INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Medical Image Super-Resolution via Diffusion Probabilistic Models for Reducing False Negatives in Classification

Oleksandr Novytskyi

Master in Computer Engineering

Supervisor:
PhD João Pedro Afonso Oliveira da Silva, Associate Professor,
Iscte – Instituto Universitário de Lisboa

October, 2025

TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Medical Image Super-Resolution via Diffusion Probabilistic Models for Reducing False Negatives in Classification

Oleksandr Novytskyi

Master in Computer Engineering

Supervisor:
PhD João Pedro Afonso Oliveira da Silva, Associate Professor,
Iscte – Instituto Universitário de Lisboa

October, 2025

Acknowledgment

I like to convey my profound appreciation to those persons whose unwavering support, direction, and inspiration were crucial to the accomplishment of this Master's thesis.

I express my sincere gratitude to my supervisor, Doctor João Pedro Oliveira, Associate Professor, for his outstanding academic advice, inspiring encouragement, and incredible patience during this research. His incisive thoughts and steady backing were pivotal in advancing our project from conception to fruition.

I would like to extend my gratitude to the Instituto de Telecomunicações (IT) and Iscte - Instituto Universitário de Lisboa, for their support and for providing the computational and infrastructural conditions for the development of this thesis.

I would like to recognize the incredible dedication of Dr. Vitor Moura Gonçalves, Neurosurgeon, and Dr. Ivan Protsenko, Family Physician. Their involvement, professionalism, and assistance rendered the achievement of this undertaking feasible.

I am profoundly grateful to my family. To my children, I express my gratitude for your faith in me and your constant support; you are an enduring source of inspiration. I extend my deepest gratitude to my wife, whose tireless encouragement and exceptional support surmounted every obstacle, furnishing the emotional resilience necessary to attain and complete this degree. Her sacrifice and encouragement were the main factors that made it possible to complete this work.

This thesis represents the culmination of my Master's studies; yet, I truly aspire for it to be merely a major milestone in my ongoing academic path.

Resumo

A super-resolução em imagens médicas é uma técnica crucial destinada a melhorar a resolução espacial e a qualidade da imagem, o que pode impactar significativamente o desempenho de sistemas automatizados de diagnóstico. Este estudo investiga a eficácia de um modelo probabilístico de difusão guiada para aumentar a resolução de imagens médicas, com o objetivo de reduzir a classificação de falsos negativos em análises baseadas em redes neurais.

Foi implementada uma estrutura experimental abrangente, utilizando um conjunto de dados público de imagens de Resonancia Magrnetica cerebrais, um classificador Residual Net- work (ResNet18) otimizado e um modelo de difusão guiada personalizado para super-resolução. O modelo foi concebido para gerar imagens de alta resolução a partir de entradas de baixa resolução, preservando características relevantes para o diagnóstico.

A avaliação quantitativa demonstra que o modelo de super-resolução baseado em difusão guiada melhora substancialmente a qualidade das imagens reconstruídas, resultando numa redução das previsões de falsos negativos pelo classificador. Esta abordagem fornece informação nova e de alta fidelidade que aumenta a capacidade do classificador em detetar padrões patológicos subtis que poderiam perder-se em dados de baixa resolução.

O estudo evidencia o potencial da super-resolução baseada em difusão para apoiar uma análise de imagens médicas precisa e fiável, sendo a implementação deste trabalho disponibilizada publicamente no GitHub: `https://github.com/artofnext/master_thesis`.

Palavras Chave: Super-Resolução; Modelos de Difusão; Imagiologia Médica; Melhoria de Imagem; Ressonância Magnética Cerebra; Diagnóstico Assistido por Computador.

Abstract

Super Resolution (SR) in medical imaging is a technique aimed at improving spatial resolution and image quality, which can significantly impact the performance of automated diagnostic systems. This study investigates the effectiveness of a guided diffusion probabilistic model for enhancing medical image resolution to reduce False Negatives (FN) classification in neural network-based analysis.

A comprehensive experimental framework was employed, utilizing a publicly available dataset of brain Magnetic Resonance Imaging (MRI) scans, an optimized Residual Network (ResNet)18 classifier, and a custom-trained guided diffusion model for SR. The model is designed to generate high-resolution images from low-resolution inputs while preserving diagnostically relevant features.

Quantitative evaluation demonstrates that the guided diffusion SR model substantially improves the quality of reconstructed images, resulting in a reduction of false-negative predictions by the classifier. The approach provides novel, high-fidelity image information that enhances the classifier's ability to detect subtle pathological patterns that may be lost in low-resolution data.

The study highlights the potential of diffusion-based SR to support accurate and reliable medical image analysis, and the implementation of this work is publicly available on GitHub: `https://github.com/artofnext/master_thesis`.

KEYWORDS: Super-Resolution; Diffusion Models; Medical Imaging; Image Enhancement; Brain MRI; Computer-Aided Diagnosis.

Contents

List of Figures

List of Tables

List of Acronyms

AI: Artificial Intelligence

CNN: Convolutional Neural Network

CT: Computed Tomography

CUDA: Compute Unified Device Architecture

DDPM: Denoising Diffusion Probabilistic Model

DL: Deep Learning

DM: Diffusion Model

ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks

FID: Fréchet Inception Distance

FNR: False Negative Rate

FN: False Negatives

FiLM: Feature-wise Linear Modulation

GAN: Generative Adversarial Network

GB: Gigabyte

Glow: Generative Flow with Invertible 1x1 Convolutions

GPU: Graphics Processing Unit

HR: High Resolution

IPT: Image Processing Transformer

LPIPS: Learned Perceptual Image Patch Similarity

LR: Low Resolution

MAE: Mean Absolute Error

ML: Machine Learning

MRI: Magnetic Resonance Imaging

MSE: Mean Squared Error

OOM: Out-of-memory

PIL: Python Imaging Library

PixelCNN: Pixel Convolutional Neural Networks

PixelRNN: Pixel Recurrent Neural Networks

PSNR: Peak Signal-to-Noise Ratio

RAM: Random Access Memory

RealNVP: Real-Valued Non-Volume Preserving transformations

ReLU: Rectified Linear Unit

ResNet: Residual Network

SRCNN: Super-Resolution Convolutional Neural Network

SRGAN: Super-Resolution Generative Adversarial Network

SR: Super Resolution

SSIM: Structural Similarity Index

SwinIR: Shifted window transformer-based Image Restoration

VAE: Variational Autoencoder

VGG: Visual Geometry Group

CHAPTER 1

Introduction

This chapter is intended to reveal the motivation of the dissertation, formulate the research questions, outline the main objectives, and present the structure of the thesis.

1.1. Motivation

Quality medical images play a key role in the diagnosis and treatment process. They make it possible to detect the disease in the early stages, significantly improving the prognosis of the patient's treatment. Recently, medical image Super Resolution (SR) is an important image processing technology designed to improve spatial resolution and overall image quality. Low Resolution (LR) images often lack the complex detail necessary to recognize small lesions, early-stage malignancies, microcalcifications, or minor structural abnormalities that may suggest disease. This affects the accuracy of diagnosis, especially important in the early stages of the disease when a False Negatives (FN) in diagnosis can prevent timely treatment and worsen the prognosis for recovery. SR can transform LR images into High Resolution (HR) counterparts, revealing intricate features. By enhancing the visibility of small abnormalities, it directly tackles a prevalent source of false negatives - the difficulty in visually identifying the pathology. A novel integration of Diffusion Model (DM) for the SR task is proposed in this study. The purpose of this work is to directly address the challenge of subtle pathologies that can lead to FN. The performance of DMs in image synthesis has recently been shown to be state-of-the-art [45]. These models excel at synthesizing photorealistic and high-fidelity details, which are essential for distinguishing subtle medical traits. The primary objective is to train a diffusion-based SR model to not only upsample the LR input into a HR image, but also to steer this generative process by making use of image class information (for example, "unhealthy" or "healthy").

Under the influence of this class-conditional generation, the SR model is compelled to give priority to the synthesis of characteristics that are pertinent to the putative diagnostic label of the image. For example, in an LR image that has been identified as exhibiting modest indicators of disease, the class-guided diffusion process will be explicitly conditioned to yield an HR image in which the subtle pathological features that are nonetheless crucial are maximally resolved and distinguishable. After that, the synthetic SR pictures that were produced are sent into a separate image classification model that is located farther downstream. The overall classification performance is anticipated to witness a significant improvement as a result of the provision of these improved HR images to the classifier. This improvement is assumed to be particularly noticeable in metrics that

pertain to classification safety. Specifically, it is hypothesized that the method will improve accuracy and precision, and more importantly, it will minimize the False Negative Rate (FNR). The idea behind this that it is accomplished by making microcalcifications, microscopic lesions, or minor structural anomalies that were previously disregarded plainly visible, and therefore accurately identifiable by the classifier. By utilizing the generative abilities of diffusion models for direct clinical benefit, this integrated approach means that the resultant generated SR pictures can also improve the possibility for accurate automatic diagnosis. This is accomplished by exploiting the generative abilities of DM.

1.2. Research questions

This work aims to answer the following research questions:

(1) Can a guided diffusion-based SR model generate HR images from LR inputs while preserving visually meaningful structural information relevant for downstream classification tasks?
(2) How does the use of generated super-resolved images influence the performance of a machine learning classification model, in terms of accuracy, precision and false negative rate, compared to classification using the original HR images under identical evaluation conditions?
(3) Are the observed differences in classification performance between super-resolved images and baseline inputs statistically significant when evaluated on the same test samples?
(4) What are the main advantages, limitations, and potential sources of bias associated with using a guided diffusion-based SR model as a preprocessing step for image classification, and what steps are required to further evaluate and improve such an approach?

1.3. Objectives

The primary objective of this study is to investigate capability of a class-guided DM SR model in enhancing medical image quality to minimize the FN rate of subsequent automated neural network classification. The process of solving the problem includes the following steps:

(1) Establish a guided DM for medical images SR: to design, implement, and train a DM that is capable of performing SR on LR medical images, while also critically incorporating class conditioning to guide the generation process based on the diagnostic label of the image ("healthy" or "unhealthy").
(2) Effectiveness of the class-guided DM SR process for reduced FN: to validate that the class-guided DM model accurately generates HR medical images that significantly resolve subtle or small pathological features that were blurred in the original LR inputs, directly addressing the difficulty in visually identifying pathology, which is a key source of false negatives.

(3) In order to improve the performance of downstream classification, it is necessary to incorporate the SR images that were generated into a conventional image classification model and to demonstrate a measurable improvement in key classification metrics, specifically:
   - The accuracy and precision of classification should be improved.
   - Classification safety can be improved by minimizing FNR, and the prognosis for early-stage disease identification can be improved as well.

1.4. Ethical and Regulatory Considerations

Medical image analysis systems function in a safety-critical arena, where algorithmic results might impact diagnostic reasoning and subsequent therapeutic judgments. The utilization of generative models for medical picture SR presents ethical and regulatory challenges that surpass those associated with traditional image processing tasks.

Diffusion-based SR techniques can alter image content in nuanced manners that may be indistinguishable from authentic anatomical structures. Although these alterations may enhance downstream machine-learning performance, they also provoke issues about interpretability, dependability, and the potential introduction of artificial features that correlate with disease labels instead of the underlying pathology. These dangers necessitate a prudent and transparent assessment of any claimed performance improvements.

This work is exclusively a methodological and experimental investigation. The produced super-resolved images are assessed solely inside a regulated computational environment and are not designed for clinical use or diagnostic determinations. All studies are performed on a completely anonymized dataset, with ethical considerations prioritizing statistical validation, reproducibility, and critical interpretation over absolute performance optimization.

1.5. Contribution

This work makes three primary contributions, which demonstrate the adaption and analysis of generative DM in medical imaging. These contributions are as follows:

(1) An application and modification of a class-conditional guided DM SR framework: the successful implementation and adaptation of a state-of-the-art DM for medical imaging SR. The primary innovation is in the methodical use of diagnostic class conditioning to direct the process of backward diffusion. This ensures that the HR output that is created enhances the clarity of minor traits that are significant.

(2) Demonstration of FN reduction through empirical evidence. This work offers empirical evidence that demonstrates the usefulness of a guided DM SR technique in addressing an important safety parameter in medical images classification. The research exhibits a measurable reduction in the FNR of a standard downstream classification system, which directly supports increased early disease identification. This reduction is achieved by focusing the SR process on ambiguous images.

(3) Establishment and evaluation of a dependable approach for integrating the DM SR phase as an efficient feature-enhancement pre-processing step for conventional discriminative classifiers is the objective of the evaluation of practical integration into automated diagnosis systems. The purpose of this work is to demonstrate the practical performance advantages and greater robustness that may be gained by feeding automated diagnosis pipelines with high-fidelity images that have been enhanced with DM.

1.6. Theoretical background

The topic of SR employing DM is one of the most active areas in the field of generative Artificial Intelligence (AI), and the state-of-the-art is fast evolving, particularly for specific domains such as medical imaging [60] [62] [35]. DM SR approaches are used to solve ill-posed problems by modeling the reconstruction as a stochastic, iterative denoising process. This process begins with pure noise and is guided by the LR input. The goal of this process is to generate a photorealistic HR output while effectively capturing the complex distribution of high-frequency details. This section provides an overview of the theoretical foundations and current state-of-the-art.

1.6.1. Super Resolution problem definition

The problem of SR [33] in medical imaging refers to the task of reconstructing a HR image from a given LR image [56]. Such LR images are typically obtained through various acquisition modalities including Magnetic Resonance Imagin (MRI), Computed Tomography (CT), X-ray, or ultrasound. Due to physical, technical, or dose-related constraints, these imaging modalities may not always achieve high spatial resolution. Consequently, there arises a need for computational approaches that can enhance the spatial resolution and improve the diagnostic quality of medical images.

Formally, let us denote the LR image in digital format as

$$x \in \mathbb{R}^{w' \times h' \times c},$$

where w' and h' represent the width and height of the LR image respectively, and c corresponds to the number of channels (for example, $c = 1$ in grayscale medical scans such as MRI, and $c = 3$ for colored modalities). The objective of SR is to estimate its corresponding HR image

$$y \in \mathbb{R}^{w \times h \times c},$$

where $w > w'$ and $h > h'$, i.e., the target image possesses higher spatial resolution.

The degradation process from y to x can be mathematically expressed as [33]:

$$x = \mathcal{D}(y; \delta) = (y * k) \downarrow_s + n \tag{1.1}$$

Here, $\mathcal{D}$ represents a general degradation function parameterized by δ. The degradation pipeline involves several steps: a blurring operation as a convolution using a kernel k, a subsequent downsampling (denoted by $\downarrow_s$) with scaling factor s, and the addition of

noise n. The scaling factor is formally defined as $s \in \mathbb{N}$, such that $h = s \cdot h'$ and $w = s \cdot w'$. This degradation model reflects realistic conditions in medical imaging where hardware and acquisition protocols limit the achievable resolution, and noise or blur is often introduced due to motion, acquisition artifacts, or system imperfections.

The central aim of SR is to approximate the inverse mapping of $\mathcal{D}$, which is generally non-unique and ill-posed. This ill-posedness stems from the fact that multiple plausible HR images can correspond to the same LR input, making the reconstruction problem underdetermined. To address this challenge, Machine Learning (ML) models attempt to learn a mapping function $\mathcal{M}$ that approximates the inverse transformation [33]:

$$\hat{y} = \mathcal{M}(x; \gamma), \tag{1.2}$$

where γ denotes the parameters of the model, typically corresponding to the weights of a neural network. The goal of the model is to produce an estimate $\hat{y}$ that is perceptually and quantitatively close to the ground truth HR image y.

From an optimization perspective, this problem is framed as the minimization of a loss function $\mathcal{L}$ that measures the discrepancy between the estimated $\hat{y}$ and the reference y. The optimization problem can be formulated as [33]:

$$\hat{\gamma} = \arg\min_{\gamma} \mathcal{L}(\hat{y}, y). \tag{1.3}$$

It is important to note the connection between Equations 1.2 and 1.3. Since the reconstructed image $\hat{y}$ is itself a function of the model parameters γ through the mapping $\hat{y} = \mathcal{M}(x; \gamma)$, the optimization problem in Equation 1.3 can be interpreted as adjusting γ such that the output of the model best approximates the ground truth y. In other words, the loss function $\mathcal{L}(\hat{y}, y)$ is indirectly a function of γ, because $\hat{y}$ depends on γ. Thus, minimizing $\mathcal{L}$ with respect to γ corresponds to training the model parameters so that the generated SR image $\hat{y}$ becomes as close as possible to the true HR image y.

Loss Functions in SR. The choice of the loss function $\mathcal{L}$ is crucial, as it defines the optimization objective and significantly influences the visual and diagnostic quality of the reconstructed images. In SR, several categories of loss functions [22, 45] are commonly used:

- Pixel-wise losses: The most fundamental loss functions are based on direct pixel-level comparison[29]. The Mean Squared Error (MSE) is defined as:

$$\mathcal{L}_{\text{MSE}}(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2, \tag{1.4}$$

  where N is the number of pixels in the image. This loss correlates with maximizing Peak Signal-to-Noise Ratio (PSNR) but often results in overly smooth images.

The Mean Absolute Error (MAE) or L1 loss [59] is defined as:

$$\mathcal{L}_{\text{MAE}}(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \,, \tag{1.5}$$

which is less sensitive to outliers and typically produces sharper reconstructions compared to MSE.

- Structural similarity losses: Instead of comparing pixels directly, the Structural Similarity Index (SSIM) considers local luminance, contrast, and structural information [59]:

$$\mathcal{L}_{\text{SSIM}}(\hat{y}, y) = 1 - \text{SSIM}(\hat{y}, y), \tag{1.6}$$

where $\text{SSIM}(\cdot)$ is a perceptual metric ranging between -1 and 1, with higher values indicating better structural similarity.

- Perceptual losses: These losses compare deep feature representations extracted from a pretrained convolutional network [59]. Let $\phi_l(\cdot)$ denote the activation at layer l:

$$\mathcal{L}_{\text{perc}}(\hat{y}, y) = \frac{1}{C_l H_l W_l} \left\| \phi_l(\hat{y}) - \phi_l(y) \right\|_2^2 , \tag{1.7}$$

where C_l, H_l, W_l are the dimensions of the feature map. This loss ensures perceptual fidelity beyond pixel matching.

In practice, modern SR systems employ a weighted combination of these losses [29, 59]:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{pixel}} + \lambda_2 \mathcal{L}_{\text{perc}} + \lambda_3 \mathcal{L}_{\text{SSIM}},$$

where λ_i are hyperparameters controlling the contribution of each loss term. In medical imaging, this combination is carefully designed to preserve fine anatomical details and avoid generating artifacts or "hallucinated" structures that could mislead clinical interpretation.

1.6.2. Diffusion-based Super Resolution Model

Diffusion models represent a class of generative models that have recently demonstrated state-of-the-art performance in image synthesis and restoration tasks, including SR [49, 45, 19, 40, 60]. The fundamental concept of the diffusion model is derived from non-equilibrium statistical physics, where it can be interpreted as a gradual transformation of structured data into noise, followed by a learned reversal of this process. More concretely, the model consists of two complementary phases: a forward diffusion process, in which data is progressively degraded by the addition of Gaussian noise, and a backward diffusion process, in which this noise is iteratively removed to recover the original signal. Together, these processes form a highly flexible and expressive probabilistic generative framework [49].

1.6.3. Forward Diffusion Process

The forward diffusion process is defined as a fixed Markov chain that incrementally perturbs a data sample x_0 over a sequence of T timesteps [19, 45]. At each step t, Gaussian noise is added according to a predefined variance schedule $\{\beta_t\}_{t=1}^T$, with $\beta_t \in (0,1)$ typically chosen to increase slowly with t. Formally, the transition probability between successive latent states x_{t-1} and x_t is given by:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I), \tag{1.8}$$

where q denotes the forward process, $\mathcal{N}$ is the multivariate Gaussian distribution, and I is the identity covariance matrix. The mean $\sqrt{1-\beta_t}\, x_{t-1}$ ensures that the new state x_t remains correlated with its predecessor, while $\beta_t I$ controls the variance of the injected noise [19].

An important property of this process is that a closed-form expression exists for sampling x_t directly from the original clean data x_0 without explicitly computing every intermediate step. This direct formulation is given as:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} \cdot x_0, (1-\bar{\alpha}_t) I), \tag{1.9}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. This equation highlights that x_t can be expressed as a convex combination of the clean image x_0 and Gaussian noise. As $t \to T$, $\bar{\alpha}_t \to 0$ and the distribution of x_T approaches an isotropic Gaussian, i.e., the original image becomes fully destroyed by noise.

Algorithm 1 represents the forward diffusion process that progressively adds Gaussian noise to a HR image x_0 over T steps according to a variance schedule $\{\beta_t\}_{t=1}^T$ [19]. After sufficient steps, the image approaches isotropic Gaussian noise.

Algorithm 1 Forward diffusion process

Require: Clean image x_0, noise schedule $\{\beta_t\}_{t=1}^T$, total steps T

1: **for** $t = 1$ to T **do**
2: Sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$
3: Compute $\alpha_t = 1 - \beta_t$
4: Compute cumulative product $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$
5: Generate noisy image:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$$

6: **end for**

Ensure: Sequence $\{x_t\}_{t=1}^T$ with $x_T \approx \mathcal{N}(0, I)$

1.6.4. Backward Diffusion Process

The backward process is also parameterized as a Markov chain that aims to invert the corruption introduced during forward diffusion [19, 45]. Unlike the forward process, which is fixed, the backward process is learned via a neural network with parameters θ. Starting from pure Gaussian noise $x_T \sim \mathcal{N}(0, I)$, the model sequentially denoises the input until

it reconstructs a clean data sample x_0. The joint distribution of the backward process is expressed as:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \tag{1.10}$$

where $p(x_T) = \mathcal{N}(0, I)$ serves as the prior distribution, and $p_\theta(x_{t-1}|x_t)$ defines the learned transition at each step. Each backward step is modeled as a Gaussian distribution with mean and variance predicted by the neural network:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{1.11}$$

where $\mu_\theta(x_t, t)$ denotes the predicted denoised mean at timestep t, and $\Sigma_\theta(x_t, t)$ (often fixed or simplified to $\beta_t I$) controls the variance. The learning objective for the model is to approximate the true backward process distribution $q(x_{t-1}|x_t, x_0)$ using $p_\theta(x_{t-1}|x_t)$.

The backward diffusion process shown in Algorithm 2 iteratively denoises a noisy image x_T using a trained model $\epsilon_\theta(x_t, t)$ to reconstruct the HR image [19].

Algorithm 2 Backward diffusion (reconstruction) process

Require: Noisy image $x_T \sim \mathcal{N}(0, I)$, trained model $\epsilon_\theta(x_t, t)$, noise schedule $\{\beta_t\}_{t=1}^{T}$

1: **for** $t = T$ down to 1 **do**

2: Compute $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

3: Predict noise with neural network:

$$\hat{\epsilon} = \epsilon_\theta(x_t, t)$$

4: Estimate clean image mean at step t:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\hat{\epsilon}\Big)$$

5: **if** $t > 1$ **then**

6: Sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$

7: Update:

$$x_{t-1} = \mu_\theta(x_t, t) + \sqrt{\beta_t}\,\epsilon$$

8: **else**

9: Set $x_{t-1} = \mu_\theta(x_t, t)$

10: **end if**

11: **end for**

Ensure: Reconstructed high-resolution image x_0

1.6.5. Training Objective and Reparameterization Trick

To train the diffusion model, the reverse distribution $p_\theta(x_{t-1}|x_t)$ is not learned directly. Instead, the problem is reformulated in terms of the noise ϵ introduced during the forward process. Using Equation (1.9), a noisy sample x_t can be expressed as:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{1.12}$$

This reformulation is known as the *reparameterization trick*, where the randomness is isolated in the variable ϵ and the transformation becomes deterministic with respect to x_0 and t. The key benefit of this trick is that it enables efficient gradient-based optimization, since expectations over random variables can be rewritten as expectations over deterministic functions of fixed noise samples. This approach, originally popularized in Variational Autoencoder (VAE) [25] [26] [14], is essential for stable and tractable training of diffusion models.

The neural network $\epsilon_\theta(x_t, t)$ is trained to predict the noise ϵ given x_t and timestep t. The training objective is therefore a simple denoising loss:

$$\mathcal{L}_{\text{DM}}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\, \|\epsilon - \epsilon_\theta(x_t, t)\|_2^2 \,\right]. \tag{1.13}$$

This objective encourages the model to learn the backward process by denoising step by step. In practice, it can be shown that minimizing this loss is equivalent to maximizing the likelihood of the data under the learned generative model. As a result, the reparameterization trick not only simplifies the training procedure but also provides a mathematically grounded way of connecting the forward and backward diffusion processes through noise prediction.

1.6.6. Application to Super Resolution

For SR tasks, diffusion models are conditioned on an observed LR image x_{LR}. The backward process is then guided not only by noise removal but also by ensuring consistency with the LR input [45, 19]. This conditioning can be implemented via concatenation, feature injection, or cross-attention mechanisms. Formally, the backward transition becomes:

$$p_\theta(x_{t-1}|x_t, x_{\text{LR}}) = \mathcal{N}(x_{t-1}, \mu_\theta(x_t, t, x_{\text{LR}}), \Sigma_\theta(x_t, t)).$$

By conditioning the denoising process on x_{LR}, the model generates a plausible HR reconstruction x_0 that aligns with the structural content of the input while restoring missing fine details.

1.7. Thesis structure

This thesis is organized into five chapters, each addressing a critical component of the research on medical image SR using diffusion probabilistic models.

Chapter 1, Introduction, establishes the motivation, research questions, objectives, and key contributions of the study. It provides the theoretical background necessary to understand the SR problem and outlines the potential of diffusion models for enhancing medical image quality and classification performance.

Chapter 2, Literature Review, presents a systematic review of existing work in the field of medical image SR. It discusses traditional non-generative and modern generative approaches, emphasizing diffusion-based models and their applications. The chapter concludes by identifying the research gap that this thesis aims to address.

Chapter 3, Methodology, describes the architecture and operational principles of the proposed guided diffusion model. It details the dataset preparation, model training procedures, hyperparameter optimization, and evaluation metrics. This chapter also outlines the experimental design used to assess model performance in downstream classification tasks.

Chapter 4, Experimental Results, reports the outcomes of the conducted experiments. It includes quantitative metrics, qualitative visual comparisons, and statistical analyses such as the McNemar test to evaluate the performance difference between classifications on ground-truth and SR-generated images. The chapter further discusses the advantages, limitations, and potential applications of the proposed approach.

Chapter 5, Conclusions and Future Work, summarizes the main findings, highlights the contributions and limitations of the study, and proposes directions for future research, including model generalization, computational optimization, and possible clinical applicability.

CHAPTER 2

Literature Review

This chapter provides a literature review and examines the development of SR methodologies within the broader context of image reconstruction and enhancement. The review traces the historical and conceptual evolution of SR techniques from classical interpolation-based methods to contemporary approaches founded on Deep Learning (DL). It distinguishes between non-generative methods, which employ deterministic reconstruction strategies aimed at minimizing pixel-level discrepancies, and generative methods, which model the inherent one-to-many mapping between low- and high-resolution domains.

The following subsections of the chapter highlight how non-generative models tend to preserve structural fidelity while often lacking realistic textural detail, whereas generative models, such as generative adversarial networks Generative Adversarial Networks (GANs) and DMs, introduce probabilistic frameworks that better capture complex image distributions. Building on this progression, the chapter outlines the principles of DM models and especially emphasizes their adaptation for medical image SR tasks. Finally, it identifies a gap in the existing literature regarding class-conditional guidance within diffusion-based SR frameworks, providing the conceptual basis for the methodological approach pursued in this thesis.

2.1. Systematic Literature Review

The literature review was conducted systematically to ensure comprehensive coverage of relevant scientific publications, beginning with targeted searches in academic databases followed by an analysis of references and citation chaining within the discovered works. Across the most important scientific indexing databases, the initial search was carried out

TABLE 2.1. Literature search. Represents the amount of works for different search sources in each phase.

Search source	**Phase 1**	**Phase 2**	**Phase 3**
www.scopus.com	127	34	28
ieeexplore.ieee.org	25	24	20
link.springer.com	15	12	12
dl.acm.org	17	15	14
research.ebsco.com	60	22	20
scholar.google.com	216	43	27
Review articles	-	-	8
Total			32

by utilizing a combination of the following general parameters:

- Computer Vision, Medical Image Computing, Machine Learning, and Artificial Intelligence are some of the subject areas that will be covered.
- Title Keywords (Boolean Search): medical AND (images OR image) AND ((super AND resolution) OR enhancing).
- From 2015 to 2025 range of years.
- Papers accessible through open access or institutional subscriptions that are only available in the English language are the only ones that are considered.

The results of the literature search process, detailing the final selection, are shown in Table 2.1. The rational selection of relevant papers was organized into three distinct phases to ensure rigor and minimize the likelihood of omitting high-impact publications:

Phase 1: Initial filtering. The initial corpus was filtered by applying the keyword and year criteria, followed by a filter for only peer-reviewed papers. High-impact review articles were specifically included in this phase to establish broad domain context.

Phase 2: Abstract screening. A critical screening of the abstract and introduction of each paper was performed to assess its direct relevance to the SR topic and its methods (generative models).

Phase 3: Prioritization and deep analysis. The subsequent selection favored works that utilized generative models, especially those employing DM. Preference was given to works with a later publication time and those possessing a substantial citation index.

The quantitative analysis of the selected works, categorized by the architecture used (Table 2.2), showed the clear dominance of established deep learning methods, such as Convolutional Neural Network (CNN)s [42, 48, 12, 9, 24, 61, 43, 5, 7, 56, 5] and GANs [1, 63, 44, 31, 16, 34, 10, 17, 52, 21, 57], with a comparatively small amount dedicated to DM methods [46, 58, 11]. This distribution reflects the recent emergence of DMs in the SR domain. Furthermore, an analysis by image modality (Table 2.3) indicated a greater focus on MRI and CT data, likely due to the high diagnostic value and associated acquisition costs of these modalities, making them particularly attractive for the application of SR.

TABLE 2.2. Quantitative analysis of architecture type use cases.

Architecture type	**Use cases**
CNN	39%
GAN	34%
Transformer	21%
Diffusion	6%
Total	100%

2.2. Non-generative methods

Methods for improving image quality can broadly be divided into two categories: traditional image enhancement techniques and learning-based SR approaches. Traditional methods typically rely on interpolation and filtering operations, such as denoising and deblurring [50]. These approaches use manually tuned parameters and are limited in

TABLE 2.3. Quantitative analysis of dataset modalities.

Image Modality	**Use cases**
MRI	35%
CT	27%
X-ray	15%
Ultrasound	6%
Fundoscopy	5%
Dermoscopy	5%
Not specified	6%
Total	100%

their ability to adapt to different types of images, noise patterns, or structural content. A central challenge of these methods is increasing spatial resolution in a meaningful way.

Linear interpolation techniques provide a straightforward and computationally efficient approach to upsampling; however, they do not enhance the perceptual quality of the image. Nonlinear interpolation methods, including nearest neighbor, bilinear, and bicubic, can yield slightly better visual results by incorporating higher-order relationships between pixels [13]. Despite these improvements, such methods are fundamentally limited because they do not leverage the statistical properties or complex patterns present in natural images [50]. Consequently, approaches capable of learning nonlinear mappings from LR to HR images emerged.

These learning-based non-generative methods, commonly referred to as SR networks, often employ DL techniques such as CNN [56, 42, 48, 12, 9, 24, 61, 43]. By their design, these networks typically feature a sequential arrangement of convolutional layers without shortcut connections or complex branching. The data flows linearly from the input to output through successive convolutional transformations.

A notable example is Super-Resolution Convolutional Neural Network (SRCNN) [9], which uses pre-upsampling of the LR input followed by several convolutional layers to learn a pixel-wise mapping. However, subsequent studies such as Shi et al., 2020 [47] have demonstrated that pre-upsampling is suboptimal, increasing computational cost without significantly improving performance. Moreover, shallow architectures are prone to vanishing or exploding gradient problems when extended with additional layers. To address this, recursive blocks and residual connections were introduced [51], while techniques such as batch normalization and regularization further stabilize training and improve generalization [30].

In general, non-generative methods are largely deterministic, producing a single fixed output for a given input. While they are computationally simpler and easier to train, they cannot fully address the ill-posed nature of SR, where multiple HR images may correspond to the same LR input. As a result, these methods have been largely outperformed by generative approaches [56].

2.3. Generative Methods

The progression of generative methods for SR can be described as an evolution process from simpler probabilistic approaches to more advanced, large-scale generative frameworks. Each family of models arose to address the limitations of its predecessors, seeking to balance realism, efficiency, and fidelity in reconstructing HR images from LR inputs. Generative methods were developed to better address the inherent ill-posedness of the SR problem [56]. Unlike non-generative approaches, which learn deterministic mappings between input and output images, generative methods model the underlying probability distributions of HR data [33]. This allows for a non-deterministic output, enabling multiple plausible HR reconstructions for the same LR input, which is more consistent with the nature of ill-posed problems.

The Table 2.4 provides a comparative assessment of prevalent machine learning model families, outlining their distinct architectural approaches, performance advantages, and identified weaknesses.

In general, generative methods have demonstrated superior capability in handling the ill-posed nature of the SR problem, providing multiple plausible solutions and higher perceptual quality [45]. This comparison motivates the exploration of diffusion-based models, which combine the advantages of probabilistic modeling with iterative denoising strategies, offering both high-quality image reconstruction and improved stability in training. The subsequent methodology chapter builds upon these insights to propose a diffusion-based SR framework tailored for efficient and accurate HR reconstruction. The following subsections describe several contemporary generative methods and approaches currently employed in the field.

2.3.1. Variational Autoencoders

The introduction of VAEs by Kingma and Welling [25] represented one of the earliest probabilistic generative approaches. Unlike deterministic autoencoders, which map inputs to fixed latent vectors, VAEs encode inputs into a latent probability distribution (usually Gaussian), from which latent variables are sampled. A key innovation enabling this approach is the reparameterization trick (as mentioned in subsection 1.6.5), which separates stochastic sampling from the parameters of the distribution, allowing backpropagation to compute gradients effectively [26].

In the context of SR, VAEs provided a principled probabilistic framework for reconstructing HR images from degraded inputs. They capture the variability of possible reconstructions by modeling uncertainty in the latent space. This property is particularly well aligned with the ill-posed nature of SR, where multiple plausible HR images may correspond to the same LR input. However, VAEs typically rely on maximizing a likelihood-based objective (e.g., log-likelihood of the observed data), which tends to favor average solutions. As a result, VAEs often produce blurry images that lack high-frequency details and sharp textures. While extensions such as SR-VAEs [14] attempted to mitigate

this issue by adjusting decoder inputs, VAEs as a family remained limited in generating perceptually convincing images.

2.3.2. Generative Adversarial Networks

The advent of GAN [15] marked a dramatic shift in generative modeling. Instead of relying solely on reconstruction losses, GANs introduced an adversarial setup: a generator network synthesizes candidate images, while a discriminator network attempts to distinguish between real and generated samples. Through this min-max optimization, the generator learns not only to approximate the training distribution but also to produce images that are perceptually indistinguishable from real data.

For SR, this adversarial mechanism was transformative. The pioneering Super-Resolution Generative Adversarial Network (SRGAN) model [27] demonstrated that combining adversarial loss with perceptual losses (e.g., feature-space losses using Visual Geometry Group (VGG) CNN [48]) could produce images with sharp, photo-realistic textures, going far beyond what VAEs or traditional CNN-based methods achieved. Later, Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) [55] refined the approach by introducing residual-in-residual dense blocks, removing batch normalization for better stability, and adopting a more sophisticated relativistic adversarial loss. These innovations allowed GAN-based SR models to become a new standard for high-fidelity perceptual quality.

Nevertheless, GANs are not without limitations. Training is notoriously unstable due to the delicate balance between generator and discriminator. Mode collapse, where the generator produces limited variations of outputs, remains a persistent challenge. Moreover, adversarial training is computationally demanding, requiring careful tuning of architectures and loss functions. Thus, while GANs offered sharper results than VAEs, they introduced a new set of challenges related to stability and reproducibility [3].

2.3.3. Autoregressive and Flow-based Models

In parallel to GAN developments, researchers investigated autoregressive and flow-based generative models as alternative probabilistic frameworks. Autoregressive models such as Pixel Recurrent Neural Networks (PixelRNN) and Pixel Convolutional Neural Networks (PixelCNN) [36] model the joint distribution of images as a product of conditional probabilities, generating pixels sequentially based on previously generated ones. This formulation allows for exact likelihood estimation and expressive modeling of image distributions. However, their sequential nature makes inference prohibitively slow for SR, where generating large, HR images pixel by pixel is computationally expensive.

Flow-based models, such as Real-Valued Non-Volume Preserving transformations (RealNVP) [8] and Generative Flow with Invertible 1x1 Convolutions (Glow) [24], took a different approach by learning invertible transformations between data and latent variables. This property enables exact log-likelihood computation and efficient latent-space sampling.

For SR tasks, flow-based models offer the appealing ability to explicitly model image distributions and generate diverse outputs. However, they require large memory footprints due to invertibility constraints and often underperform GANs in producing fine-grained, perceptually realistic textures. Consequently, while theoretically elegant, these models saw limited adoption for practical SR applications.

2.3.4. Transformers

The introduction of transformer architectures [53] brought another breakthrough in generative modeling. Unlike CNNs, which rely on local receptive fields, transformers employ self-attention mechanisms to capture global relationships across the input. This makes them particularly powerful for modeling long-range dependencies, an important factor in SR where distant regions of an image may provide context for reconstructing fine details.

Applied to SR, transformer-based models such as Image Processing Transformer (IPT) [4] and Shifted window transformer-based Image Restoration (SwinIR) [28] demonstrated remarkable improvements. IPT leveraged large-scale pretraining across multiple image-processing tasks, while SwinIR introduced hierarchical window-based attention for efficiency. These approaches significantly advanced SR by producing outputs with both global consistency and local sharpness. However, the computational cost of transformers is high, often requiring large-scale datasets and extensive resources for both training and inference. As a result, while transformers advanced state-of-the-art benchmarks, their deployment in constrained environments remains limited.

2.3.5. Diffusion Models

The most recent paradigm in generative modeling is represented by diffusion models [49, 35, 19]. Inspired by non-equilibrium statistical physics, diffusion models learn to backward a gradual noising process: data is corrupted step by step in a forward diffusion process, and a neural network is trained to iteratively denoise and reconstruct the original data. Unlike GANs, which rely on adversarial optimization, diffusion models minimize a well-defined denoising objective, leading to stable and reliable training.

The introduction of Stable Diffusion [40] brought diffusion models into widespread use, particularly by leveraging a latent diffusion framework. Instead of operating directly in pixel space, these models perform diffusion in a compressed latent space, reducing computational demands while maintaining high-quality results. In SR, conditional diffusion models guide the denoising process using LR images as conditions, effectively reconstructing HR versions with remarkable detail and controllability.

Diffusion models represent the current state-of-the-art in generative modeling. They combine many of the strengths of previous approaches: the stability absent in GANs, the ability to model diverse distributions like VAEs and flows, and the fine detail reconstruction characteristic of transformers. However, diffusion models are not without drawbacks. They require multiple iterative steps for image generation, making them slower compared

to single-pass models like GANs [45]. Furthermore, while latent diffusion reduces computational overhead, training and inference still demand significant resources.

Overall, the historical progression of generative SR models reflects a series of innovations, each addressing prior shortcomings. VAEs introduced a probabilistic foundation but lacked perceptual sharpness. GANs delivered photo-realistic detail but introduced instability. Autoregressive and flow-based models offered exact distribution modeling but were impractical in efficiency [2]. Transformers extended generative power through global context modeling but at a high computational cost [62]. Diffusion models now stand at the frontier, balancing stability, diversity, and fidelity, though challenges in computational efficiency remain. This trajectory underscores the continuous search for models that reconcile realism, efficiency, and accessibility in the pursuit of SR.

TABLE 2.4. Comparison of Generative Model Families for SR

Model Family	**Approach**	**Strengths**	**Weaknesses / Limitations**
VAEs	Encode inputs into latent probability distributions; reconstruct HR images via stochastic decoding; rely on reparameterization trick.	Probabilistic framework; stable training; captures uncertainty; theoretically well-grounded.	Reconstructions are often blurry; lack high-frequency details; limited perceptual quality.
GANs	Generator–discriminator adversarial training; combine pixel, perceptual, and adversarial losses for SR.	Produces sharp, photo-realistic details; strong perceptual quality (e.g., SRGAN, ESRGAN).	Training instability; mode collapse; sensitive to hyperparameters; high computational cost.
Autoregressive Models	Pixel-by-pixel generation; joint distribution modeled as product of conditionals (e.g., PixelCNN, PixelRNN).	Exact likelihood modeling; expressive probability distributions; strong theoretical guarantees.	Very slow inference for large images; impractical for real-time SR.
Flow-based Models	Invertible transformations between image and latent space; exact log-likelihood estimation (e.g., RealNVP, Glow).	Tractable probabilistic modeling; efficient latent space sampling; interpretable.	High memory usage; weaker perceptual quality compared to GANs; complex architectures.
Transformers	Self-attention captures global dependencies; encoder–decoder or hierarchical designs (e.g., IPT, SwinIR).	Excellent at modeling long-range context; scalable; strong benchmark performance; versatile.	High training and inference cost; requires large-scale datasets; slower compared to CNN-based methods.
DMs	Gradual noising (forward) and iterative denoising (backward) processes; conditional guidance for SR; latent diffusion for efficiency.	Stable training; high perceptual fidelity; diverse, controllable outputs; state-of-the-art quality (e.g., Denoising Diffusion Probabilistic Model (DDPM), Stable Diffusion).	Slow sampling due to iterative steps; computationally expensive; requires optimization for practical use.

2.4. Diffusion Models and Super Resolution

Recent progress in image SR has been significantly influenced by the emergence of DMs [41]. These models, originally formalized by Sohl-Dickstein et al. [49], are grounded in concepts

from non-equilibrium statistical physics. They operate on the principle of gradually destroying structure in the data through a forward diffusion process and then reconstructing it in backward. By leveraging this bidirectional process, diffusion models form a powerful class of generative frameworks that have demonstrated state-of-the-art results in high-fidelity image generation and SR tasks.

The general workflow of a diffusion model consists of two complementary stages. During the forward diffusion process, structured input data such as an HR image is progressively corrupted by the addition of Gaussian noise across multiple timesteps, eventually transforming it into nearly pure noise. The backward process, parameterized by a learnable model, is then trained to iteratively denoise these corrupted representations, reconstructing the original HR structure step by step. This backward stage is naturally expressed as a Markov chain, where each denoising step depends only on the immediately preceding state, making the process theoretically tractable and stable.

A notable strength of diffusion models is their flexibility: the backward process can be guided by conditioning on external signals such as text embeddings, class labels, or LR images. In the case of SR, the model leverages an input LR image as conditional information, guiding the denoising process toward a plausible HR reconstruction that is consistent with the given input. This conditioning framework also aligns diffusion models with broader multi-modal generative tasks, including text-to-image synthesis and image inpainting.

Despite their advantages, diffusion models present several challenges. On the positive side, they are known for generating high-quality, diverse, and realistic samples, with theoretical guarantees of approximating complex data distributions. They also avoid some of the training instabilities observed in adversarial approaches such as GANs. However, their most prominent drawback is computational inefficiency. The iterative nature of the backward process requires hundreds to thousands of denoising steps, which results in substantial computational and memory demands [19, 40, 2]. This makes them resource-intensive compared to more lightweight SR methods. Furthermore, the dependence on carefully chosen variance schedules and large-scale datasets can complicate training and deployment in real-world medical imaging scenarios.

In summary, diffusion models provide a fundamentally robust and theoretically principled framework for SR, excelling in reconstruction quality and generative flexibility. Nevertheless, their practical adoption is currently limited by computational constraints, motivating ongoing research into more efficient sampling strategies, architectural optimizations, and lightweight generative frameworks for SR.

2.5. Evaluation

SR approaches are difficult to evaluate since the reconstruction problem is ill-posed, allowing many viable HR results from a single LR input. Thus, evaluating reconstructed images requires combining objective signal fidelity criteria (pixel-wise comparison) with

subjective visual quality and, most importantly, the system's intended diagnostic performance. This section describes the quantitative measurements used to evaluate these three essential SR quality factors [62].

The PSNR serves as a metric for quantifying the disparity between the SR image and the corresponding ground truth HR image, specifically in relation to pixel values [23]. A higher PSNR typically signifies an improvement in quality. Nonetheless, it does not consistently align with human perception in a precise manner.

The SSIM evaluates the degree of similarity between the SR image and the corresponding HR ground truth image by taking into account factors such as luminance, contrast, and structural information [59]. SSIM frequently demonstrates a closer correlation with human visual perception in comparison to PSNR.

The Learned Perceptual Image Patch Similarity (LPIPS) metric quantifies the perceptual similarity between images by employing a pre-trained deep neural network [22]. LPIPS frequently demonstrates a strong correlation with human assessments of image quality.

The Fréchet Inception Distance (FID) serves as a metric for evaluating the similarity between the feature distributions of SR images and their corresponding ground truth HR images, utilizing the Inception network for this comparative analysis. FID is frequently employed as a metric to assess the quality of generated images, encompassing those that have undergone SR processes [62].

Scaling factor is the most important property, therefore evaluation of the model's performance across various upscaling factors is essential for understanding its limitations and capabilities.

Artifact analysis involves a thorough examination of SR images to identify the presence of artifacts, including but not limited to blurring, ringing, and checkerboard patterns.

2.6. Discussion

Although a wide range of SR approaches have been proposed, relatively few works evaluate their effectiveness in downstream classification tasks. In particular, the influence of generated SR images on reducing false negatives when processed by neural network classifiers remains underexplored. Since false negatives correspond to missed detections, their reduction is especially relevant for improving the overall reliability of classification systems. This creates a research gap where SR models should not only be assessed based on visual quality metrics such as PSNR or SSIM, but also on their impact on classification accuracy.

Diffusion models, which have demonstrated remarkable results in generative tasks such as image synthesis and restoration, provide a promising direction for SR. Their iterative denoising procedure enables the recovery of structural details that can be critical for feature extraction in classifiers. However, existing diffusion-based methods often rely on

large and computationally expensive architectures that are impractical for many real-world scenarios [19, 41]. These resource demands present a barrier to their widespread adoption, particularly when efficiency and scalability are required.

CHAPTER 3

Methodology

This chapter outlines the methodological framework adopted in this thesis. It describes the design of the proposed diffusion-based SR model, the integration of conditional-guided denoising, and the strategies used to evaluate its effectiveness. Furthermore, details of the experimental setup, datasets, preprocessing steps, and evaluation metrics are presented to ensure reproducibility and to provide a clear foundation for subsequent analysis.

The focus of this thesis is therefore to investigate the feasibility of constructing a lightweight diffusion-based SR model designed to enhance classification performance, with a particular emphasis on reducing false negatives disease classifications. The central approach involves conditional-guided denoising, where the input of class label serves as conditioning information to progressively reconstruct a HR output. While conditional generation has been explored in previous studies (e.g., Wang et al. [54]), these efforts have not applied stable diffusion frameworks in the context of classification-focused SR. This research aims to close that gap by exploring a minimalist diffusion design that balances reconstruction quality with limited computational resources, thereby improving downstream classification outcomes.

To achieve this objective, a class-guided diffusion-based super-resolution model is developed. The proposed model consists of several main components, including a U-Net backbone for image generation and embedding modules for class-conditioning, timestep, LR image. The model is trained to generate SR reconstructions from LR inputs while being guided by class information. In parallel, a classifier network based on Residual Network (ResNet-18) architecture is fine-tuned and trained to classify HR and generated SR images to evaluate the effectiveness of the diffusion-based super-resolution model in the context of reducing FN classifications. The challenge in this research was the scarcity of publicly available medical datasets that contain paired LR and HR images captured simultaneously in real-world clinical settings that were also classified. Consequently, the common and established methodology of synthetically generating LR images by downscaling the HR source images were adopted, which effectively models the degradation process observed in typical real-life scenarios.

The experimental procedure involves using the trained diffusion model to super-resolve LR images, which are then classified by the ResNet-18 network. The resulting classifications are compared to those obtained from the corresponding HR ground-truth images. This comparison provides a quantitative basis to assess whether the generated SR images produced by the proposed diffusion model yield classification results that are statistically better of those of the HR images, thereby demonstrating improved precision and accuracy.

All classification results throughout the experiments are obtained using the same trained ResNet-18 classifier to ensure consistency.

3.1. Diffusion model architecture

The core of the diffusion SR model is the denoising neural network, commonly a form of a U-Net [42], owing to its efficacy in image-to-image translation tasks. The denoising network requires knowledge of the current timestep to determine the level of noise present; hence, the scalar timestep t must be encoded into a comprehensible representation for the neural network. This is achieved by using noise scheduler and time step embedding. In the conditional SR task, the model must be directed by the LR image. This LR image is integrated into U-Net to assist the model in predicting spatially dependent data for HR image reconstruction.

The network architecture, shown in Figure 3.1, uses a DM structure on a U-Net backbone. HR images can be generated from pure noise using a LR input with this design. The system workflow comprises two phases: the forward and backward process. U-Net is trained to do backward diffusion (reconstruction), shown by solid blue arrows. After starting with random noise, the network anticipates and removes it over a succession of discrete timesteps to refine the image until the HR image is reconstructed. The dashed gray arrows indicate forward diffusion (learning), where noise is analytically introduced to the ground-truth HR image at various timesteps to give the U-Net noisy inputs. The U-Net uses conditioning methods at many encoder and decoder layers to adjust the unconditional diffusion process for SR spatial conditioning (LR image): the LR input guides high-frequency detail development by providing structural and low-frequency context. To ensure the HR image is geometrically compatible with the initial LR input, injections (red arrows) are embedded across the U-Net's feature maps. Temporal conditioning (timestep) provide the network a time embedding from diffusion sequence step (green arrows). This embedding helps the U-Net determine the noise level, enabling it to predict and eliminate it at each phase of the backward diffusion process. The class embedding for the guided denoising process is conditionally not shown in the figure.

3.1.1. U-NET structure

The U-Net design is derived from the neural network for image processing introduced in [42], with several modifications. It comprises two primary, nearly symmetric components: the encoder (downsampling pathway) and the decoder (upsampling pathway), featuring "skip connections" that preserve spatial information. Encoder and decoder constructed with analogous blocks comprising many layers.

The downsampling U-Net block progressively diminishes the spatial resolution of the input while augmenting the channel depth, hence retrieving hierarchical features. The upsampling incrementally enhance the spatial resolution of the features by rebuilding the image from the acquired representations. These blocks comprises two sub-blocks of convolutional layers, succeeded by Rectified Linear Unit (ReLU) activation and normalization,

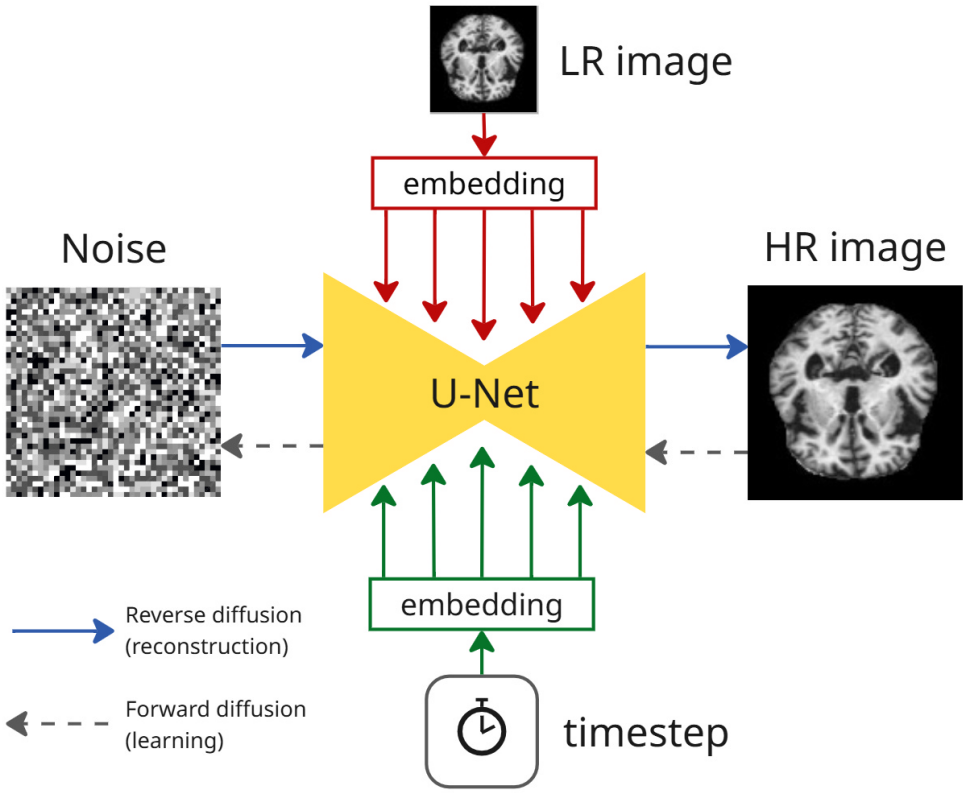


FIGURE 3.1. Overall Architecture of the Conditioned Diffusion U-Net for SR

with a dropout layer placed between them. Each block integrates embedded timestep and conditional elements. A downsample (or upsample) convolutional layer is located at the end of the each block. The only difference in upsampling layer that it employs a fractionally-strided convolution [12]. Blocks next to the U-Net bottleneck incorporate an self-attention sub-block following the downsample layer, another self-attention block placed after the last upsampling block.

Additionally, there are input and output blocks that bring asymmetry to the U-Net design. The input block is a convolutional layer that converts input data to the feature dimension of the initial block. The output block comprises two concluding convolutional layers interspersed with a ReLU activation layer, which transforms the features to the suitable output channel dimensions. "Skip connections" link features from the encoder path directly to the decoder path at matching spatial resolutions by concatenation.

3.1.2. Noise Scheduler Design

A noise scheduler regulates the amount of noise introduced to the data during the forward diffusion process and, conversely, the amount of noise eliminated during the backward diffusion process. It is a predetermined function that regulates the pace and method

of information degradation and reconstruction. As mentioned, $\beta_t \in [0, 1]$ variance from Equation 1.8 that represents variance and schedules the amount of noise added at the each step of degradation process. The most simple method is to schedule it linearly, as indicated by Ho et al. [2], where the variance spans from 0.0001 to 0.02 and the noise incrementally grows with each step:

$$\beta_t = \beta_{start} + (\beta_{end} - \beta_{start}) \cdot (t/T); \tag{3.1}$$

where t is a diffusion timestep, T is a total number of diffusion timesteps.

This strategy, while simple, results in inconsistent learning; during the initial stages, minimal noise is introduced, causing the model try to reconstruct practically flawless images. Conversely, in the latter stages, the excessive noise renders the denoising process nearly impractical. Additionally, in the initial stages, this results in minimal gradients that hinder substantial learning. To address these issues Nichole et al. [35] propose an improved cosine scheduler that aims to provide a smoother and more robust schedule for $\bar{\alpha}_t$ (the cumulative signal retention), rather than directly β_t. It ensures that $alpha_t$ decays slowly at the beginning and end, and more rapidly in the middle, creating a bell-shaped distribution:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \text{ where } f(t) = \cos\left(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2}\right)^2, \tag{3.2}$$

and s is a small positive constant ($s = 0.008$) to avoid numerical instability near $t = 0$; then the α_t and β_t calculated as follows:

$$\alpha_t = \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, \beta = 1 - \alpha_t \tag{3.3}$$

While cosine scheduler shows better generation quality it is more complex computationally, requires correct implementation and it's specific shape might not be optimal for all diffusion tasks.

In this work, both approaches were used to determine the optimal configuration for this work. Although the cosine scheduler is designed to preserve image information longer, our experiments showed that the marginal gains in image quality did not justify the increased computational complexity. Consequently, the linear scheduler was selected as the final approach due to its efficiency and the lack of substantial performance differences between the two methods.

3.1.3. Time Step Embedding

The objective of timestep embedding is to inform the neural network of the present timestep t in the diffusion process and correspondingly calculate the current noise level of the input image. The most common and effective method for time step embedding comes from positional encoding in Transformer networks [53]. This technique shows simplicity, efficacy, and capacity to accommodate diverse timestep ranges. It maps the integer timestep t into a higher-dimensional vector utilizing sine and cosine functions of varying

frequency:

$$PE(t)_{2i} = \sin\left(\frac{t}{10000^{2i/D}}\right), \tag{3.4}$$

$$PE(t)_{2i+1} = \cos\left(\frac{t}{10000^{2i/D}}\right), \tag{3.5}$$

where t is a current timestep, D is a time embedding vector dimensionality, i is a counter that determines frequency component and ranges from 0 to $D/2 - 1$.

3.1.4. Conditional Embedding

The diffusion model can generate an image from noise; however, in the absence of supplementary instruction, it will only yield the most probable representation. To reconstruct a HR image corresponding to a LR one, the model must be supplied with information from the LR image. This approach enables the model to acquire basic data and critical information for processing during the reconstruction phase.

Various methodologies exist to incorporate the LR image to aid in the backward diffusion process. The simplest method is the concatenation of the up-scaled image tensor to align with the required dimensions of the model layer. Another prevalent method is processing the image through a straightforward CNN which makes the embedding process learnable. The comparison of these two strategies revealed no improvements in model correctness, leading to the decision to adopt the simplest option, namely concatenation. LR image tensors are concatenated to each layer of the U-Net to aid in preserving spatial information that directs the SR process.

3.1.5. Normalization

The normalization procedure was utilized primarily to stabilize the layer statistics independent of the batch size. This offers numerous advantages: it aids in preventing vanishing or exploding gradients, facilitating more stable and efficient training; it accommodates significant variations in magnitude and distribution of noise across time steps, consistently managing input statistics; it enhances generalization to unseen data; and it provides a straightforward and effective method for incorporating conditional information.

Group normalization was employed because to the very small batch size, rendering batch normalization insufficiently effective. Furthermore, as noted in section 3.1.6, the Feature-wise Linear Modulation (FiLM) class embedding methodology was employed to incorporate conditional information on image classification.

3.1.6. Class-guided approach

The class-guided or class-conditional technique enables the diffusion SR model to reconstruct HR images based on the specified class hints. This is accomplished by incorporating class-embedded modulation into the U-Net layers. Multiple methodologies exist to regulate the denoising process utilizing class embedding: early fusion (or concatenation), integration of class embedding with timestep embedding, cross-attention [40], and adaptive normalization [45]. Upon evaluating the merits and drawbacks of each strategy, with

the selected model architecture, FiLM emerges as the most suitable and effective approach [37]. FiLM layers incorporate two parameters, scaling γ and shifting β, which are derived from specific conditioning information and are learnable throughout the training phase. The γ and β parameters operate on the feature maps (or activations) of the neural network as follows: $x_i = \gamma x_{i-1} + \beta$, where x_i represents the feature map at the current step. This implements a class-specific affine transformation to the network's intermediate activations, hence directing the denoising process.

3.1.7. Classifier-Free Guidance

Guidance is a crucial aspect of picture generation; nevertheless, in real-life scenarios, the precise classification of the image is often unknown. The evident answer to this issue is to employ a classifier that initially categorizes the LR image, followed by a generator that produces a HR image based on this classification. However, one alternative solution to the problem suggested by Ho et all in [20] involves the generator making inferences independently, without a designated class, producing a picture that implicitly suggests a particular class through its attributes in a LR format. This is accomplished by training the guided diffusion model, which incorporates a "null" class ($\emptyset$) with a probability of 10–20%. The inference process then can be represented as follows:

$$\hat{\epsilon}(x_t, t) = \epsilon_\theta(x_t, t, \emptyset), \tag{3.6}$$

where: $\hat{\epsilon}(x_t, t)$ is a result probability for timestep t;E $\epsilon_\theta(x_t, t, \emptyset)$ is a probability for $\emptyset$ "null" class.

3.2. Experiment Design

The main objective of this experiment was to assess the evaluative capability of a new guided diffusion model for medical picture enhancement. The study sought to illustrate that the SR images produced by this model could yield more precise and dependable classification than their original LR versions. The central hypothesis posited that the model's capacity to synthesis diagnostically pertinent information, including those that are subtle or invisible in the ground truth images, would enhance classification performance. This chapter delineates the comprehensive approach utilized to assess this hypothesis, encompassing dataset preparation, model design, experimental procedures, and evaluation metrics.

3.2.1. Dataset Used

The experiment employed a publicly accessible dataset of brain MRI scans "Best Alzheimer's MRI Dataset" [6]. This dataset consisted of 11,500 scans. Each scan was diligently categorized into one of two classifications: healthy or unhealthy. The "unhealthy" category signifies one of the three phases of neurodegenerative illness. The dataset was divided into two subsets: a training set comprising 10,230 samples and a test set consisting of 1,270 samples. The test set was just utilized for final evaluation and was not incorporated

into any aspect of the diffusion model training or fine-tuning procedure to guarantee an unbiased evaluation of the model's performance.

3.2.2. Classificator model

A robust classification model was necessary to assess the output of the diffusion model. A ResNet18 architecture, a pre-trained CNN [38], was chosen for this objective. ResNet, characterized by its residual connections, alleviates the vanishing gradient issue in deep networks, facilitating efficient training on intricate picture data [18]. The model was modified by substituting the first and last layers to tailor it to the intended objectives. The initial layer was transformed into a convolutional layer for the processing of grayscale pictures, while the final output layer was adjusted to yield a binary classification (healthy/unhealthy). The model was subsequently fine-tuned on the brain MRI training dataset for 35 epochs, employing a learning rate of 10^{-4} with cross-entropy loss criterion and the Adam optimizer. This optimized model functioned as the principal "diagnostic" instrument for both the ground truth and the generated images.

3.2.3. Experiment procedure

The experiment was conducted through a multi-phase procedure to guarantee an equitable and thorough comparison.

(1) Ground Truth Classification: The optimized ResNet18 model was employed to categorize the original, HR ground truth images from the test set. The forecasts and their associated confidence metrics were documented. These outcomes constituted the benchmark for comparison.
(2) SR Image Generation: The reference images in the test set were reduced to a lower resolution (e.g., from 256×256 to 128×128 pixels). The LR images were subsequently input into the guided diffusion model to produce SR images.
(3) The previously fine-tuned ResNet18 model from Step 1 was utilized to classify the resulting SR images. Once more, forecasts and confidence scores were documented.
(4) A comprehensive comparative analysis was performed between the results of Step 1 and Step 3. The aim was to ascertain whether the classification performance on the generated images was superior to or at least comparable with that on the ground truth images.

3.2.4. Evaluation

To thoroughly evaluate the classification model's performance on both the ground truth and generated images, a set of evaluation metrics was employed, with particular focus on reducing false negatives. In a medical diagnostic environment, a false negative (misclassify a "unhealthy" picture as "healthy") might yield grave repercussions, rendering its mitigation a paramount objective. The subsequent metrics were utilized:

- Accuracy

- Precision
- Recall (or Sensitivity)
- False Negative Rate (FNR = 1 - Recall)

To compare classification performance between the two image types on the test set, McNemar's [32] test was performed. The test emphasizes discordant pairs, where the two models classify differently.

A 2x2 Contingency Table 3.1 compares classification results for ground truth and produced images, with rows and columns displaying results for each kind. Discordant pairings b and c are the test variables.

TABLE 3.1. Contingency Table

	Generated image classification is correct	Generated image classification is incorrect
True image classification is correct	a	b
True image classification is incorrect	c	d

The cells indicate the ones that follow:

a: Quantity of pairs of true and generated images that were categorized correctly.

b: The quantity of accurately categorized true images however misclassified generated ones.

c: The quantity of misclassified true images yet accurately classified generated ones.

d: Quantity of misclassified true and generated images.

The hypotheses for McNemar's test are articulated as follows: The null hypothesis (H_0) asserts that there is no statistically significant disparity in the misclassification rates of the two types of images: $H_0 : b = c$

The alternative hypothesis (H_1) posits that there is a large disparity in the misclassification rates of the two types of images: $H_1 : b \neq c$.

The McNemar's test statistic, χ^2, is computed using the discordant pairs (b and c).

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{3.7}$$

The value is subsequently compared to the threshold value of a chi-squared distribution with one degree of freedom. If the computed p-value is below the established significance threshold (e.g. $\alpha = 0.05$), the null hypothesis (H_0) is dismissed. This would yield statistically meaningful evidence that one group of images has a lower misclassification rate than the other. If the p-value is larger than or equal to the significance level, the null

hypothesis should not be rejected, indicating that any observed performance difference is likely attributable to random chance.

3.3. Implementation

This section breaks down the actual implementation of the project and specifies the tools, methods, and approaches used. It is divided into subsections that outline the computing environment and software stack used, detail the methodology for training the models, and explain the approaches applied to improve model performance and ensure computational efficiency.

3.3.1. Software and Hardware

The research study used various essential software tools and libraries to accomplish its objectives. Python was the programming language for the project because of its comprehensive ecosystem and user-friendliness.

The PyTorch library served as the principal framework for the development and implementation of machine learning models. The dynamic computational graph and resilient characteristics were crucial for constructing and training the models. FlashAttention was used to enhance memory efficiency and computational performance during training. In addition, the Memory Snapshot tool from pytorch.cuda was used to provide a detailed visualization of Graphics Processing Unit (GPU) memory utilization. This was essential for diagnosing and rectifying Out-of-memory (OOM) faults by facilitating a comprehensive examination of memory allocation on the GPU.

The NumPy library was utilized for array operations and numerical computations, offering a robust instrument for data manipulation. The einops library was incorporated into the workflow to streamline intricate tensor operations and reorganizations. This package facilitated more natural and comprehensible coding when handling multidimensional data. Matplotlib was employed for data analysis and visualization, producing diverse plots and charts essential for comprehending model performance and data attributes. Finally, the Python Imaging Library (PIL) library was utilized for all requisite image processing and manipulation activities, including loading, resizing, and storing picture data.

All computational operations, involving model training and inference, were performed on a Dell Precision 7560 laptop, including an Intel i7 processor and 64 Gigabyte (GB) of Random Access Memory (RAM). The system employed an NVIDIA RTX A3000 Laptop GPU for accelerated computing. This GPU, which has 6.44 GB of dedicated memory, was utilized to manage the intensive parallel processing necessary for deep learning. The GPU software environment was set up with Compute Unified Device Architecture (CUDA), facilitating the utilization of PyTorch with 2.6.0+cu124 version.

3.3.2. Training process

This project employs medical MRI datasets sourced from the Altzheimer MRI dataset [6]. The dataset is partitioned into training and testing subsets.

The dataset undergoes pre-processing to generate pairs of LR and HR images. A down-sampling method utilizing a scaling factor (e.g., x2) and a cropping dimension (e.g. 128x128) is employed to produce LR inputs. In addition, image pre-processing includes image patching and random reflection. Loss computation entails optimizing pixel-wise discrepancies through Mean Squared Error (MSE). The Adam optimizer facilitates accelerated convergence through the dynamic adjustment of learning rates.

The dataset class is utilized to prepare LR and HR image pairings. The DataLoader class facilitates the generation of iterable batches for training purposes. Batches are randomized to promote model generalization.

The training cycle is executed utilizing the train method, which is responsible for:

- Initializing the dataset loading process.
- Configuring model parameters and the optimizer.
- Systematically adjusting weights according to the computed loss throughout several epochs.
- Regularly saving model checkpoints to retain trained weights.

Graphics Processing Unit Acceleration: during training, computations are expedited by NVIDIA CUDA with effective GPU memory management. Allocated GPU memory utilization about 5.8 GB. The total training time was up to 10 hours. The loss of the training process by epoch in logarithmic scale is represented in Figure 3.2

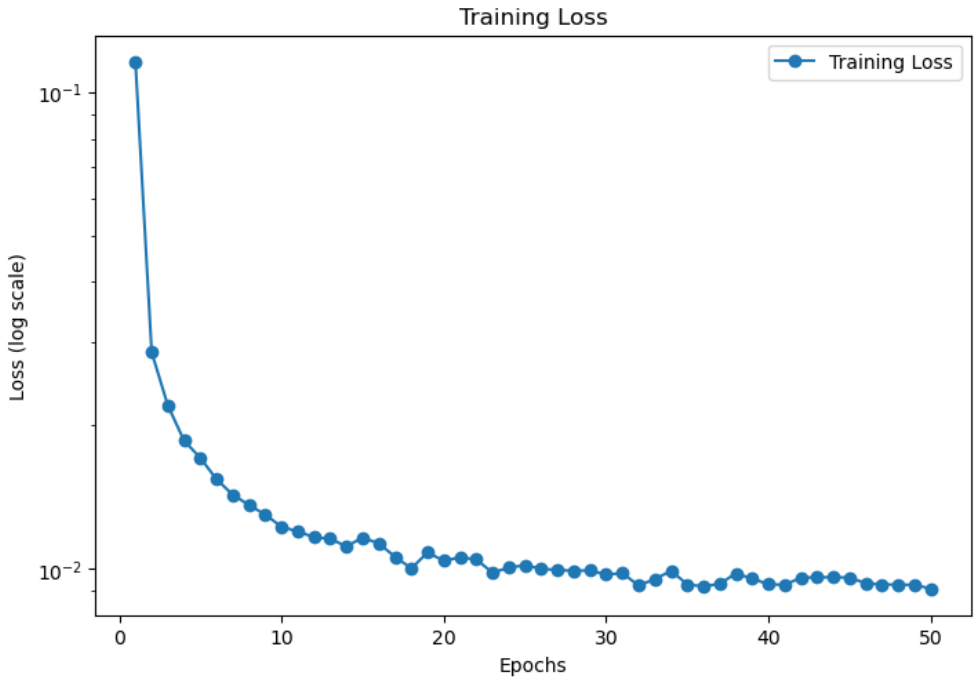


FIGURE 3.2. Training loss over epochs

3.3.3. Optimization and hyperparameters

The optimization approach entailed a careful modification of the model's architecture. In parallel, essential hyperparameters (see Table 3.2) were identified through a sequence

of empirical tests. This iterative method was essential for reconciling computational efficiency with performance, ultimately resulting in a model that was both successful and viable to train within the specified resource limitations.

TABLE 3.2. Hyperparameter values used for training guided diffusion model.

Hyperparameter	**Value**
Number of Epochs	50
Crop Size	64
Batch Size	32
Scale Factor	2
Learning Rate	2e-5
Number of Diffusion Steps	500

A set of optimizations were performed to achieve strong performance within limited computational resources. To address the need for training with small batch sizes, Group Normalization was utilized instead of Batch Normalization, which guarantees steady statistics. The network architecture was constructed using a minimalist U-Net design for parameter efficiency, incorporating two attention layers: one at the bottleneck and another at the final layer, to improve feature refinement and output quality without significant overhead.

Class information was incorporated utilizing a FiLM class embedding, in conjunction with a distinct conditional embedding achieved using concatenation without learnable parameters enabling cost-free feature integration. The efficiency of training was enhanced by employing a diminished number of diffusion steps in the generating process. Ultimately, hardware use was optimized by adjusting the model to efficiently utilize the majority of the allocated GPU memory, hence enabling the maximum feasible batch size and model capacity within the specified limitations.

CHAPTER 4

Experimental results

4.1. Performing the experiment

The experiment was done utilizing a combination of custom Python scripts and a Jupyter Notebook, which served as the primary interface for executing the different phases of the investigation. All code, encompassing model implementations, data loaders, and experimental scripts, was written in Python and employed the PyTorch deep learning framework. The complete source code and associated materials for this thesis are publicly available on GitHub by link in Annex B 5.4. The notebook, included in the project repository as experiment.ipynb 5.4, methodically adhered to the four-step protocol specified in the section 3.2.3 Experiment Procedure.

Execution steps:

- System Configuration: The first phase was to verify the hardware and software configuration of the system. This step guaranteed the experimental environment was appropriately configured for the intended computations.
- The primary focus of the experiment was the training of the guided diffusion model. The training process starts by a function call utilizing pre-defined hyperparameters, such as the number of epochs, learning rate, and batch size. The model completed training for 100 epochs, during which the loss exhibited a consistent decline, signifying effective learning.
- SR Generation and Classification: Subsequent to the training, the experiment progressed to the evaluation phase. The images in the test set were initially downscaled to a reduced resolution. The LR images were subsequently input into the trained diffusion model to produce their HR equivalents. The original HR ground truth images and the recently generated SR images were subsequently categorized using the pre-trained and fine-tuned ResNet18 model. The classification predictions and confidence scores for each image category were recorded for further investigation.
- The third phase was a thorough comparison of the classification outcomes. The forecasts and metrics for the ground truth images have been compared with those for the generated images. Essential measures, including Accuracy, Precision, Recall and False Negative Rate, were computed. McNemar's test was performed on a contingency table to ascertain whether a statistically significant difference existed in the classification performance between the two image sets.

4.2. Results

The experimental results give a quantitative evaluation of the guided diffusion model's efficacy in producing medical images appropriate for classification. This part provides an examination of the metrics and statistical evaluation. The results are organized into a comparative analysis of categorization performance, a statistical assessment of the outcomes, and a visual examination of the produced images.

Table 4.1 and Table 4.2 present the confusion matrices for the ResNet18 classification performance on the ground truth images from the test part of the dataset and generated from the LR counterpart of them with guided DM SR images, respectively. The results demonstrate that the classification model achieved high accuracy on both image sets, with improved performance when using SR-generated images. Specifically, the SR-based classification shows an increase in correctly identified healthy and unhealthy samples, reducing false negatives compared to the predicted ground truth images results. This indicates that the SR reconstruction effectively enhances image quality and preserves diagnostically relevant features, leading to more reliable classification outcomes.

TABLE 4.1. Confusion Matrix: Ground true image classification

Real image class	Predicted: Healthy	Predicted: Unhealthy
Healthy	627	12
Unhealthy	40	600

TABLE 4.2. Confusion Matrix: Generated SR images classification

Real image class	Generated image predicted class: Healthy	Generated image predicted class: Unhealthy
Healthy	636	3
Unhealthy	24	616

The performance of the ResNet18 classifier on the original ground truth (HR) test images provided the baseline for this study. The classifier achieved a recall of 93.8% and an accuracy of 95.9% on the test dataset. When the identical classifier was applied to the SR images generated from corresponding LR, the recall was recorded at 96.3%, and the accuracy at 97.9%. The results are summarized in Table 4.3: Resulting Metrics indicates the Accuracy, Precision, Recall and FNR for both the ground true HR and generated SR images sets. To further assess whether the observed improvement in classification performance was statistically significant, a McNemar's test was conducted between the

ground truth and SR image predictions. The test evaluates the consistency of classification results for paired samples, providing evidence of whether the difference is due to chance. The McNemar test indicates a statistically significant difference when $p-value < 0.05$, confirming that the improvement obtained using SR-generated images is meaningful and not random variation. Table 4.4 represent the resulting Contingency Table for the McNemar's test.

TABLE 4.3. Resulting Metrics

	Ground true image classification	Generated image classification
Accuracy	0.959	**0.979**
Precision	0.980	**0.995**
Recall	0.938	**0.963**
FNR	0.063	**0.038**

TABLE 4.4. Resulting Contingency Table

	Generated image classification is correct	Generated image classification is incorrect
True image classification is correct	1212	**15**
True image classification is incorrect	**40**	12

According to the McNemar test formula (equation 3.7), the χ^2 score was calculated.

$$\chi^2 = \frac{(15-40)^2}{15+40} = 11.364 \tag{4.1}$$

4.2.1. Statistical significance

The statistical significance of the classification improvement was assessed using McNemar's test, which evaluates the differences in paired classification outcomes between ground truth and SR images. The analysis specifically focused on discordant pairs, where the classifier's predictions differed between the two image types. A total of 55 discordant pairs were identified in this experiment.

- In 15 cases the classifier correctly identified the class using the ground truth image but misclassified the corresponding SR-generated image.
- In contrast, the classifier misclassified 40 ground truth images but correctly classified their SR-generated counterparts.

This finding suggests that the guided diffusion-based SR model introduced additional diagnostically meaningful details that enhanced the classifier's decision-making process. The notable instances in which the classifier accurately predict class of the SR image but

faltered with the original LR image underscore the hypothesis that the guided diffusion model effectively produced novel and classification relevant information, facilitating a more precise prediction by the classifier.

4.2.2. Visual comparison

A comparative visual analysis of a representative medical image in four phases represented in Figures 4.1 and 4.2 demonstrates the effectiveness of the guided diffusion SR model.

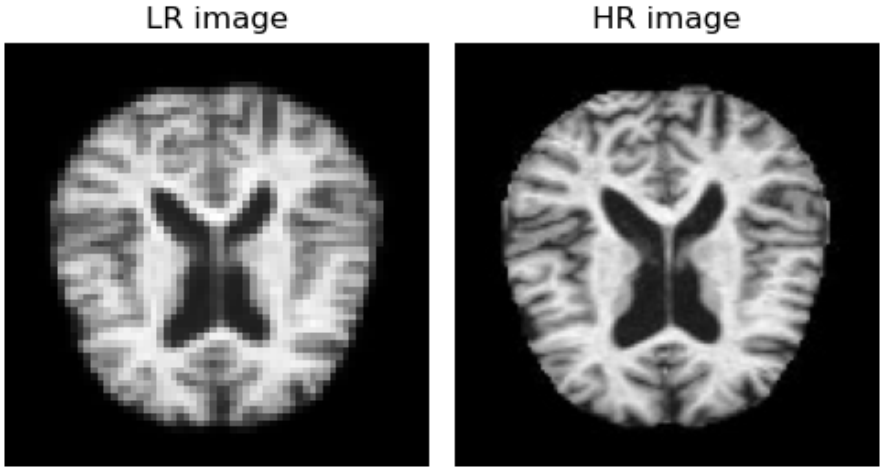


FIGURE 4.1. Side-by-side LR and HR images visual comparison

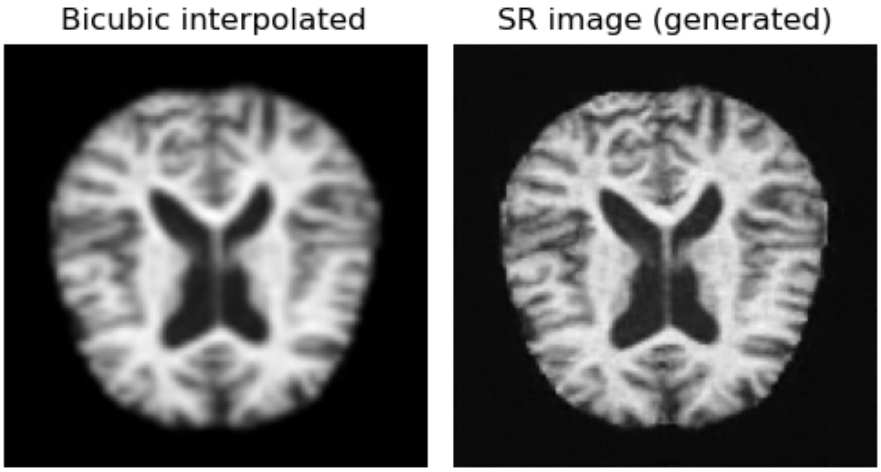


FIGURE 4.2. Side-by-side bicubic upscaled and SR-generated images visual comparison

The Original HR Image represents the unadulterated foundation of the truth. Downscaled images illustrates the scan that has been programmatically downscaled. Pixelation and blurring conceal essential characteristics. Numerous attributes in the preliminary HR assessment are now unclear or absent, making diagnosis difficult or unfeasible. Upscaled image was enhanced by bicubic interpolation from the reduced-resolution image. Increasing the image size does not generate any new data. Interpolation blurs pixels, resulting

in a less detailed, indistinct image. SR by guided diffusion model image accurately reconstructs the fine details and textures of the HR image. The model reinstated sharp edges and clear structures that were diminished during downscaling. These facts are diagnostically pertinent. This demonstrates that the model can generate a high-fidelity image from a low-quality source.

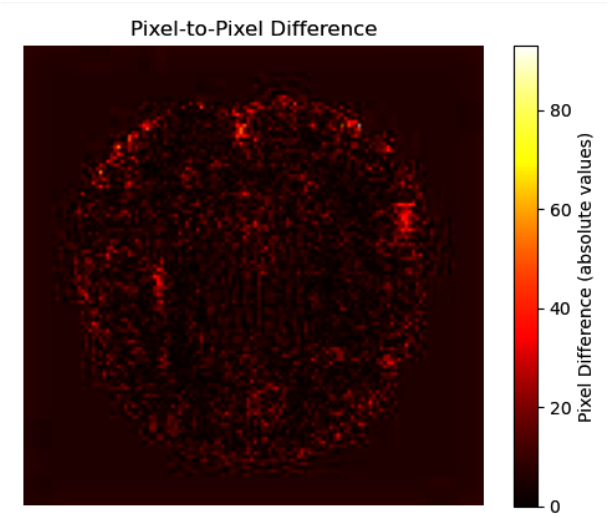


FIGURE 4.3. Absolute difference heatmap between HR and generated SR images

The Figure 4.3 shows the absolute pixel-wise difference between a HR image and its generated SR counterpart. Values closer to zero are displayed in black, while the yellow color highlights regions where the SR image deviates more from the HR reference. This visualization helps evaluate the underlying enhancements of the SR reconstruction in context of classification.

4.3. Interpretation of the results

A significant result from the McNemar test indicates that the improvement in classification performance observed on the generated SR images was not due to random chance, thus providing strong statistical evidence for the efficacy of the SR process. The substantial difference between the c and b values (40 vs. 15) strongly suggests that the diffusion model image's enhancement was not a fluke but a genuine improvement.

The test McNemar statistic showed a chi-squared value of $\chi^2 = 11.364$. A corresponding $p - value$ of 0.000749 is substantially below the conventional significance level of 0.05. The extremely low $p-value$ implies that the observed performance difference, notably the large number of situations where the classifier was correct on the generated SR image but not on the ground true image, is statistically significant. Solid statistical data supports the main hypothesis that diffusion model output increases classification accuracy.

This work demonstrated the potential of a guided diffusion model for medical SR, although it possessed both advantages and limitations that must be acknowledged when interpreting the findings.

Advantages:

- The proposed method improves image clarity by generating high-quality SR images from LR inputs. The SR images preserve and enhance classification-relevant features, achieving performance comparable to or exceeding that obtained from HR images. This suggests that the diffusion model can reconstruct subtle structural details that are not discernible in the original LR data, which is crucial for accurate classification.
- A key strength of this study lies in its emphasis on reducing the false negative rate. Misclassifying an unhealthy case as healthy can lead to serious consequences in medical image analysis. The results demonstrate that the model's improved recall and reduced false negative rate are particularly valuable in contexts where minimizing missed detections is critical.
- Reproducible and Controlled Experimentation: The use of a publicly available dataset and a structured, multi-phase Jupyter Notebook workflow ensures reproducibility. The experimental design is transparent and verifiable, with clearly documented procedures from model training through evaluation.
- Statistical Validation: McNemar's test provides a rigorous statistical framework for comparing classification methods. By analyzing discordant pairs, the study achieves a more comprehensive assessment of classifier performance beyond simple accuracy or recall metrics, confirming the statistical significance of observed differences.

Limitations:

- Dependence on a Single Classification Model: The research relies on the performance of ResNet18. Although this architecture is robust and widely used, alternative classifiers or fine-tuning strategies may yield different results. The observed improvements may partially stem from how ResNet18 interprets the enhanced features produced by the diffusion model.
- While the guided diffusion model aims to generate feature-rich and information-preserving SR images, it may also introduce artifacts or synthesize structures not present in the original data. This study did not include a qualitative, expert-level assessment to identify and mitigate such artifacts. Future work should incorporate expert review to ensure the visual and analytical validity of generated features.
- The experiments were conducted using brain MRI scans related to neurodegenerative conditions. The generalizability of the proposed method to other imaging modalities or clinical scenarios remains unverified. Additional experiments on diverse datasets are necessary to confirm the model's broader applicability.
- Computational Cost: Training guided diffusion models requires substantial computational resources and extended training times. Even with a high-performance

GPU, the long training duration represents a significant limitation that may hinder deployment in resource-constrained environments.

4.4. Discussion

SR models in clinical applications can enhance medical diagnosis, research, and therapeutic planning. This study demonstrates that SR images enhance and maintain downstream classification efficacy, hence facilitating numerous essential therapeutic applications.

Numerous medical imaging techniques, particularly those emphasizing patient safety or rapidity, produce LR images. Low-dose CT scans minimize radiation exposure, while rapid MRI sequences optimize scanner efficiency. The SR model may process LR images in real-time or near real-time to generate HR images comprehensible to radiologists. This improves visual quality and facilitates the detection of minor anomalies, perhaps resulting in earlier and more precise diagnoses.

The research demonstrated that the SR model could generate realistic, high-fidelity visuals from LR input. This is essential for the augmentation of medical AI data. Patient confidentiality concerns and infrequent conditions render medical databases restricted, costly, and difficult to get. This strategy can enhance the robustness and generalizability of machine learning classifiers by generating an extensive synthetic library of SR images from LR data. This can enhance subsequent tasks such as lesion segmentation and illness classification, even with less training data.

Generating HR images from LR inputs may improve clinical efficiency and resource distribution. To maximize patient throughput, hospitals ought to implement expedited, lower-resolution imaging techniques, while SR models refine images during post-processing. This could reduce patient wait times and allocate costly HR imaging equipment for other essential procedures. In remote or resource-constrained areas without specialist imaging equipment, SR technology may enable the utilization of simpler, portable imaging devices, which professionals might enhance for remote diagnostics.

This technology possesses significant potential; yet, generative models, particularly those trained on medical data, may exhibit algorithmic bias.

Algorithmic Bias in Synthesis: The diffusion model acquires knowledge from training data patterns to generate features. The model may amplify subtle data biases, generating or reinforcing features in the SR image that are absent in the patient. If the training data for a condition predominantly exhibits a certain artifact, the model may "hallucinate" or incorporate it into SR images from other patients, resulting in misdiagnosis.

Confirmation Bias: Generative model can create and mitigate cognitive biases. Automation bias may arise when a medical professional relies on an image processed by a SR model. Even if the enhanced components are counterfeit, they may be excessively utilized. This highlights the necessity for skilled professionals to oversee and authenticate generated medical images.

CHAPTER 5

Conclusions and future work

This chapter summarizes the main findings of the study and outlines future steps to corroborate its relevance and correct its existing problems.

5.1. Summary

This thesis presented an investigation into the use of guided diffusion probabilistic model for SR of medical images, focusing particularly on their potential to improve classification performance by reducing false negatives. The research explored how diffusion-based image reconstruction could enhance image fidelity and feature visibility, thereby supporting more accurate automated classification outcomes.

The proposed framework combined a guided diffusion model with a ResNet18 classifier, forming a complete pipeline for medical image enhancement and evaluation. The diffusion model was trained to reconstruct HR medical images from LR inputs while preserving and amplifying structurally relevant features. Through this class-conditioned guidance mechanism, the model generated SR images that emphasized subtle textures and patterns, often imperceptible in the original data.

Experimental evaluation, conducted on a publicly available brain MRI dataset, demonstrated measurable improvements in downstream classification metrics, particularly in the reduction of false negative predictions. When comparing classification results between ground truth and SR-generated images, the McNemar statistical test confirmed that the improvement was statistically significant, indicating that the enhancement was not due to random variation. The study therefore provides good evidence supporting the feasibility of diffusion-based SR as a tool for improving image-based classification reliability.

Beyond quantitative improvements, visual assessments revealed that the generated SR images contained enhanced fine-grained structural details while maintaining consistency with their corresponding ground-truth images. These enhancements were especially notable in regions containing subtle pathological cues that often contribute to misclassification in lower-resolution inputs.

Nevertheless, this research represents primarily a proof of concept rather than a finalized clinical solution. While the proposed diffusion SR model demonstrated promising results, several limitations remain. The experiments were conducted under controlled conditions using a single dataset, minimalist architecture, and binary classification task. Therefore, further research is necessary to validate the generalizability of these findings

across different imaging modalities, anatomical regions, and classification models. Additionally, the computational demands of diffusion models continue to pose a challenge for large-scale deployment in clinical environments.

In summary, the study confirms that guided diffusion models can meaningfully enhance image resolution and classification performance in medical imaging contexts, particularly by mitigating false negatives. However, this work should be viewed as an initial step, i.e. a conceptual demonstration that establishes the feasibility and promise of diffusion-based SR. Future studies should focus on refining the model's architecture, improving computational efficiency, integrating multi-modal datasets, and conducting expert-level clinical validation to fully assess its applicability in real-world diagnostic workflows.

5.2. Limitations

Despite the study yielding promising results, certain limitations must be acknowledged. The results originate from a specific dataset of brain MRI scans, and the model's application to images of various body regions, tissues and organs remains to be assessed. The assessment utilized a singular ResNet18 classifier, and the noted enhancements may be particular to this architecture and its acquired characteristics. A more thorough investigation with various classifiers might be advantageous. Moreover, the study lacked a qualitative evaluation by medical experts to confirm that the synthesis information is actually beneficial and does not include deceptive artifacts. The substantial computational expense and time-consuming process of training the diffusion model present a practical obstacle to its extensive use in resource-constrained settings.

5.3. Ethical, Legal, and Clinical Implications

This study's findings present numerous ethical and legal implications particularly pertinent to generative modeling in medical imaging. The finding that classifiers assessed on super-resolved images may exceed the performance of those utilizing original HR images necessitates careful interpretation. This effect may be partially attributed to implicit denoising or feature enhancement, yet it also highlights the potential that the SR process exaggerates image patterns that correspond with the classifier's learned decision boundaries rather than accurately depicting the underlying anatomical structures.

This study serves as a proof of concept and not a clinically verified system. The suggested methodology is constrained by several drawbacks, including dependence on a single dataset, a restricted variety of imaging modalities, and an evaluation predominantly focused on automated classification metrics instead of expert clinical judgment. Furthermore, the SR model lacks uncertainty estimates and assurances of anatomical accuracy, both of which are critical for medical application.

The identified constraints indicate that the proposed approach is not appropriate for clinical use in its current state. The creation of super-resolved images may produce hallucinated or exaggerated characteristics that, if utilized without proper controls, could

skew clinical interpretation or subsequent decision-making. Therefore, generated SR images should not be treated as accurate reconstructions of human anatomy and must not be utilized directly for diagnostic purposes.

From a legal and regulatory standpoint, systems that alter medical images for diagnostic purposes are generally categorized as high-risk medical software according to the European Union Medical Device Regulation guidelines [39]. Compliance with these regulatory standards would require comprehensive clinical validation, prospective investigations, performance audits, and expert evaluation, none of which is included in the scope of this thesis.

This work addresses ethical risks throughout the research phase by ensuring a clear distinction between training and test data, providing precise statistical significance values instead of binary assertions, and explicitly recognizing methodological constraints. Subsequent research must emphasize rigorous validation across diverse datasets, integration of uncertainty estimation, and methodical human-in-the-loop assessment to ascertain if the detected enhancements reflect clinically significant insights rather than artifacts generated by the process.

5.4. Future Research Directions

This study's findings and limitations indicate numerous potential directions for future investigation. Initially, it is essential to evaluate the model's generalizability by implementing the SR framework across several medical imaging datasets, encompassing different anatomical regions and imaging modalities. A direct collaboration with radiologists and other medical professionals would be essential for the qualitative evaluation of the generated images, ensuring that the synthesis data is clinically pertinent and free from detrimental artifacts. A crucial avenue for future research is the establishment of a real-time or near-real-time inference pipeline to render the SR technique feasible for incorporation into clinical workflows. Addressing these issues would further reinforce the efficacy of guided diffusion models as a powerful instrument for medical picture enhancement and a significant resource in diagnostic medicine.

References

[1] Waqar Ahmad et al. "A new generative adversarial network for medical images super resolution". en. In: *Scientific Reports* 12.1 (June 2022), p. 9533. ISSN: 2045-2322. DOI: 10.1038/s41598-022-13658-4. URL: https://www.nature.com/articles/s41598-022-13658-4 (visited on 10/12/2025).

[2] Jacob Austin et al. *Structured Denoising Diffusion Models in Discrete State-Spaces*. Feb. 22, 2023. arXiv: 2107.03006 [cs]. URL: http://arxiv.org/abs/2107.03006.

[3] Aaz Azis and Yogi Saputra. "Comparative Analysis of Variational Autoencoder (VAE) and Generative Adversarial Network (GAN) Algorithms for image". In: *JESII: Journal of Elektronik Sistem InformasI* 1 (Dec. 2023), pp. 75–81. DOI: 10.31848/jesii.v1i2.3299.

[4] Hanting Chen et al. "Pre-Trained Image Processing Transformer". en. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, June 2021, pp. 12294–12305. ISBN: 978-1-6654-4509-2. DOI: 10.1109/CVPR46437.2021.01212. URL: https://ieeexplore.ieee.org/document/9577359/ (visited on 10/12/2025).

[5] Xin Cheng et al. "Medical Image Super-Resolution Reconstruction Based on Multi-Level Adaptive CNN and Hybrid Transformer". In: *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. ISSN: 2156-1133. Dec. 2024, pp. 5675–5682. DOI: 10.1109/BIBM62325.2024.10822649. URL: https://ieeexplore.ieee.org/abstract/document/10822649 (visited on 10/12/2025).

[6] Luke Chugh. *Best Alzheimer's MRI Dataset 99% Accuracy*. en. URL: https://www.kaggle.com/datasets/lukechugh/best-alzheimer-mri-dataset-99-accuracy (visited on 06/29/2025).

[7] Farah Deeba et al. "Wavelet-Based Enhanced Medical Image Super Resolution". In: *IEEE Access* 8 (2020), pp. 37035–37044. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2974278. URL: https://ieeexplore.ieee.org/abstract/document/9000539 (visited on 10/12/2025).

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. en. arXiv:1605.08803 [cs]. Feb. 2017. DOI: 10.48550/arXiv.1605.08803. URL: http://arxiv.org/abs/1605.08803 (visited on 10/17/2025).

[9] Chao Dong et al. "Image Super-Resolution Using Deep Convolutional Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (Feb. 2016), pp. 295–307. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2015.2439281. URL: https://ieeexplore.ieee.org/document/7115171.

[10] Weizhi Du and Shihao Tian. "Transformer and GAN-Based Super-Resolution Reconstruction Network for Medical Images". In: *Tsinghua Science and Technology* 29.1 (Feb. 2024), pp. 197–206. ISSN: 1007-0214. DOI: 10.26599/TST.2022.9010071. URL: https://ieeexplore.ieee.org/abstract/document/10225288 (visited on 10/12/2025).

[11] Vishal Dubey. *Temporal and Spatial Super Resolution with Latent Diffusion Model in Medical MRI images*. arXiv:2410.23898 [eess]. Oct. 2024. DOI: 10.48550/arXiv.2410.23898. URL: http://arxiv.org/abs/2410.23898 (visited on 10/12/2025).

[12] Vincent Dumoulin and Francesco Visin. *A guide to convolution arithmetic for deep learning*. en. arXiv:1603.07285 [stat]. Jan. 2018. DOI: 10.48550/arXiv.1603.07285. URL: http://arxiv.org/abs/1603.07285 (visited on 06/02/2025).

[13] Shreyas Fadnavis. "Image Interpolation Techniques in Digital Image Processing: An Overview". In: *International Journal Of Engineering Research and Application* 4 (Nov. 2014), pp. 2248–962270.

[14] Ioannis Gatopoulos, Maarten Stol, and Jakub M. Tomczak. *Super-Resolution Variational Auto-Encoders*. June 30, 2020. DOI: 10.48550/arXiv.2006.05218. arXiv: 2006.05218 [cs]. URL: http://arxiv.org/abs/2006.05218.

[15] Ian J. Goodfellow et al. *Generative Adversarial Networks*. June 10, 2014. DOI: 10.48550/arXiv.1406.2661. arXiv: 1406.2661 [stat]. URL: http://arxiv.org/abs/1406.2661.

[16] Yuchong Gu et al. "MedSRGAN: medical images super-resolution using generative adversarial networks". en. In: *Multimedia Tools and Applications* 79.29 (Aug. 2020), pp. 21815–21840. ISSN: 1573-7721. DOI: 10.1007/s11042-020-08980-w. URL: https://doi.org/10.1007/s11042-020-08980-w (visited on 10/12/2025).

[17] Rohit Gupta, Anurag Sharma, and Anupam Kumar. "Super-Resolution Using GANs for Medical Imaging". In: *Procedia Computer Science* 173 (2020), pp. 28–35. ISSN: 18770509. DOI: 10.1016/j.procs.2020.06.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050920315076.

[18] Kaiming He et al. *Deep Residual Learning for Image Recognition*. en. arXiv:1512.03385 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1512.03385. URL: http://arxiv.org/abs/1512.03385 (visited on 10/18/2025).

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. Dec. 16, 2020. arXiv: 2006.11239 [cs, stat]. URL: http://arxiv.org/abs/2006.11239.

[20] Jonathan Ho and Tim Salimans. *Classifier-Free Diffusion Guidance*. en. arXiv:2207.12598 [cs]. July 2022. DOI: 10.48550/arXiv.2207.12598. URL: http://arxiv.org/abs/2207.12598 (visited on 06/30/2025).

[21] MF Jiang et al. "FA-GAN: Fused attentive generative adversarial networks for MRI image super-resolution". English. In: *COMPUTERIZED MEDICAL IMAGING*

AND GRAPHICS 92 (Sept. 2021). ISSN: 0895-6111. DOI: 10.1016/j.compmedimag.2021.101969.

[22] Younghyun Jo, Sejong Yang, and Seon Joo Kim. "Investigating Loss Functions for Extreme Super-Resolution". en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Seattle, WA, USA: IEEE, June 2020, pp. 1705–1712. ISBN: 978-1-7281-9360-1. DOI: 10.1109/CVPRW50498.2020.00220. URL: https://ieeexplore.ieee.org/document/9150941/ (visited on 10/18/2025).

[23] Jeremy Jones. *Spatial Resolution | Radiology Reference Article | Radiopaedia.Org*. Radiopaedia. DOI: 10.53347/rID-6318. URL: https://radiopaedia.org/articles/spatial-resolution.

[24] Diederik P. Kingma and Prafulla Dhariwal. *Glow: Generative Flow with Invertible 1x1 Convolutions*. 2018. arXiv: 1807.03039 [stat.ML]. URL: https://arxiv.org/abs/1807.03039.

[25] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". en. In: *Foundations and Trends® in Machine Learning* 12.4 (2019). arXiv:1906.02691 [cs], pp. 307–392. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000056. URL: http://arxiv.org/abs/1906.02691 (visited on 10/17/2025).

[26] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. Dec. 10, 2022. DOI: 10.48550/arXiv.1312.6114. arXiv: 1312.6114 [stat]. URL: http://arxiv.org/abs/1312.6114.

[27] Christian Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, July 2017, pp. 105–114. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.19. URL: http://ieeexplore.ieee.org/document/8099502/.

[28] Jingyun Liang et al. *SwinIR: Image Restoration Using Swin Transformer*. Aug. 23, 2021. arXiv: 2108.10257 [cs, eess]. URL: http://arxiv.org/abs/2108.10257.

[29] Zhengyang Lu and Ying Chen. *Single Image Super Resolution based on a Modified U-net with Mixed Gradient Loss*. en. arXiv:1911.09428 [eess]. Nov. 2019. DOI: 10.48550/arXiv.1911.09428. URL: http://arxiv.org/abs/1911.09428 (visited on 10/18/2025).

[30] Ping Luo et al. *Towards Understanding Regularization in Batch Normalization*. Apr. 24, 2019. DOI: 10.48550/arXiv.1809.00846. arXiv: 1809.00846 [cs]. URL: http://arxiv.org/abs/1809.00846.

[31] Dwarikanath Mahapatra, Behzad Bozorgtabar, and Rahil Garnavi. "Image super-resolution using progressive generative adversarial networks for medical image analysis". In: *Computerized Medical Imaging and Graphics* 71 (Jan. 2019), pp. 30–39. ISSN: 0895-6111. DOI: 10.1016/j.compmedimag.2018.10.005. URL: https://

www.sciencedirect.com/science/article/pii/S0895611118305871 (visited on 10/12/2025).

[32] *McNemar Test - an overview | ScienceDirect Topics*. URL: https://www.sciencedirect.com/topics/medicine-and-dentistry/mcnemar-test (visited on 10/18/2025).

[33] Brian Moser et al. "Hitchhiker's Guide to Super-Resolution: Introduction and Recent Advances". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.8 (Aug. 2023), pp. 9862–9882. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2023.3243794. arXiv: 2209.13131 [cs]. URL: http://arxiv.org/abs/2209.13131.

[34] Priyanka Nandal et al. "Super-Resolution of Medical Images Using Real ESRGAN". In: *IEEE Access* 12 (2024), pp. 176155–176170. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3497002. URL: https://ieeexplore.ieee.org/abstract/document/10752667 (visited on 10/12/2025).

[35] Alex Nichol and Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. en. arXiv:2102.09672 [cs]. Feb. 2021. DOI: 10.48550/arXiv.2102.09672. URL: http://arxiv.org/abs/2102.09672 (visited on 05/21/2025).

[36] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. *Pixel Recurrent Neural Networks*. en. arXiv:1601.06759 [cs]. Aug. 2016. DOI: 10.48550/arXiv.1601.06759. URL: http://arxiv.org/abs/1601.06759 (visited on 10/17/2025).

[37] Ethan Perez et al. *FiLM: Visual Reasoning with a General Conditioning Layer*. en. arXiv:1709.07871 [cs]. Dec. 2017. DOI: 10.48550/arXiv.1709.07871. URL: http://arxiv.org/abs/1709.07871 (visited on 06/06/2025).

[38] *resnet18 — Torchvision main documentation*. URL: https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html (visited on 10/18/2025).

[39] *Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC (Text with EEA relevance.)* en. Legislative Body: CONSIL, EP. Apr. 2017. URL: http://data.europa.eu/eli/reg/2017/745/oj (visited on 12/22/2025).

[40] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. Apr. 13, 2022. arXiv: 2112.10752 [cs]. URL: http://arxiv.org/abs/2112.10752. Pre-published.

[41] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. Apr. 13, 2022. DOI: 10.48550/arXiv.2112.10752. arXiv: 2112.10752 [cs]. URL: http://arxiv.org/abs/2112.10752.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. en. arXiv:1505.04597 [cs]. May 2015. DOI:

10.48550/arXiv.1505.04597. URL: http://arxiv.org/abs/1505.04597 (visited on 05/30/2025).

[43] Sudhakar Sengan et al. "Images super-resolution by optimal deep AlexNet architecture for medical application: A novel DOCALN1". EN. In: *Journal of Intelligent & Fuzzy Systems* 39.6 (Dec. 2020). Publisher: SAGE Publications, pp. 8259–8272. ISSN: 1064-1246. DOI: 10.3233/JIFS-189146. URL: https://journals.sagepub.com/action/showAbstract (visited on 10/12/2025).

[44] Walid El-Shafai et al. "Hybrid Single Image Super-Resolution Algorithm for Medical Images". en. In: *Computers, Materials & Continua* 72.3 (2022), pp. 4879–4896. ISSN: 1546-2226. DOI: 10.32604/cmc.2022.028364. URL: https://www.techscience.com/cmc/v72n3/47559 (visited on 10/12/2025).

[45] Arundhati S. Shanbhag et al. *Diffusion Models, Image Super-Resolution And Everything: A Survey*. Feb. 6, 2024. arXiv: 2401.00736 [cs]. URL: http://arxiv.org/abs/2401.00736.

[46] Wennuo Shi, Siyao Zhou, and Lunhao Hu. "Medical Oral Image Super-Resolution Reconstruction Algorithm Based on Stable Diffusion Model". In: *2023 International Conference on Artificial Intelligence and Automation Control (AIAC)*. Nov. 2023, pp. 104–107. DOI: 10.1109/AIAC61660.2023.00039. URL: https://ieeexplore.ieee.org/abstract/document/10491714 (visited on 10/12/2025).

[47] Wenzhe Shi et al. *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*. Sept. 23, 2016. DOI: 10.48550/arXiv.1609.05158. arXiv: 1609.05158 [cs]. URL: http://arxiv.org/abs/1609.05158.

[48] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. en. arXiv:1409.1556 [cs]. Apr. 2015. DOI: 10.48550/arXiv.1409.1556. URL: http://arxiv.org/abs/1409.1556 (visited on 10/17/2025).

[49] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning Using Nonequilibrium Thermodynamics". In: ().

[50] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Cham: Springer International Publishing, 2022. ISBN: 978-3-030-34371-2 978-3-030-34372-9. DOI: 10.1007/978-3-030-34372-9. URL: https://link.springer.com/10.1007/978-3-030-34372-9.

[51] Ying Tai, Jian Yang, and Xiaoming Liu. "Image Super-Resolution via Deep Recursive Residual Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). July 2017, pp. 2790–2798. DOI: 10.1109/CVPR.2017.298. URL: https://ieeexplore.ieee.org/document/8099781.

[52] Yasuhiko Terada et al. “Clinical Evaluation of Super-Resolution for Brain MRI Images Based on Generative Adversarial Networks”. In: *Informatics in Medicine Unlocked* 32 (2022), p. 101030. ISSN: 23529148. DOI: 10.1016/j.imu.2022.101030. URL: https://linkinghub.elsevier.com/retrieve/pii/S2352914822001721.

[53] Ashish Vaswani et al. *Attention Is All You Need*. Aug. 1, 2023. arXiv: 1706.03762 [cs]. URL: http://arxiv.org/abs/1706.03762. Pre-published.

[54] Jingwei Wang et al. “Medical Image Super-Resolution via Diagnosis-Guided Attention”. en. In: *2023 IEEE International Conference on Multimedia and Expo (ICME)*. Brisbane, Australia: IEEE, July 2023, pp. 462–467. ISBN: 978-1-6654-6891-6. DOI: 10.1109/ICME55011.2023.00086. URL: https://ieeexplore.ieee.org/document/10219730/ (visited on 06/06/2025).

[55] Xintao Wang et al. “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks”. In: *Computer Vision – ECCV 2018 Workshops*. Ed. by Laura Leal-Taixé and Stefan Roth. Vol. 11133. Cham: Springer International Publishing, 2019, pp. 63–79. ISBN: 978-3-030-11020-8 978-3-030-11021-5. DOI: 10.1007/978-3-030-11021-5_5. URL: https://link.springer.com/10.1007/978-3-030-11021-5_5.

[56] Zhihao Wang, Jian Chen, and Steven C. H. Hoi. “Deep Learning for Image Super-Resolution: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.10 (Oct. 2021), pp. 3365–3387. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2020.2982166. URL: https://ieeexplore.ieee.org/document/9044873.

[57] Yan Xia et al. “Super-Resolution of Cardiac MR Cine Imaging using Conditional GANs and Unsupervised Transfer Learning”. en. In: *Medical Image Analysis* 71 (July 2021), p. 102037. ISSN: 13618415. DOI: 10.1016/j.media.2021.102037. URL: https://linkinghub.elsevier.com/retrieve/pii/S1361841521000839 (visited on 04/19/2023).

[58] Yushen Xu et al. “Simultaneous Tri-Modal Medical Image Fusion and Super-Resolution Using Conditional Diffusion Model”. en. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Ed. by Marius George Linguraru et al. Cham: Springer Nature Switzerland, 2024, pp. 635–645. ISBN: 978-3-031-72104-5. DOI: 10.1007/978-3-031-72104-5_61.

[59] Tomoki Yoshida et al. *Image Super-Resolution using Explicit Perceptual Loss*. en. Version Number: 1. 2020. DOI: 10.48550/ARXIV.2009.00382. URL: https://arxiv.org/abs/2009.00382 (visited on 10/18/2025).

[60] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. *ResShift: Efficient Diffusion Model for Image Super-resolution by Residual Shifting*. Oct. 18, 2023. arXiv: 2307.12348 [cs]. URL: http://arxiv.org/abs/2307.12348. Pre-published.

[61] Shengxiang Zhang et al. “A Fast Medical Image Super Resolution Method Based on Deep Learning Network”. In: *IEEE Access* 7 (2019), pp. 12319–12327. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2871626. URL: https://ieeexplore.ieee.org/abstract/document/8471089 (visited on 10/12/2025).

[62] Zhicun Zhang et al. “Network Architecture for Single Image Super-Resolution: A Comprehensive Review and Comparison”. In: *IET Image Processing* 18.9 (2024), pp. 2215–2243. ISSN: 1751-9667. DOI: 10.1049/ipr2.13100. URL: https://onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.13100.

[63] Jin Zhu et al. “Arbitrary Scale Super-Resolution for Medical Images”. In: *International Journal of Neural Systems* 31.10 (Oct. 2021). Publisher: World Scientific Publishing Co., p. 2150037. ISSN: 0129-0657. DOI: 10.1142/S0129065721500374. URL: https://www.worldscientific.com/doi/abs/10.1142/S0129065721500374 (visited on 10/12/2025).

Annex A: Experiment Jupyter Notebook

In [1]:
```
# main libraries and dependencies
import importlib
import os
os.environ['KMP_DUPLICATE_LIB_OK'] = 'True' # matplotlib on Windows
import torch
from torch import cuda
from scipy.stats import chi2

# Data class contains class indexes
from data_loader import DataClasses

# Diffusion model trainer function
import trainer
# Diffusion model inference function
from inference import generate, get_sr_model_state, get_scheduler, generate_image

# Classificator model
import classificator

# Experiment function
from experiment_runner import generate_classify_and_calculate, classify_and_calculate

# Some useful functions
import helpers
```

0. Print out important system properties

In [2]:
```
# Check what version of PyTorch is installed
print("PyTorch version: \t", torch.__version__)

# Check if FlashAttention is available
print("FlashAttention: \t", torch.backends.cuda.flash_sdp_enabled())

# Device configuration
device = torch.device("cpu")
if torch.cuda.is_available():
    device = torch.device("cuda")
    # Check total GPU memory and memory used
    gpu_id = 0  # Change for multiple GPUs
    total_memory = torch.cuda.get_device_properties(gpu_id).total_memory
    reserved_memory = torch.cuda.memory_reserved(gpu_id)
    allocated_memory = torch.cuda.memory_allocated(gpu_id)
    free_memory = total_memory - reserved_memory - allocated_memory

    # print the device name
    print("\nDevice name:\t\t", torch.cuda.get_device_properties("cuda").name)
    print("CUDA Version: \t\t", torch.version.cuda)
    print(f"Total GPU memory: \t{total_memory / 1e9:.2f} GB")
    print(f"Reserved memory: \t{reserved_memory / 1e9:.2f} GB")
    print(f"Allocated memory: \t{allocated_memory / 1e9:.2f} GB")
    print(f"Free memory : \t\t{free_memory / 1e9:.2f} GB")
```

```
PyTorch version:         2.6.0+cu124
FlashAttention:          True

Device name:             NVIDIA RTX A3000 Laptop GPU
CUDA Version:            12.4
Total GPU memory:       6.44 GB
Reserved memory:        0.00 GB
Allocated memory:       0.00 GB
Free memory :           6.44 GB
```

In [3]:
```
# Set seed to maintain experiment reproducible
helpers.set_seed(23)
```

In [4]:
```
# Paths to files
# Dataset locations
image_dir = 'C:/Users/novit/Documents/datasets/best_alzheimer_mri/train'
test_image_dir = 'C:/Users/novit/Documents/datasets/best_alzheimer_mri/test'
# Location of the generated SR images
gen_image_dir = 'C:/Users/novit/Documents/datasets/best_alzheimer_mri/gen'
# Location to save trained model's weights
ddpm_checkpoint_file = 'ddpm_checkpoints/ddpm_checkpoint_sr.pth'
classificator_checkpoint_file = 'classificator_checkpoints/classificator_checkpoint.pth'
# Location to store statistical data
stats_data_file = 'stats/stat_data.json'
```

1. Diffusion model training process

1.1. Setting up training hyperparameters

In [5]:
```
# Hyperparameters
num_steps = 500
crop_size = 64
batch_size = 32
scale_factor = 2
learning_rate = 2e-5
num_epochs = 50
```

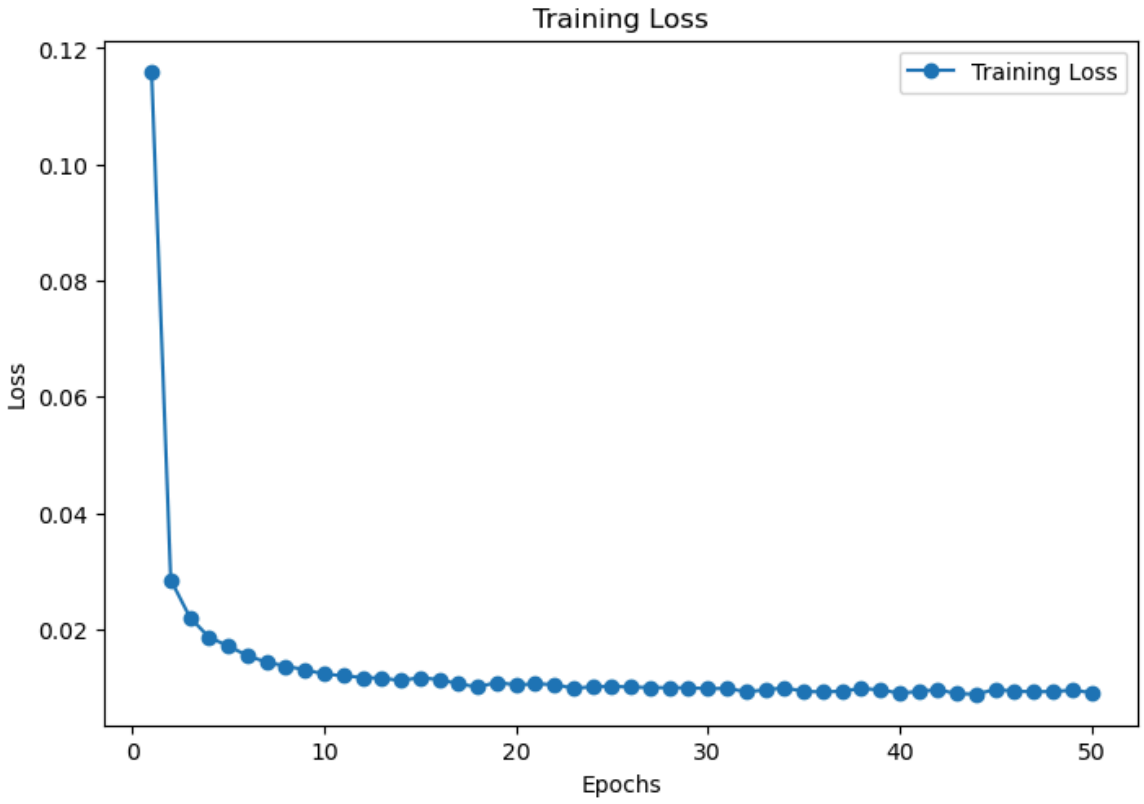
1.2. Run the training process

In [6]:
```
cuda.empty_cache()
trainer.train_sr(
    image_dir=image_dir,
    crop_size=crop_size,
    scale_factor=scale_factor,
    batch_size=batch_size,
    lr=learning_rate,
    num_time_steps=num_steps,
    num_epochs=num_epochs,
    # checkpoint_path='ddpm_checkpoints/ddpm_checkpoint_sr.pth',
    file_to_save=ddpm_checkpoint_file,
)
```

```
Dataset size: 40960
Number of classes: 3
Epoch 1/50: 100%|██████████| 1280/1280 [08:42<00:00,  2.45it/s]
Epoch 1 | Loss 0.11596
Epoch 2/50: 100%|██████████| 1280/1280 [08:46<00:00,  2.43it/s]
Epoch 2 | Loss 0.02850
Epoch 3/50: 100%|██████████| 1280/1280 [08:45<00:00,  2.43it/s]
Epoch 3 | Loss 0.02203
Epoch 4/50: 100%|██████████| 1280/1280 [09:06<00:00,  2.34it/s]
Epoch 4 | Loss 0.01859
Epoch 5/50: 100%|██████████| 1280/1280 [09:10<00:00,  2.33it/s]
Epoch 5 | Loss 0.01711
Epoch 6/50: 100%|██████████| 1280/1280 [09:12<00:00,  2.32it/s]
Epoch 6 | Loss 0.01543
Epoch 7/50: 100%|██████████| 1280/1280 [08:42<00:00,  2.45it/s]
Epoch 7 | Loss 0.01432
Epoch 8/50: 100%|██████████| 1280/1280 [08:39<00:00,  2.46it/s]
```

```
Epoch 8 | Loss 0.01363
Epoch 9/50: 100%|██████████| 1280/1280 [08:38<00:00,  2.47it/s]
Epoch 9 | Loss 0.01301
Epoch 10/50: 100%|██████████| 1280/1280 [08:38<00:00,  2.47it/s]
Epoch 10 | Loss 0.01227
Epoch 11/50: 100%|██████████| 1280/1280 [08:38<00:00,  2.47it/s]
Epoch 11 | Loss 0.01200
Epoch 12/50: 100%|██████████| 1280/1280 [08:39<00:00,  2.46it/s]
Epoch 12 | Loss 0.01167
Epoch 13/50: 100%|██████████| 1280/1280 [08:39<00:00,  2.46it/s]
Epoch 13 | Loss 0.01158
Epoch 14/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 14 | Loss 0.01116
Epoch 15/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 15 | Loss 0.01165
Epoch 16/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 16 | Loss 0.01132
Epoch 17/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 17 | Loss 0.01061
Epoch 18/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 18 | Loss 0.01004
Epoch 19/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 19 | Loss 0.01084
Epoch 20/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 20 | Loss 0.01041
Epoch 21/50: 100%|██████████| 1280/1280 [08:43<00:00,  2.44it/s]
Epoch 21 | Loss 0.01059
Epoch 22/50: 100%|██████████| 1280/1280 [08:42<00:00,  2.45it/s]
Epoch 22 | Loss 0.01048
Epoch 23/50: 100%|██████████| 1280/1280 [08:42<00:00,  2.45it/s]
Epoch 23 | Loss 0.00983
Epoch 24/50: 100%|██████████| 1280/1280 [08:38<00:00,  2.47it/s]
Epoch 24 | Loss 0.01008
Epoch 25/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 25 | Loss 0.01019
Epoch 26/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 26 | Loss 0.01002
Epoch 27/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 27 | Loss 0.00996
Epoch 28/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 28 | Loss 0.00992
Epoch 29/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 29 | Loss 0.00994
Epoch 30/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 30 | Loss 0.00974
Epoch 31/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 31 | Loss 0.00981
Epoch 32/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 32 | Loss 0.00924
Epoch 33/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 33 | Loss 0.00953
Epoch 34/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.48it/s]
Epoch 34 | Loss 0.00992
Epoch 35/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 35 | Loss 0.00929
Epoch 36/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 36 | Loss 0.00920
```

```
Epoch 37/50: 100%|██████████| 1280/1280 [08:35<00:00,  2.48it/s]
Epoch 37 | Loss 0.00932
Epoch 38/50: 100%|██████████| 1280/1280 [08:35<00:00,  2.48it/s]
Epoch 38 | Loss 0.00980
Epoch 39/50: 100%|██████████| 1280/1280 [08:35<00:00,  2.48it/s]
Epoch 39 | Loss 0.00955
Epoch 40/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 40 | Loss 0.00899
Epoch 41/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 41 | Loss 0.00927
Epoch 42/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 42 | Loss 0.00960
Epoch 43/50: 100%|██████████| 1280/1280 [08:37<00:00,  2.47it/s]
Epoch 43 | Loss 0.00888
Epoch 44/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 44 | Loss 0.00884
Epoch 45/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 45 | Loss 0.00960
Epoch 46/50: 100%|██████████| 1280/1280 [08:35<00:00,  2.48it/s]
Epoch 46 | Loss 0.00934
Epoch 47/50: 100%|██████████| 1280/1280 [08:35<00:00,  2.48it/s]
Epoch 47 | Loss 0.00921
Epoch 48/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 48 | Loss 0.00928
Epoch 49/50: 100%|██████████| 1280/1280 [08:35<00:00,  2.48it/s]
Epoch 49 | Loss 0.00947
Epoch 50/50: 100%|██████████| 1280/1280 [08:36<00:00,  2.48it/s]
Epoch 50 | Loss 0.00911
```



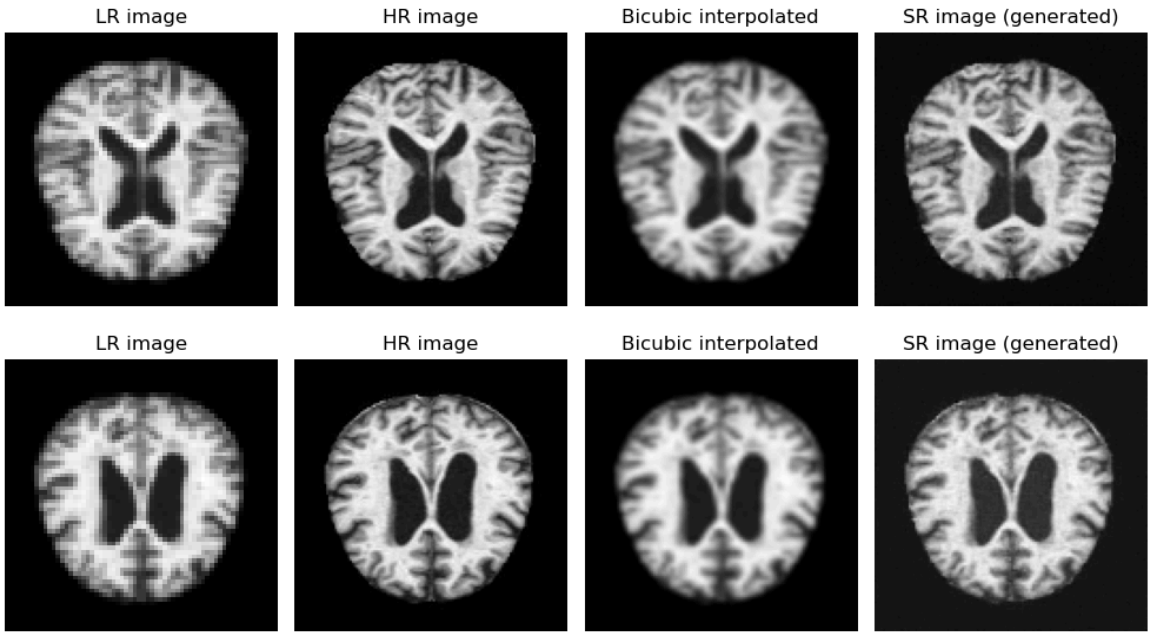
1.3. Run the inference to produce super-resolved images

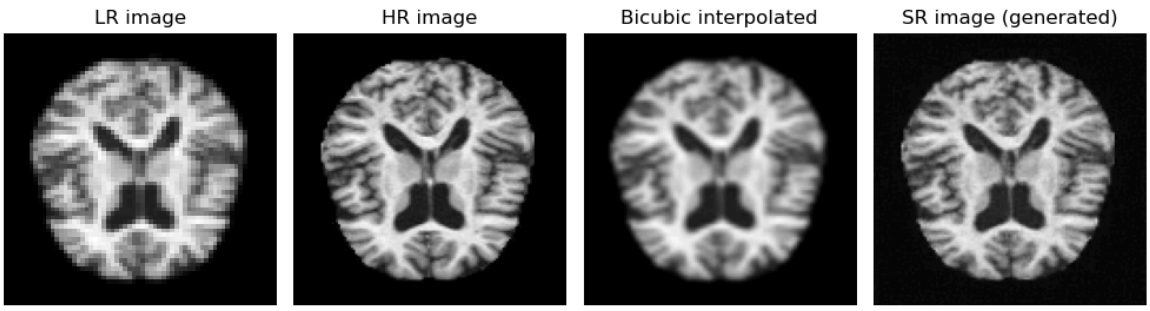
In [7]:
```
# Load some images from dataset to make inference
crop_size = 128
num_images = 10  # number images to load
# Load and log the first 10 elements from the DataLoader
# lrs - Low-resolution images list
# hrs - Hi-resolution images list
# clss - List of ground true class of images
lrs, hrs, clss, _ = helpers.image_loader(
    image_dir=test_image_dir,
    num_images=num_images,
    crop_size=crop_size,
    scale_factor=scale_factor,
)
```

In [8]:
```
# Images to show (up to 10)
num_images = 3
assert len(lrs) >= num_images, 'Up to {len(lrs)} images can be shown!'
# loading saved model weight
# model = get_sr_model_state('ddpm_checkpoints/ddpm_checkpoint_sr_1750339897.472253', num_cla
model = get_sr_model_state(ddpm_checkpoint_file, num_classes=3)
# creating scheduler
scheduler = get_scheduler(num_steps)
generated_images = []
# Get "null" class for guideless inference
null_class = torch.tensor([DataClasses.get_null_class_index(DataClasses)])
# generating image examples (will take some time)
for idx in range(num_images):
    generated_image = generate_image(128, lrs[idx], null_class, model, scheduler, num_steps)
    generated_images.append(generated_image)
    cuda.empty_cache()
```

1.4 Show images side-by-side: low-res, hi-res, bicubic upscaled low-res and super-res

In [9]:
```
# display generated SR images
for idx in range(num_images):
    helpers.show_images(lrs[idx], hrs[idx], generated_images[idx].squeeze(0))
```





2. Classificator model training process

2.1. Set up training hyperparameters

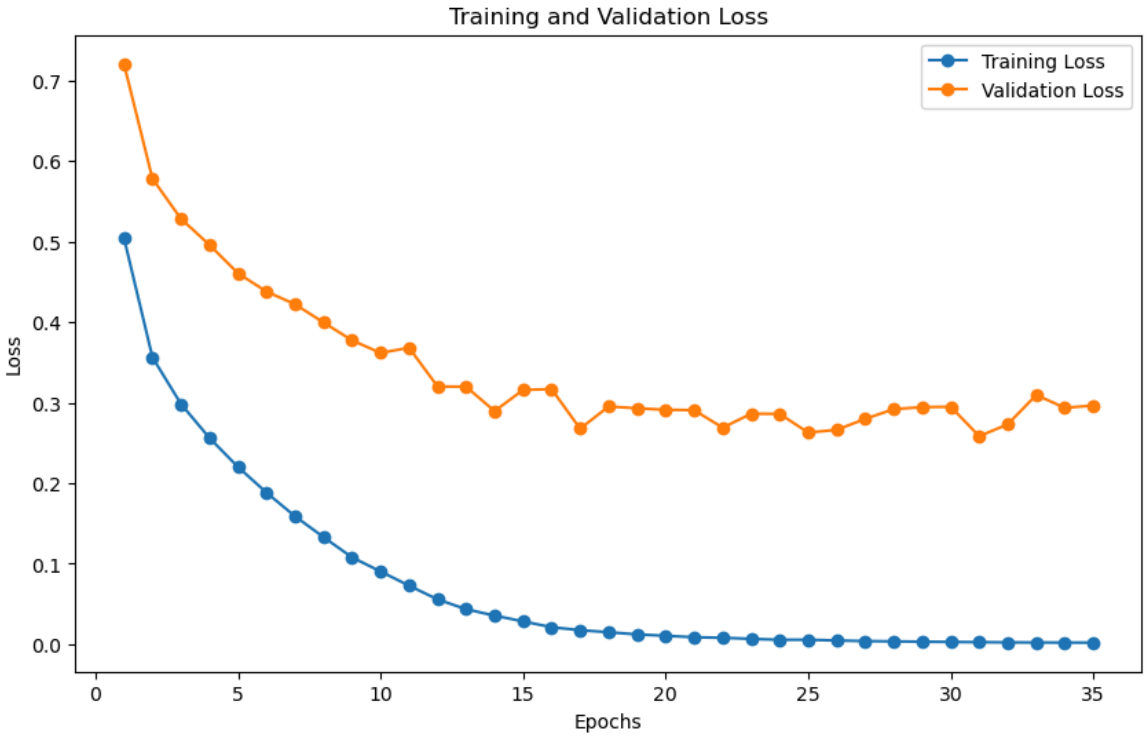
In [10]:
```
cuda.empty_cache()
# Hyperparameters
batch_size = 256
learning_rate = 0.00001
num_epochs = 35 # 35 is optimal for alzheimer dataset
crop_size = 128
```

2.2. Running the training process

In [11]:
```
classificator.trainer(
    train_image_dir=image_dir,
    val_image_dir=test_image_dir,
    num_epochs=num_epochs,
    batch_size=batch_size,
    crop_size=crop_size,
    learning_rate=learning_rate,
    device=device,
    file_to_save=classificator_checkpoint_file,
)
```

```
No pre-trained model found or path not specified. Starting training from scratch.
Epoch [1/35], Train Loss: 0.5048, Train Accuracy: 0.7621, Valid Loss: 0.7209, Valid Accuracy:
0.5246,
Epoch [2/35], Train Loss: 0.3564, Train Accuracy: 0.8597, Valid Loss: 0.5783, Valid Accuracy:
0.6943,
Epoch [3/35], Train Loss: 0.2985, Train Accuracy: 0.8758, Valid Loss: 0.5285, Valid Accuracy:
0.7279,
Epoch [4/35], Train Loss: 0.2563, Train Accuracy: 0.8971, Valid Loss: 0.4961, Valid Accuracy:
0.7584,
Epoch [5/35], Train Loss: 0.2203, Train Accuracy: 0.9149, Valid Loss: 0.4608, Valid Accuracy:
0.7725,
Epoch [6/35], Train Loss: 0.1884, Train Accuracy: 0.9313, Valid Loss: 0.4381, Valid Accuracy:
0.7881,
Epoch [7/35], Train Loss: 0.1590, Train Accuracy: 0.9445, Valid Loss: 0.4224, Valid Accuracy:
0.8108,
Epoch [8/35], Train Loss: 0.1331, Train Accuracy: 0.9610, Valid Loss: 0.3997, Valid Accuracy:
0.8147,
Epoch [9/35], Train Loss: 0.1080, Train Accuracy: 0.9696, Valid Loss: 0.3775, Valid Accuracy:
0.8233,
Epoch [10/35], Train Loss: 0.0905, Train Accuracy: 0.9772, Valid Loss: 0.3619, Valid Accuracy:
0.8444,
Epoch [11/35], Train Loss: 0.0728, Train Accuracy: 0.9863, Valid Loss: 0.3683, Valid Accuracy:
0.8389,
Epoch [12/35], Train Loss: 0.0558, Train Accuracy: 0.9907, Valid Loss: 0.3202, Valid Accuracy:
0.8671,
Epoch [13/35], Train Loss: 0.0436, Train Accuracy: 0.9949, Valid Loss: 0.3200, Valid Accuracy:
0.8593,
Epoch [14/35], Train Loss: 0.0356, Train Accuracy: 0.9960, Valid Loss: 0.2896, Valid Accuracy:
0.8765,
Epoch [15/35], Train Loss: 0.0285, Train Accuracy: 0.9971, Valid Loss: 0.3159, Valid Accuracy:
0.8694,
Epoch [16/35], Train Loss: 0.0212, Train Accuracy: 0.9991, Valid Loss: 0.3169, Valid Accuracy:
0.8726,
Epoch [17/35], Train Loss: 0.0176, Train Accuracy: 0.9991, Valid Loss: 0.2678, Valid Accuracy:
0.8968,
Epoch [18/35], Train Loss: 0.0150, Train Accuracy: 0.9993, Valid Loss: 0.2953, Valid Accuracy:
0.8866,
Epoch [19/35], Train Loss: 0.0123, Train Accuracy: 0.9999, Valid Loss: 0.2932, Valid Accuracy:
0.8882,
Epoch [20/35], Train Loss: 0.0106, Train Accuracy: 0.9999, Valid Loss: 0.2913, Valid Accuracy:
0.8905,
Epoch [21/35], Train Loss: 0.0088, Train Accuracy: 0.9999, Valid Loss: 0.2908, Valid Accuracy:
0.8944,
Epoch [22/35], Train Loss: 0.0081, Train Accuracy: 1.0000, Valid Loss: 0.2691, Valid Accuracy:
0.8866,
Epoch [23/35], Train Loss: 0.0068, Train Accuracy: 1.0000, Valid Loss: 0.2864, Valid Accuracy:
0.8874,
Epoch [24/35], Train Loss: 0.0056, Train Accuracy: 1.0000, Valid Loss: 0.2862, Valid Accuracy:
0.8913,
Epoch [25/35], Train Loss: 0.0056, Train Accuracy: 0.9999, Valid Loss: 0.2629, Valid Accuracy:
0.8952,
Epoch [26/35], Train Loss: 0.0047, Train Accuracy: 1.0000, Valid Loss: 0.2664, Valid Accuracy:
0.8937,
Epoch [27/35], Train Loss: 0.0040, Train Accuracy: 1.0000, Valid Loss: 0.2799, Valid Accuracy:
0.8991,
Epoch [28/35], Train Loss: 0.0036, Train Accuracy: 1.0000, Valid Loss: 0.2921, Valid Accuracy:
0.8991,
Epoch [29/35], Train Loss: 0.0032, Train Accuracy: 1.0000, Valid Loss: 0.2946, Valid Accuracy:
0.8984,
Epoch [30/35], Train Loss: 0.0029, Train Accuracy: 1.0000, Valid Loss: 0.2950, Valid Accuracy:
0.8968,
Epoch [31/35], Train Loss: 0.0027, Train Accuracy: 1.0000, Valid Loss: 0.2584, Valid Accuracy:
0.8991,
Epoch [32/35], Train Loss: 0.0024, Train Accuracy: 1.0000, Valid Loss: 0.2733, Valid Accuracy:
0.9101,
Epoch [33/35], Train Loss: 0.0022, Train Accuracy: 1.0000, Valid Loss: 0.3097, Valid Accuracy:
```

```
0.8976,
Epoch [34/35], Train Loss: 0.0021, Train Accuracy: 1.0000, Valid Loss: 0.2937, Valid Accuracy:
0.8984,
Epoch [35/35], Train Loss: 0.0020, Train Accuracy: 1.0000, Valid Loss: 0.2966, Valid Accuracy:
0.8976,
```



2.3. Run classificator to evaluate previously generated images

In [12]:
```
# Loading saved classificator's weights and instantiating classificator instance
# classificator_checkpoint_path = 'classifier_models\\classification_model1750175641.079124.p
model = classificator.get_model_state(classificator_checkpoint_file, num_classes=2)
# evaluating images previously generated by diffusion model in 1.3.
predicted_cls_idxs = []
for idx in range(num_images):
    predicted_cls_idxs.append(classificator.classify_image(model, generated_images[idx]))
for idx in range(num_images):
    print(f'Image true class: {clss[idx][0]} \tGenerated image predicted class: {predicted_cl
```

```
Image true class: 0     Generated image predicted class: 0
Image true class: 0     Generated image predicted class: 1
Image true class: 0     Generated image predicted class: 0
```

3. Experiment

3.1. Set parameters

In [13]:
```
# Paths to files - repeated for convenience
# Dataset locations
test_image_dir = 'C:/Users/novit/Documents/datasets/best_alzheimer_mri/test'
# Location of the generated SR images
gen_image_dir = 'C:/Users/novit/Documents/datasets/best_alzheimer_mri/gen'
# Location to save trained model's weights
ddpm_checkpoint_file = 'ddpm_checkpoints/ddpm_checkpoint_sr.pth'
classificator_checkpoint_file = 'classificator_checkpoints/classificator_checkpoint.pth'
# Location to store statistical data
stats_data_file = 'stats/stat_data.json'
```

3.2. Run experiment to generate and classify images

In [ ]:
```
# Generate SR images from the test LR ones and save them
generate(test_image_dir, gen_image_dir, ddpm_checkpoint_file)
```

```
source_image_path:  C:/Users/novit/Documents/datasets/best_alzheimer_mri/test
Number of images:               1279
Number of images to generate:   1279
```

In [19]:
```
stats_data = classify_and_calculate(test_image_dir, gen_image_dir, classificator_checkpoint_f
```

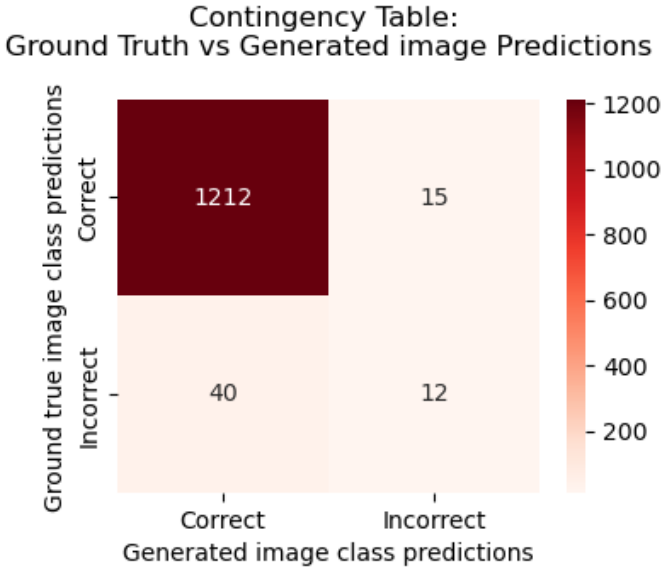
```
Processing test dataset: 100%|██████████| 1279/1279 [00:25<00:00, 50.76it/s]
```

3.3. Display achieved metrics

In [20]:
```
# Read saved statistical data object
try:
    stats_data
except NameError:
    stats_data = helpers.load_json(stats_data_file)

# unpack statistical data from data object
mcnemar_stats = stats_data['mcnemar_stats']

base_stats_data = stats_data['base_stats_data']
is_correct_predictions = stats_data['is_correct_predictions']

# true image classes
image_classes = base_stats_data['image_classes']
# predicted true image class
ground_truth_predictions = base_stats_data['ground_truth_predictions']
# predicted generated image class
generated_predictions = base_stats_data['generated_predictions']

# Two sets of images predicted classification
is_correct_ground_true_predictions = is_correct_predictions['is_correct_ground_true_predictio
is_correct_generated_predictions = is_correct_predictions['is_correct_generated_predictions']
```

3.3.1 Contingency table for McNemar test

In [22]:
```
importlib.reload(helpers)
ct_df = helpers.show_conf_matrix(
    is_correct_ground_true_predictions,
    is_correct_generated_predictions,
    title='Contingency Table: \nGround Truth vs Generated image Predictions\n',
    indexes=['Correct', 'Incorrect'],
    columns=['Correct', 'Incorrect'],
    xlabel='Generated image class predictions',
    ylabel='Ground true image class predictions',
    color='Reds'
)
```



In [23]:
```
# Access the values from the DataFrame
a = ct_df.loc['Correct', 'Correct']
b = ct_df.loc['Correct', 'Incorrect']
c = ct_df.loc['Incorrect', 'Correct']
d = ct_df.loc['Incorrect', 'Incorrect']
# Printing variables for calculate McNemar chi2
print(f"a: {a}")
print(f"b: {b}")
print(f"c: {c}")
print(f"d: {d}")
```

```
a: 1212
b: 15
c: 40
d: 12
```

3.3.2 McNemar chi-squared and p-value

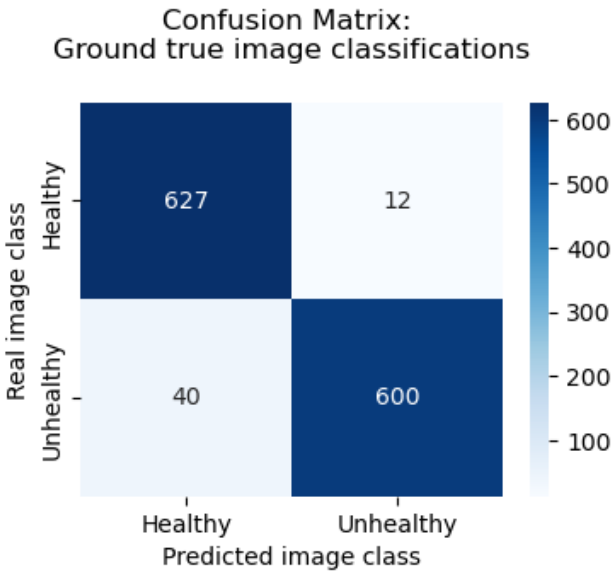
In [24]:
```
# Perform the McNemar test
result = (b-c)**2/(b+c)

# Table dimensionality
rows, cols = ct_df.shape

# Calculate degree of freedom
degree_of_freedom = (rows - 1)*(cols - 1)

# Calculate p-value from the chi-squared statistic
p_value = chi2.sf(result, df=degree_of_freedom)

print("McNemar's Test Results:")
print(f"Chi-squared statistic: \t{result:.3f}")
print(f"P-value: \t\t\t\t{p_value}")
```

```
McNemar's Test Results:
Chi-squared statistic:  11.364
P-value:                                0.0007489604476389001
```

3.3.3 Confusion matrix and metrics for Ground true image classifications

In [25]:
```
true_cm_df = helpers.show_conf_matrix(
    image_classes,
    ground_truth_predictions,
    title='Confusion Matrix: \nGround true image classifications\n',
```

```
    indexes=['Healthy', 'Unhealthy'],
    columns=['Healthy', 'Unhealthy'],
    xlabel='Predicted image class',
    ylabel='Real image class',
)
```



In [26]:
```
# Access the values from the DataFrame
TN = true_cm_df.loc['Healthy', 'Healthy']
FP = true_cm_df.loc['Healthy', 'Unhealthy']
FN = true_cm_df.loc['Unhealthy', 'Healthy']
TP = true_cm_df.loc['Unhealthy', 'Unhealthy']

print(f"True Positives (TP): \t{TP}")
print(f"False Negatives (FN): \t{FN}")
print(f"False Positives (FP): \t{FP}")
print(f"True Negatives (TN): \t{TN}")
```

```
True Positives (TP):    600
False Negatives (FN):   40
False Positives (FP):   12
True Negatives (TN):    627
```

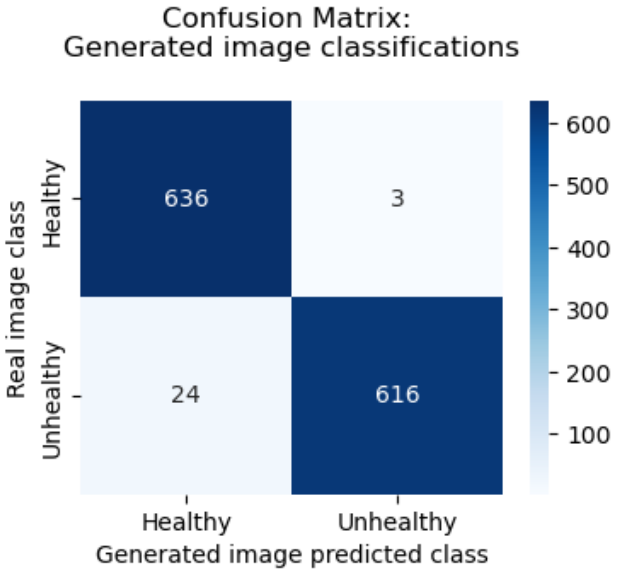
In [27]:
```
accuracy = (TP+TN)/(TP+TN+FP+FN)
precision = TP/(TP+FP)
recall = TP/(TP+FN)
FNR = FN/(TN+FP)

print(f"True image classification metrics:")
print(f"  Accuracy: \t{accuracy:.3f}")
print(f"  Precision: \t{precision:.3f}")
print(f"  Recall: \t\t{recall:.3f}")
print(f"  FNR: \t\t\t{FNR:.3f}")
```

```
True image classification metrics:
  Accuracy:     0.959
  Precision:    0.980
  Recall:               0.938
  FNR:                  0.063
```

3.3.4 Confusion matrix and metrics for Generated image classifications

In [28]:
```
# Generated image classification confusion matrix
gen_cm_df = helpers.show_conf_matrix(
    image_classes,
```

```
    generated_predictions,
    title='Confusion Matrix: \nGenerated image classifications\n',
    indexes=['Healthy', 'Unhealthy'],
    columns=['Healthy', 'Unhealthy'],
    xlabel='Generated image predicted class',
    ylabel='Real image class',
)
```



In [29]:
```
# Access the values from the DataFrame
TN = gen_cm_df.loc['Healthy', 'Healthy']
FP = gen_cm_df.loc['Healthy', 'Unhealthy']
FN = gen_cm_df.loc['Unhealthy', 'Healthy']
TP = gen_cm_df.loc['Unhealthy', 'Unhealthy']

print(f"True Positives (TP): \t{TP}")
print(f"False Negatives (FN): \t{FN}")
print(f"False Positives (FP): \t{FP}")
print(f"True Negatives (TN): \t{TN}")
```

```
True Positives (TP):    616
False Negatives (FN):   24
False Positives (FP):   3
True Negatives (TN):    636
```

In [30]:
```
accuracy = (TP+TN)/(TP+TN+FP+FN)
precision = TP/(TP+FP)
recall = TP/(TP+FN)
FNR = FN/(TN+FP)

print(f"Generated image classification metrics:")
print(f"  Accuracy: \t{accuracy:.3f}")
print(f"  Precision: \t{precision:.3f}")
print(f"  Recall: \t\t{recall:.3f}")
print(f"  FNR: \t\t\t{FNR:.3f}")
```

```
Generated image classification metrics:
  Accuracy:     0.979
  Precision:    0.995
  Recall:               0.963
  FNR:                  0.038
```

Annex B: Code Repository

The complete source code and associated materials for this thesis are publicly available on GitHub. You can access the repository at the following link: `https://github.com/artofnext/master_thesis` or using QR code shown on Figure 5.1.

FIGURE 5.1. QR-encoded link to the code repository