics

To evaluate the performance of the implemented model, we used a set of quality metrics, that capture both pixel-level fidelity and perceptual similarity between the reconstructed image $\hat{x}$ and the ground-truth image x.

3.2.1 Mean Absolute Error (L1)

The L1 distance between the reconstructed and the ground truth is computed as:

$$L1 = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i| \tag{43}$$

It measures the mean absolute difference between corresponding pixels, treating all deviations equally and exhibiting robustness to outliers [25].

3.2.2 Mean Squared Error (MSE)

MSE:

$$\mathrm{MSE}(x, \hat{x}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{44}$$

It emphasizes larger errors more strongly than L1 [29].

3.2.3 Peak Signal-to-Noise Ratio (PSNR)

PSNR is defined as:

$$\mathrm{PSNR}(x, \hat{x}) = 10 \cdot \log_{10}\left(\frac{L^2}{\mathrm{MSE}(x,\hat{x})}\right) \tag{45}$$

where L is the maximum possible pixel value.

Higher PSNR indicates better reconstruction fidelity [30].

3.2.4 Structural Similarity Index Measure (SSIM)

SSIM measures how structurally similar two images are by comparing their luminance, contrast, and spatial structure.

The standard formulation is:

$$SSIM(\hat{x}, x) = \frac{(2\mu_x\mu_{\hat{x}}+C_1)(2\sigma_{x\hat{x}}+C_2)}{(\mu_x^2+\mu_{\hat{x}}^2+C_1)(\sigma_x^2+\sigma_{\hat{x}}^2+C_2)} \quad (46)$$

where $\mu,\ \sigma^2,\ \sigma_{x\hat{x}}$ denote the local means, variances, and cross-covariance, and C_1, C_2 are constants to stabilize division. SSIM ranges from -1 to 1, with higher values indicating better structural alignment [27].

3.2.5 Learned Perceptual Image Patch Similarity (LPIPS)

LPIPS measures perceptual distance using deep features from pretrained neural networks. It is computed as:

$$LPIPS(\hat{x}, x) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\phi_l(x)_{h,w} - \phi_l(\hat{x})_{h,w} \right) \right\|_2^2 \quad (47)$$

where $\phi_l(\cdot)$ denotes the activation at layer l, w_l are learned channel-wise weights, and (h, w) iterates over spatial dimensions. Lower LPIPS values indicate better perceptual quality [8].

3.3 Evaluation Results

The evaluation is conducted on the test set, and the average scores for matrices PSNR, SSIM, L1, MSE, and LPIPS are summarized in Table 1.

Metric	Value
PSNR	[23.53 - 23.68]
SSIM	[0.6927 – 0.6980]
L1	[0.0891 – 0.905]
MSE	[0.0192 – 0.0198]
LPIPS	[0.0788 – 0.0808]

Table 1: Average Performance Metrics on the Test Set

Interpretation of Metrics:

- PSNR measures pixel-level fidelity. A score between [23.53 – 23.68] indicates reasonable restoration quality, though not extremely high due to the complexity of face structures and motion blur artifacts.
- SSIM (Structural Similarity Index) focuses on structural information such as edges and textures. The score in the range [0.6927 – 0.6980] confirms moderate structural preservation, enhanced by the inclusion of edge maps.
- L1 and MSE are standard pixel-wise distance metrics. Their values ranges ([0.0192 – 0.0198] and [0.0192 – 0.0198], respectively) indicate consistent pixel-level closeness to the ground truth.
- LPIPS evaluates perceptual similarity using deep features. The result [0.0788, 0.0808] reflects visually plausible reconstructions that align well with human perception, even when pixel-wise metrics are suboptimal.

Summary:

The combination of moderate PSNR and SSIM with low LPIPS suggests that the model achieves a balanced trade-off between fidelity and perceptual realism. This validates the multi-objective training strategy, including the use of edge maps and latent supervision, as effective in recovering both detail and structure in blurred facial images.

3.4 Qualitative Results

Figure 5 presents visual examples of the model's performance on the CelebABlur dataset. Each column corresponds to a different image, and each row contains a distinct representation:

1. Top row: Blurred input image.
2. Second row: Extracted edge map.

3. Third row: Reconstructed output from the model.
4. Bottom row: Ground-truth high-quality image.

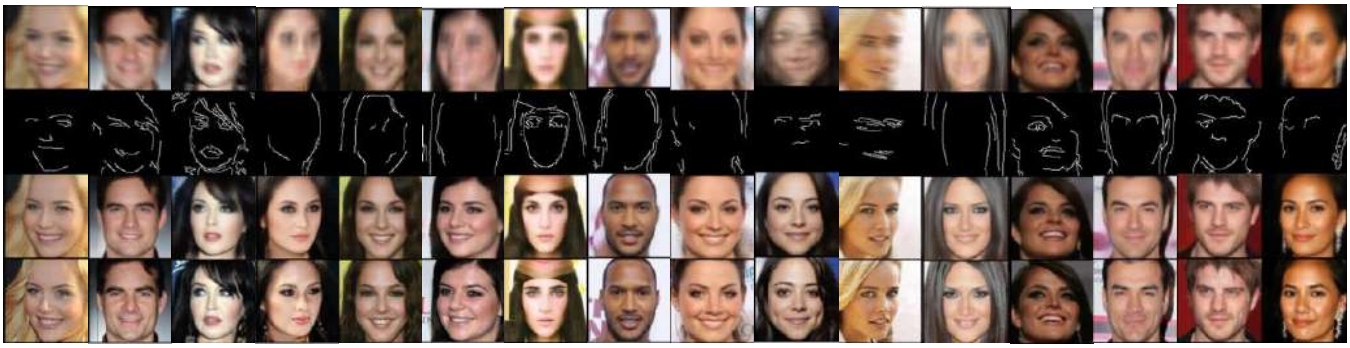

Figure 5: Qualitative results of developed model on Image Deblurring

Observations:

- Facial reconstruction quality: The model demonstrates a strong ability to restore facial features such as eyes, mouth, and facial outlines. In all examples, the restored faces closely resemble the ground truth, confirming that the architecture effectively leverages the latent representation and edge information.
- Edge map contribution: The presence of clear contours in the edge map enables the model to maintain structural alignment and sharp transitions in the reconstructed output — particularly around the jawline, ears, and hairline.
- Limitations: The model shows a tendency to underperform when background elements or complex textures are present.

3.5 Advantages of the work

The proposed image restoration framework introduces several advantages over conventional and baseline approaches:

- Integration of Structural Guidance via Edge Maps.
  By incorporating an edge map alongside the blurred image as input, the model benefits from explicit structural cues. This guides the network to focus on

transitions and object boundaries, significantly improving the fidelity of restored edges. This idea is particularly beneficial in reconstructing fine facial details that are commonly lost in blurred inputs.

- Latent Space Consistency for Semantic Alignment
  The introduction of latent-space supervision via a dedicated encoder enables semantic alignment between restored and ground-truth images. This approach reduces perceptual artifacts and ensures that reconstructions are not only visually plausible but also structurally coherent at a deeper feature level.
- Joint Training of All Components
  Unlike modular or multi-stage pipelines, the architecture is trained end-to-end. This allows gradients to flow across the DeblurEncoder, Generator, and Latent Encoder simultaneously, resulting in better convergence and more harmonized feature learning across modules.
- Use of Composite Loss Function
  The training objective integrates multiple loss functions. This composite formulation balances low-level pixel fidelity, perceptual realism, structural accuracy, and latent representation alignment, thereby addressing multiple aspects of restoration quality in a single framework.

CONCLUSIONS

- In this work, several existing image restoration methods were reviewed and analyzed. These include Image restoration with StyleGAN [2], where the image is not restored directly but through a search in the generator's latent space. The method searches for a latent code z that allows the generator to produce an image closely matching the degraded input. Posterior-Mean Rectified Flow [3], that operates as reverse modeling of the distribution: it reconstructs a high-quality image starting from random noise, moving backward along a trajectory that minimizes the MSE. DiffIR: Efficient Diffusion Model for Image Restoration [4] – a diffusion-based model that gradually adds noise to a prior representation and then learns to recover it. This enables structurally accurate reconstructions with fewer iterations compared to traditional diffusion models.
- A novel image restoration architecture was proposed and implemented, targeting the reconstruction of high-quality facial images from blurred inputs. The algorithm comprises three primary components: DeblurEncoder, Generator, and Latent Encoder, which are trained jointly from scratch. One of the key innovations of this work is the incorporation of edge maps as an additional input channel, allowing the network to focus on sharp transitions and critical boundaries, thereby improving restoration quality.
- To train model we used a composite loss, that is a weighted sum of the five components: L1 pixel loss, LPIPS perceptual similarity, SSIM structural similarity, total variation regularization, and a latent consistency loss to align semantic features between the reconstructed and ground truth images. The training process included dynamic learning rate scheduling and model checkpointing based on validation performance.

- Quantitative results demonstrate that the proposed system achieves promising results. Qualitative inspection further supports the model's ability to produce visually coherent and structurally plausible reconstructions. However, certain limitations remain in restoring non-facial background elements, especially in cases with high scene complexity.
- Overall, this work contributes a lightweight and interpretable framework for edge-aware image restoration. The methodology is extensible and may serve as a foundation for future research involving extending the model to handle color artifacts and adapting it to other types of image degradation.

REFERENCES

[1] J. Canny, “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[2] Poirier-Ginter, Y., & Lalonde, J.-F. (2023). *Robust Unsupervised StyleGAN Image Restoration*. arXiv preprint arXiv:2302.06733. https://doi.org/10.48550/arXiv.2302.06733

[3] Ohayon, G., Michaeli, T., & Elad, M. (2024). *Posterior-Mean Rectified Flow: Towards Minimum MSE Photo-Realistic Image Restoration*. arXiv preprint arXiv:2410.00418. https://doi.org/10.48550/arXiv.2410.00418

[4] Xia, B., Zhang, Y., Wang, S., Wang, Y., Wu, X., Tian, Y., Yang, W., & Van Gool, L. (2023). *DiffIR: Efficient Diffusion Model for Image Restoration*. arXiv preprint arXiv:2303.09472. https://doi.org/10.48550/arXiv.2303.09472

[5] Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

[6] Abdal, R., Qin, Y., & Wonka, P. (2019). *Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?* arXiv preprint arXiv:1904.03189. https://doi.org/10.48550/arXiv.1904.03189

[7] Watt, J., Borhani, R., & Katsaggelos, A. K. (2016). *Machine learning refined: Foundations, algorithms, and applications*. Cambridge University Press. https://doi.org/10.1017/CBO9781316402276

[8] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” *CVPR*, 2018.

[9] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[10] Dror Freirich, Tomer Michaeli, and Ron Meir. A theory of the distortion-perception tradeoff in wasserstein space. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 25661–25672. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d77e68596c15c53c2a33ad143739902d-Paper.pdf.

[11] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[12] Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising diffusion probabilistic models*. Advances in Neural Information Processing Systems (NeurIPS), 33, 6840–6851. https://papers.nips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., & others. (2021). *An image is worth 16×16 words: Transformers for image recognition at scale*. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2010.11929

[14] Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021). *Pre-trained Image Processing Transformer*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12299–12310). https://doi.org/10.1109/CVPR46437.2021.01211

[15] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). *High-Resolution Image Synthesis with Latent Diffusion Models.* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:2112.10752. https://doi.org/10.48550/arXiv.2112.10752

[16] Patro, S. G. K., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*. https://doi.org/10.48550/arXiv.1503.06462

[17] Radford, A., Metz, L., & Chintala, S. (2016). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. arXiv preprint arXiv:1511.06434.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[19] Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully convolutional networks for semantic segmentation*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3431–3440). https://doi.org/10.1109/CVPR.2015.7298965

[20] Blumberg, S. B., Raví, D., Xu, M.-C., Figini, M., Kokkinos, I., & Alexander, D. C. (2022). *Deformably-Scaled Transposed Convolution*. arXiv preprint arXiv:2210.09446. https://doi.org/10.48550/arXiv.2210.09446

[21] Dumoulin, V., & Visin, F. (2016). *A guide to convolution arithmetic for deep learning*. arXiv:1603.07285. https://doi.org/10.48550/arXiv.1603.07285

[22] Xu, B., Wang, N., Chen, T., & Li, M. (2015). *Empirical Evaluation of Rectified Activations in Convolutional Network*. arXiv preprint arXiv:1505.00853. https://doi.org/10.48550/arXiv.1505.00853

[23] T. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial Feature Learning," *arXiv preprint arXiv:1605.09782*, 2016.

[24] V. Dumoulin, I. Belghazi, B. Poole, et al., "Adversarially Learned Inference," *arXiv preprint arXiv:1606.00704*, 2016.

[25] He, X., & Cheng, J. (2022). *Revisiting L1 Loss in Super-Resolution: A Probabilistic View and Beyond*. arXiv preprint arXiv:2201.10084.

[26] Wang, C., Li, S., He, D., & Wang, L. (2022). *Is L2 Physics-Informed Loss Always Suitable for Training Physics-Informed Neural Network?* arXiv preprint arXiv:2206.02016.

[27] Nilsson, J., & Akenine-Möller, T. (2020). *Understanding SSIM*. arXiv preprint arXiv:2006.13846.

[28] Estrela, V. V., Magalhaes, H. A., & Saotome, O. (2016). *Total Variation Applications in Computer Vision*. In N. K. Kamila (Ed.), *Handbook of Research on Emerging Perspectives in Intelligent Pattern Recognition, Analysis, and Image Processing* (pp. 24–47). IGI Global. https://doi.org/10.4018/978-1-4666-8654-0.ch002. Also available as arXiv preprint arXiv:1603.09599.

[29] Hodson, T. O., Over, T. M., & Foks, S. S. (2021)**.** Mean squared error, deconstructed. *Journal of Advances in Modeling Earth Systems, 13*(10), e2021MS00268. https://doi.org/10.1029/2021MS00268

[30] Huynh-Thu, Q., & Ghanbari, M. (2008). *Scope of validity of PSNR in image/video quality assessment*. Electronics letters, 44(13), 800-801. DOI: 10.1049/el:20080522