UTRECHT UNIVERSITY
Faculty of Science
Department of Information and Computing Sciences
MSc Artificial Intelligence

TRADITIONAL AUGMENTATION VERSUS DEEP GENERATIVE DIFFUSION AUGMENTATION FOR ADDRESSING CLASS IMBALANCE IN CHEST X-RAY CLASSIFICATION

A THESIS BY
Xinhao Lan
1082620

Project supervisor Prof. dr. Albert Salah
Daily supervisor Evi M.C. Huijben
Second examiner Dr. Itir Onal Ertugrul

Utrecht University

Abstract

Medical image analysis has advanced rapidly with the integration of deep learning techniques. However, the challenge of unbalanced datasets and the need for effective preprocessing methods remain significant difficulties in achieving optimal classification performance. This thesis aims to investigate the effectiveness of various image dataset augmentation techniques and the potential of diffusion models for chest x-ray image classification, focusing mainly on the data unbalance problem.

After obtaining a high quality dataset using various data and image preprocessing methods, we used traditional data augmentation methods such as rotation, flipping, blurring, and contrast modification to increase the number of positive class samples. In addition to traditional augmentation methods, several diffusion models are introduced to synthesize new chest x-ray images to further strengthen the minority class and address data imbalance. The performance of these methods was evaluated using specific metrics and compared to established baseline models.

The results showed that while replacing labels and masking images can introduce errors, the selected combination of preprocessing methods showed promise in improving classification performance. These results also indicated that traditional data augmentation methods, after careful fine-tuning of the hyperparameters, achieved significant performance improvements over the original baseline model. The application of diffusion models further improves the final classification results. Moreover, the images generated by the diffusion models, compared to traditional augmentation methods, do not merely modify the original images, but introduce some new image information, leading to improvements in various metrics.

Our results demonstrate the importance of augmentation methods in addressing data imbalance and improving the results of chest X-ray image classification. The research describes the most important image generation techniques that yield superior classification results while overcoming the hurdles of imbalanced datasets. These findings have profound implications for the medical field and machine learning specialists, signaling a promising path for improving diagnostic accuracy and patient care in chest X-ray image analysis. While current methods show potential, further research, particularly in the areas of stable diffusion models and deep learning-based image classification, is needed to make significant advances in the field.

Table of Contents

1. Introduction

Medical image classification holds significant potential in aiding the diagnostic and therapeutic procedures for various pathologies. The use of deep learning has clearly improved the accuracy of these classification tasks. However, the introduction of deep learning does not adequately address the challenges posed by data scarcity (Bansal et al., 2022) and low-quality labels (Wang et al., 2021), but also the class imbalance problem in certain datasets can detrimentally affect the outcomes of binary or multi-label classification tasks (Bria et al., 2020).

To tackle this issue, multiple augmentation methods have been developed to expand the minority class's size and achieve balanced class distribution in the dataset (Johnson and Khoshgoftaar, 2019). In medical image analysis, traditional image transformation methods, such as randomly cropping, rotating, and flipping horizontally and vertically have been utilized to produce numerous images from a single one to correspond to varying patient conditions (Chlap et al., 2021). Nonetheless, such methods frequently only present supplementary data to deep models' training and introduce the variation without significant modifications or new image information, and the pathological information of produced images may not reflect the realistic situation. This has resulted in deep learning being commonly employed in generative models. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), Variational Autoencoder (VAE) (Kingma and Welling, 2013), Denoising Diffusion Probability Model (DDPM) (Ho et al., 2020) and other deep generative models have been extensively applied to increase dataset size and mitigate class imbalance issues, leading to superior performance as compared to traditional data augmentation techniques. Recent studies (Dhariwal and Nichol, 2021) have demonstrated that the DDPM is superior to GANs in generating images, and have suggested revised versions of this model like Denoising Diffusion Implicit Model (DDIM) (Song et al., 2020).

This study aims to examine the impact of augmented data obtained from traditional augmentation methods and two deep generative diffusion models on the binary classification performance for medical image datasets with significant class imbalance issues. The performance of these augmentation techniques was compared and assessed for their impact on the accuracy of binary classification in medical imaging. The findings offer valuable insights into the effectiveness of these augmentation techniques in addressing class imbalance and enhancing the performance of binary classification in medical imaging.

1.1 Research questions

This thesis will research the generation of an unbalanced medical image dataset. The thesis will aim to answer the following research question:

How do augmented data obtained from traditional augmentation methods and deep generative diffusion models affect the performance of binary classification for medical image datasets with significant class imbalance?

In order to answer the research question above, we answer several sub-questions:

- **Question 1:** What type of dataset should be selected and what preprocessing methods should be used to ensure that the processed dataset is of high quality?
- **Question 2:** Which types of data augmentation methods should be implemented in the experiment and how will they influence the performance of the classification task?
- **Question 3:** What are the fine-tuning experiments for the augmentation and classification models supposed to be and which parameter needs to be optimized?
- **Question 4:** Which state-of-the-art classification model should be implemented and which one is suitable for our tasks?
- **Question 5:** What evaluation metrics should we choose to evaluate the results of the different augmentation methods, and what would be the expected effect of the diffusion model on those evaluation metrics.

For the question above, we have the following hypotheses:

- **Hypothesis 1:** The widely used medical image datasets include X-ray, MRI and CT image datasets. Since the images in the X-ray image dataset are more correlated with the physical world and have stronger visibility, we think using the X-ray image dataset is better for our task. We think the uncertain class labels in the file and the information outside the human body in the image can make the classifier less effective. our assumption is uncertain class labels should be interpreted as positive samples because that will strengthen the classifier and getting rid of the bias introduced by the information outside the body could also make the classifier more robust.
- **Hypothesis 2:** Data augmentation is generally divided into traditional data augmentation and deep learning-based data augmentation. We expect all those traditional augmentation methods still cannot avoid the impact of class imbalance on the final downstream task results. However, unlike the traditional data augmentation methods,

they only perform shape transformations or add noise. Diffusion models have the ability to overcome limitations in the availability of data, as they can generate new samples to fill in missing parts of the data distribution. We expect deep generative models should be able to generate better images of a specific class than the traditional method based on the generation evaluation metrics. For the diffusion models, some state-of-the-art models usually do not have enough open-source codes. So, we expect to use the original denoising probabilistic diffusion model and one improved version in our experiment.

- **Hypothesis 3:** For the classification model, we need to optimize some common hyperparameters such as the learning rate, number of epochs and batch size. For each method of the traditional augmentation methods, their hyperparameters are almost unique. We should fine-tune the hyperparameters such as angle or intensity. In addition, for some common hyperparameters, the filling method should have a large impact on the final classification results.
- **Hypothesis 4:** There are a number of neural networks that can be used for 2D image classification, we intend to use the most popular networks such as DenseNet and Inception-Resnet. we also intend to do the experiment with the classification network that is specifically for chest x-rays. We expect that the Denset can perform best on the binary classification.
- **Hypothesis 5:** We expect that synthetic images obtained from the diffusion model will show higher similarity to the real dataset in terms of luminance, contrast and structure compared to the traditional augmentation methods. Furthermore, we also expect that the synthetic image after the diffusion models can perform better on the downstream classification task.

1.2 Contributions of the thesis

Our contributions of this thesis can be summarised as follows:

1. **Data and Image Preprocessing:**
   - We used various strategies for label substitution, especially when faced with labels like 'Not Mentioned' or 'Uncertain'. This exploration led us to consider the potential implications and benefits of certain substitutions over others.
   - We used different mask methods to verify the influence of irrelevant features on the final result and considered using an inverted joint mask before the final classification.
2. **Classification Networks Comparison:**
   - We compared different classification networks and found that DenseNet was particularly effective for our dataset, outperforming others like Inception-

ResNet and CheXNet. This could help others working on similar projects decide which network to use.

3. **Study of Traditional Augmentation Methods:**
    - We deeply analyzed different traditional augmentation methods to understand how they can make models stronger. This study provided a detailed look into how each method and hyper-parameter affects the final results.
4. **Improved Augmentation Strategies:**
    - Our work involved experimenting with a range of augmentation strategies, highlighting the value of augmenting images that were previously classified incorrectly. This approach showed promise in specific cases, offering ways to improve the final classification results.
5. **Exploration with Diffusion Models:**
    - We dived into diffusion models to see how they compared to usual image augmentation techniques. This included a look at both basic and stable diffusion models, spotlighting the importance of matching the right model with the right dataset and the potential of fine-tuning for better results.

2. Related Work

2.1 Medical image analysis with deep learning methods

Medical image analysis tasks include several crucial classes such as image segmentation, image registration and image classification. With the development of deep learning techniques, those three tasks have emerged as the predominant domain in this area.(Shen et al., 2017).

- **Image segmentation:** Medical image segmentation is essential as it isolates and delineates specific anatomical structures or regions of interest, enabling precise diagnosis, treatment planning, and disease monitoring. Convolutional Neural Network (CNN) is a class of deep learning models widely used for medical image segmentation. The fundamental concept behind this is to train a CNN to generate a corresponding output image with pixel-level labels, signifying the class or segmentation of each pixel in the input image. Among the deep learning-based models for medical image segmentation, the U-Net architecture proposed by Ronneberger et al. (2015) is one of the most popular. The U-Net model is composed of an encoder path and a decoder path that are connected by skip connections. The encoder path includes multiple convolutional and pooling layers that progressively reduce the input image size and extract high-level features. The decoder path comprises upsampling and convolutional layers that increase the resolution of the feature maps and generate the segmentation mask. The skip connections enable the decoder to incorporate low-level features from the encoder, thereby enhancing segmentation accuracy. Since the original U-Net, several modifications and extensions have been proposed to further improve its performance. For example, the attention U-Net (Oktay et al., 2018) introduced attention gates to the skip connections, allowing the network to focus on relevant image regions during segmentation. The U-Net++ (Zhou et al., 2018) further extended the skip connections and introduced a nested U-Net architecture, which helps to capture multi-scale contextual information. Other deep learning models that have been used for medical image segmentation include DeepLab (Chen et al., 2017), Fully Convolutional Networks (FCN) (Long et al., 2015), and Mask R-CNN (He et al., 2017).
- **Image classification:** Medical image classification involves categorizing images into different classes based on their content automatically for accurate disease diagnosis, monitoring treatment efficacy, and streamlining clinical workflows. It plays a pivotal role in enhancing diagnostic precision, tailoring personalized treatments,

and facilitating efficient case studies and comparisons in vast medical databases. Deep learning models have been widely used for this task and have shown promising results. These models can automatically learn features from medical images and classify them into different categories. Some popular CNN architectures for medical image classification include VGG, ResNet, and Inception (Kora et al., 2022). These models have been used for a variety of tasks, such as identifying diseases in X-ray images, detecting tumors in magnetic resonance imaging (MRI) scans, and classifying skin lesions in dermatology. In addition to traditional CNNs, other deep learning models such as recurrent neural networks (RNNs) and attention-based models have also been applied to medical image classification tasks (Kim et al., 2021).

- **Image registration:** Medical image registration is the process of aligning two or more medical images of the same patient acquired from different modalities or at different times. It is crucial in medical imaging as it aligns multiple images to a common spatial domain, enabling the fusion of information from different modalities or time points. This alignment facilitates accurate disease monitoring, aids in interventional procedures, and enhances treatment planning, ensuring optimal patient care and outcomes. Deep learning-based methods have been applied to medical image registration tasks due to their ability to learn and model complex non-linear relationships between images. One common approach is to use a convolutional neural network to learn a similarity metric between images and then use an optimization algorithm to find the optimal registration parameters that minimize the metric. Another approach is to use a convolutional neural network to estimate the transformation parameters directly from the images themselves. Balakrishnan et al. (2019) proposed a network called VoxelMorph, which is a CNN-based registration method that learns a spatial transformation between images using an unsupervised approach. This method is capable of handling large deformations and has demonstrated state-of-the-art performance in multiple medical image registration tasks. Chen et al. (2018) proposed VoxResNet, which is a 3D CNN-based registration method that uses a residual network to model the deformation field between two images. It can handle large deformations and shows good performance. Çiçek et al. (2016) proposed Deformable U-Net, which is a modification of the popular U-Net architecture that includes a deformable convolutional layer to allow more flexible modeling of the deformation field.

2.2 Traditional data augmentation

Image augmentation in medical image analysis addresses the challenges of limited or unbalanced datasets. By transforming existing data, augmentation methods can improve model performance by enhancing dataset variety. Basic image augmentation involves geo-

metric transformations, intensity operations, noise injection, filtering, occlusion, random cropping, and mixup methods. However, the utility of each technique must be carefully assessed in the context of medical imaging.

- **Geometric Transformations:** Geometric transformations modify the spatial arrangement of pixels in an image, making them a type of image augmentation method. New images with different positions, orientations, scales, and aspect ratios can be created using geometric transformations (Kozlov, 2000). These methods offer variations in image orientation, scale, and aspect ratio, and are valuable for medical image analysis. Barber and Hose (2005) utilized geometric transformations for automatic medical image segmentation, demonstrating their potential.
- **Intensity Operations:** Intensity operations are techniques that manipulate the pixel intensities of an image without altering its overall structure. They can enhance image contrast, remove noise, or emphasize specific features. Medical image data augmentation utilizes intensity operations to generate new images by modifying the existing image's intensity values. Augmenting the training data can increase its variability and improve the performance of machine learning models. Huang et al. (2021) used histogram equalization and intensity scaling to augment MRI brain images for the task of tumor segmentation. Zeng et al. (2015) employed gamma correction, contrast stretching, and histogram equalization to augment mouse brain images for gene expression pattern annotation. Perez et al. (2018) utilized histogram equalization, contrast stretching, and intensity scaling to augment skin lesion images for melanoma classification.
- **Noise Injection:** Noise injection is a data augmentation method that adds a specific amount of noise to the image data in order to improve the model's robustness against noise in the input data. Zhao et al. (2019) proposed a novel data augmentation technique based on learned image transformations. One of the learned transformations is called "NoiseInjection", which adds Gaussian noise with a random noise level to the input image. Dalca et al. (2019) proposed a new unsupervised deep learning framework for brain MRI segmentation. Gaussian noise injection is utilized as one of the data augmentation techniques to enhance the model's robustness against noise in the input data.
- **Filtering:** Filtering can be used as a data augmentation technique to introduce variability in the appearance of images while preserving their underlying structure as well. Gaussian blurring is a concept introduced by Fukunaga and Hostetler (1975), and the Laplacian pyramid, a method for decomposing an image into a series of Gaussian-blurred and downsampled versions, is a concept introduced by Burt (1983). The first method involves reducing the high-frequency components of an image, such as noise or small details, to create a smoother version of the original image.

Sharpening is the process of improving the high-frequency components of an image, such as edges or details, to make the image look sharper. Filters that emphasize high-frequency information, such as a Laplacian filter, can be used to achieve this by highlighting the areas where the intensity of the image changes rapidly. Li et al. (2018) applied Gaussian filtering as a data augmentation method to smooth the training images and reduce the impact of noise and artifacts in the analysis of CT images. Zhang et al. (2021) augmented the training data for a deep learning model that segments the pancreas in abdominal CT scans by using blurring and sharpening filters. Their findings show that this technique enhances the robustness and accuracy of the model by reducing the impact of noise and artifacts in the input images.

- **Occlusion:** Occlusion simulates scenarios where parts of an image are obscured, training models to recognize obscured or concealed structures. This technique has been noted to boost object recognition performance in various computer vision applications. The relevance to medical imaging is the improved recognition of partially obscured anatomical structures. Bearman et al. (2016) proposed a straightforward and efficient technique for producing occluded images. The study found that deep learning models trained on the resulting dataset of occluded images showed considerable enhancement in object recognition performance, particularly while evaluating the models on test images with occlusions that were not present in the training data. Kompanek et al. (2019) utilized perturbed normalization, translation, scaling, rotation, salt, and pepper noise, along with occlusion, to augment the original image, resulting in improved performance.
- **Cropping and Mixup:** Random cropping is a data augmentation technique that involves randomly selecting a portion of an image and using only that portion for training. Random cropping is applied to medical images to increase the amount of training data and improve the robustness of segmentation models (Long et al., 2015). The random cropping method is also applied in the medical image analysis task by Wodzinski et al. (2020) and Stefan et al. (2017). Furthermore, Zhang et al. (2017) proposed a technique named 'mixup' that creates novel training data points by blending pairs of examples and their labels. The methodology relies on a combination method that introduces a data augmentation technique that merges two or more original images to create a new one. Nishio et al. (2020) utilized a technique called Random Image Cropping and Patching (RICAP). Unlike the previous method, RICAP performs random cropping of four original images, patches them together, and generates a new image.
- **Deformable Augmentation Techniques:** Deformable augmentation techniques may be utilized when basic augmentation techniques fail to provide enough variability to create a generalizable subsequent model. The deformation scale is typically limited within user-specified parameters to maintain the clinical plausibility of the

resulting augmentations (Chlap et al., 2021). Simard et al. (2003) proposed a new approach to enhance the 2D image's geometric shape by using a randomly generated displacement field method. In this method, each pixel of the image is shifted randomly in both the horizontal and vertical directions. The value of each shift is chosen randomly from a uniform distribution which belongs to the range $(-1, 1)$. Then, the resulting displacement fields, $\triangle x$ and $\triangle y$, are subject to convolution using a Gaussian kernel. The degree of smoothness of the deformation can be modified by adjusting the standard deviation, σ, of the convolution kernel. Using intermediate values of σ will create a smoother, more elastic deformation. The randomly generated displacement field method can be an effective technique for data augmentation, thus enhancing the generalization capacity of CNNs in visual analysis. This method has been applied to augment medical images for various tasks in different areas such as MRI, CT, and X-ray by Javaid et al. (2019), Zhang et al. (2020), and Novosad et al. (2020). They have reported improved performance compared to using only original images.

- **Spline interpolation:** Spline interpolation is a mathematical technique that uses a piecewise polynomial function to estimate values between existing data points (Schoenberg, 1964). This approach facilitates the creation of smooth and deformed images for generating new image data in the context of deformable image augmentation. Various tasks, including MRI, CT, and X-ray, have seen favorable downstream task outcomes with this method, as employed by Rigaud et al. (2021), Kim et al. (2019), and Sandfort et al. (2019). Moreover, Statistical Shape Models (SSMs) are extensively utilized in medical image analysis to capture and represent the anatomical structure variability of a given population (Cootes et al., 1995). Corral Acero et al. (2019) and Bhalodia et al. (2018) show that this method can be applied to medical image analysis and effectively enhance the diversity of medical datasets.

Traditional data augmentation techniques hold significance in medical image analysis, improving model accuracy and robustness. The primary distinction between basic data augmentation techniques and other methods is their lack of focus on producing lifelike images. Certain geometric transformations, such as scaling, translation, or noise injection, seem realistic due to their ability to simulate images of patients with different sizes or positions or the creation of noisy images. While some methods directly benefit medical imaging by simulating real-life clinical scenarios, others may not always be suitable and must be judiciously applied.

2.3 Deep generative data augmentation

Deep generative models have improved data augmentation, especially in the area of medical image analysis, providing innovative solutions to data scarcity and ensuring better results by generating high-fidelity, diverse samples. GANs, VAE, flow-based models and diffusion models are the most commonly used deep learning networks for data augmentation (Figure 1). These models learn the underlying distribution of the training data and produce new samples that resemble the actual data. The main advantage of this approach is that it allows the creation of an almost infinite number of new samples without the need for manual annotation. This can be particularly useful in medical imaging, where collecting large amounts of labeled data can be difficult or costly.

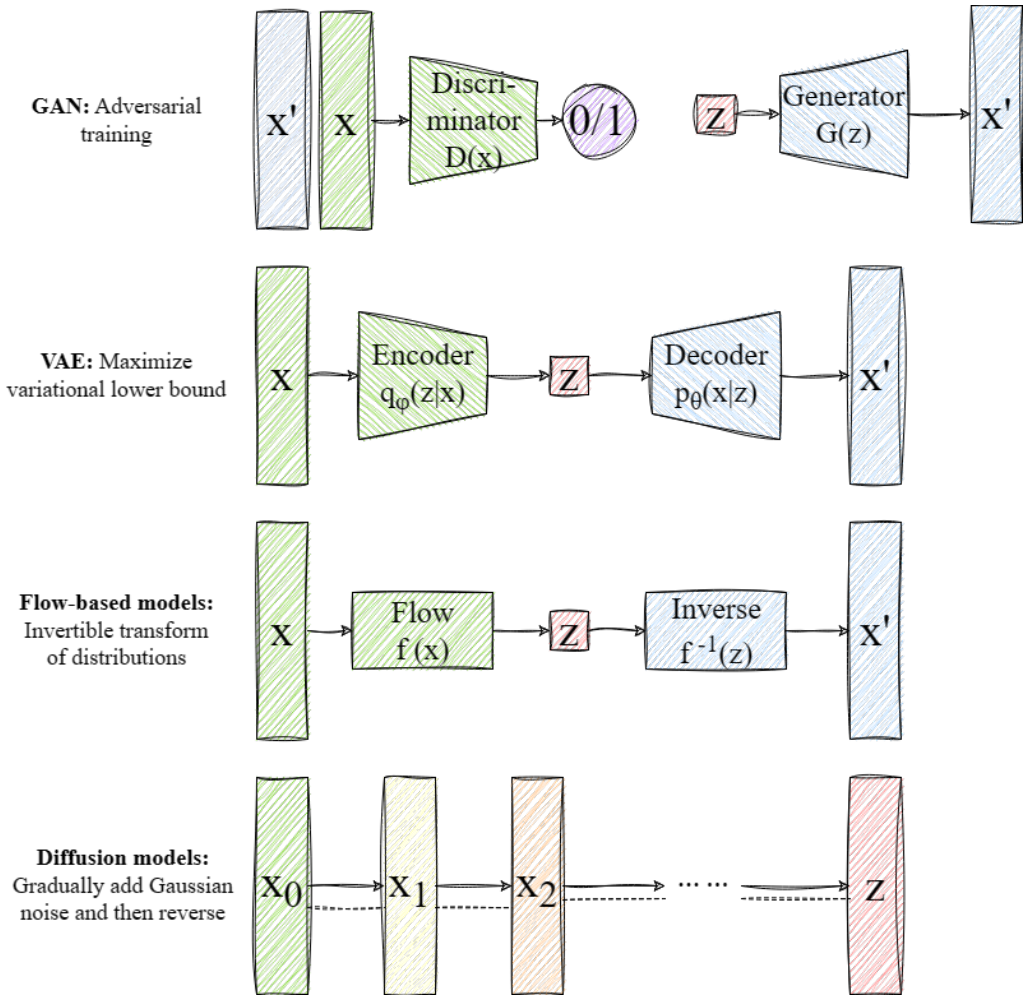


Figure 1. Four classes of the deep learning-based augmentation methods. Sketch is based on https://lilianweng.github.io/posts/2021-07-11-diffusion-models

- **GAN:** GANs consist of two neural networks: a generator and a discriminator. The generator network learns to generate new data by mapping samples from the noise distribution to the target data distribution. On the other hand, the discriminator

network learns to distinguish between real and synthetic data samples. During the training process, the generator generates synthetic samples while the discriminator network tries to distinguish between real and synthetic samples. The generator is updated to produce better synthetic samples by tricking the discriminator network. Similarly, the discriminator is updated to improve its ability to detect the differences between real and synthetic samples. Through this iterative process, the generator network learns to produce new synthetic samples that are similar to the authentic data set. In medical image processing, GAN-based data augmentation has been used to synthesize medical images for various applications, including segmentation, classification, and detection. GAN has been used to generate synthetic CT images (Zhu et al., 2018), MRI images (Waheed et al., 2020), and ultrasound images (Pang et al., 2021) to augment the limited amount of real medical images and improve the performance of medical image analysis tasks.

- **VAE:** VAE is a type of generative model that learns to encode and decode a high-dimensional input space. It consists of two parts: an encoder network that maps input data to a latent space, and a decoder network that maps latent representations back to the original data space (Kingma and Welling, 2013). The model is trained to minimize the discrepancy between the input and reconstructed data. By generating new samples in the latent space and decoding them to obtain synthetic images, the VAE model can be used for data augmentation purposes.
- **Flow-based models:** The work of Pesteie et al. (2019) introduced a generative model based on VAE for augmentation in image classification and segmentation tasks. Unlike other generative models such as GANs and VAEs, flow-based models use invertible functions to transform a simple random noise distribution, such as Gaussian, into a distribution that approximates the target data distribution. This process simplifies sampling from the target data distribution, allowing new images to be generated with similar statistical characteristics to the original data set. Kingma and Dhariwal (2018) proposed an alternative distribution mapping technique compared to the flow-based generative model for colorectal polyp synthesis in CT colonography.
- **Diffusion models:**

Ho et al. (2020) proposed the denoising diffusion probability model (DDPM), which consists of two parametric Markov chains. The main function of the model is to use variational inference to generate samples that have the same distribution as the original data after a specific time. The forward chain is responsible for perturbing the data by adding Gaussian noise to it gradually according to the pre-designed noise progression until the distribution of the data approaches a prior distribution. On the other hand, the reverse chain starts from the given prior state, uses a parameterized Gaussian transformation kernel, and gradually restores the distribution of the original data. Nichol and Dhariwal

(2021) introduced the modifications to the basic DDPM model, including the learning of variance, alterations to the loss function, and the incorporation of cosine noise, all of which significantly improved its performance.

Song et al. (2020) proposed an alternative diffusion model called Denoising Diffusion Implicit Models (DDIM). In contrast to the DDPM, DDIM does not constrain the diffusion process to be a Markov chain, enabling it to use smaller sampling steps to accelerate the generation process. Furthermore, the process of introducing random noise to generate similar artifacts in this model is certain. Dhariwal and Nichol (2021) demonstrated that reducing the diversity of images can produce high-quality images generated by the GAN model. Furthermore, the GAN model requires accurate parameter selection and a large amount of data. These limitations constrain the GAN model from performing effectively in downstream applications. This paper also demonstrated that diffusion models perform better than GANs in generating high-quality images and covering sample distributions. The authors conducted multiple experiments to determine the optimal architectures for the diffusion models. These experiments include decreasing the number of channels, increasing model depth and attention heads, applying the attention module at various resolutions, implementing the residual module of BigGAN for upsampling and downsampling, increasing the number of channels per head, and applying adaptive group normalization.

Classifier-guided diffusion models have some disadvantages. Firstly, it requires additional calculations. Moreover, the guide function and the diffusion model are trained separately, making it difficult to expand the model scale and achieve better results through joint training. Ho and Salimans (2022) proposed a replacement structure to substitute the external classifier, allowing the direct use of a diffusion model for conditional generation tasks. The content of the model input was modified. There are two types of sampling inputs available: conditional, which comprises random Gaussian noise and guidance information embedding, and unconditional. Both inputs are inputted into the same diffusion model, making it possible to generate it unconditionally and conditionally. The former method for updating noise is 2.1 while the latter method is 2.2.

$$\epsilon_\theta(x_t, t) \sim \epsilon_\theta(x_t) - \sqrt{1-\overline{\alpha}_t} \bigtriangledown_{x_t} log p_\phi(y|x_t) \tag{2.1}$$

$$\widehat{\epsilon_\theta}(x_t|y) = \epsilon_\theta(x_t) + s \cdot (\epsilon_\theta(x_t, y) - \epsilon_\theta(x_t)) \tag{2.2}$$

Additionally, Nichol et al. (2021) utilized a mask image and text token as the condition and input to the diffusion model to obtain a superior result. According to Equation 2.3, the noise should be modified. They utilized CLIP to replace the traditional classifier. This approach can be trained using a noised image x_t, and the corresponding loss function is

presented in Equation 2.4.

$$\widehat{\epsilon}_\theta(x_t|Caption) = \epsilon_\theta(x_t) + s \cdot (\epsilon_\theta(x_t, Caption) - \epsilon_\theta(x_t)) \tag{2.3}$$

$$\widehat{\mu}_\theta(x_t|c) = \mu_\theta(x_t|c) + s \cdot \Sigma_\theta(x_t|c) \bigtriangledown_{x_t} (f(x_t) \cdot g(c)) \tag{2.4}$$

A new model named DALLE-1 was proposed by Ramesh et al. (2021), which incorporates a vision transformer, Contrastive Language-Image Pre-Training (CLIP), and a discrete variational autoencoder. The model's architecture can be seen in Figure 2. During the training process, images were used to obtain the image tokens with the help of a dVAE. Afterwards, text tokens were obtained using a caption and a text encoder. Finally, the image and text tokens were combined and fed into the Transformer model, with the image tokens mapped into the text tokens vocabulary. For the generation process, obtain text tokens from a caption using the encoder and image tokens using the transformer. The trained image decoder generates images using the image tokens. Since image generation involves sampling, the generated images are sorted using the CLIP model. The image with the highest similarity to the text features is chosen as the final output of the generation process.

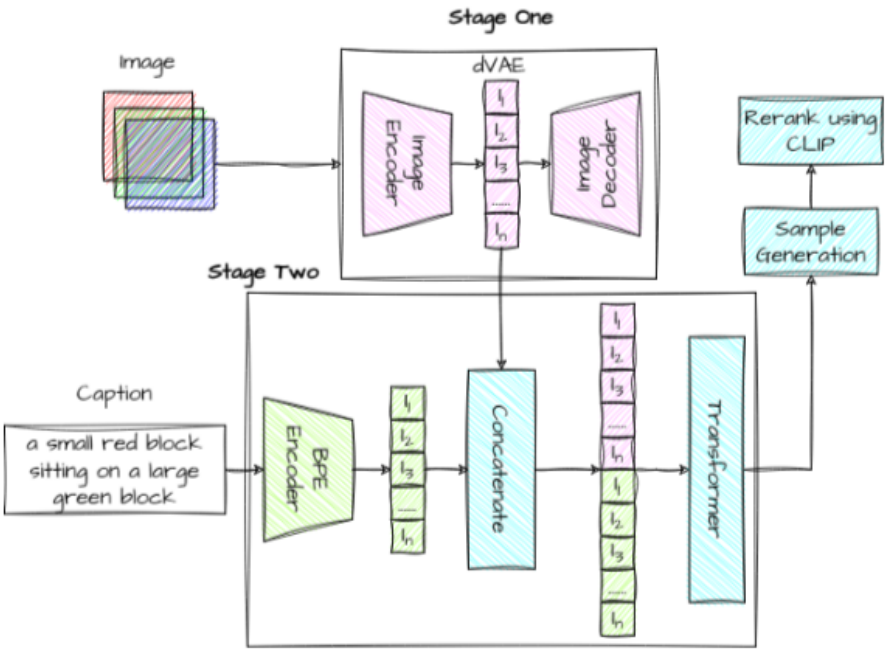


Figure 2. The structure of the DALLE-1 model. Sketch is based on Ramesh et al. (2021)

Ramesh et al. (2022) proposed the DALLE-2 model, which follows this approach. The model includes CLIP, a prior network, and the diffusion model, as seen in Figure 3. In this model, the prior network converts text into text tokens, while the decoder converts text tokens into images. The prior network is trained to align text tokens with image

tokens during the training process. Both autoregressive and diffusion models are used, but the diffusion model performs better in this case. The model can also divide the hidden representation of the image. The image feature can be separated based on the CLIP image embedding (Z) and the feature at the time of sampling by DDIM (X_T). This hidden encoding enables highly precise reconstruction and additional editing of images. For instance, by fixing Z while adjusting X_T, the model can generate an image with a semantically similar image style as the original, but differing in certain details. Similarly, by setting a fixed X_T during the interpolation of Z, the model can achieve a desired output.

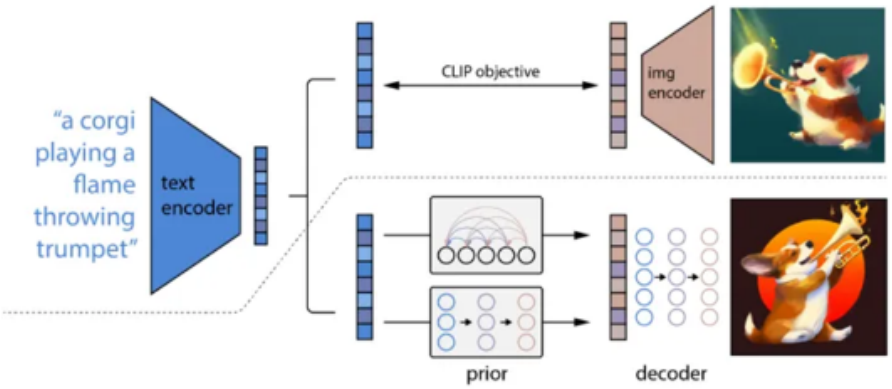


Figure 3. The structure of the DALLE-2 model. Graph is reprinted from Ramesh et al. (2022)

Saharia et al. (2022) proposed the Imagen model (Figure 4), which comprises a diffusion model and two super-resolution networks. They found it crucial to use a dynamic threshold during the inference stage to clip the results generated at different time steps. This threshold is dynamically adjusted to ensure optimal results. Without the dynamic threshold, the model tends to oversaturate the generated results due to the large guide gradient. Dynamic predictions lead to more realistic results from the model. Additionally, this paper discovered that large pre-trained models provide more efficient text encoding compared to CLIP.

Following this, the Imagic model was proposed by Kawar et al. (2022) (Figure 5). Initially, the target text is encoded and the initial text embedding e_{tgt} is obtained, after which it is optimized to reconstruct the input images resulting in e_{opt}. Subsequently, the generative model is fine-tuned to improve fidelity to the input image with a fixed e_{opt}. Lastly, the final editing result is generated by interpolating e_{opt} with e_{tgt}. The generation of different faces on the VAE model is similar to replacing the diffusion model with a simple encoder and decoder. Nevertheless, the feature space and generation ability of the diffusion model are superior to those of the VAE.

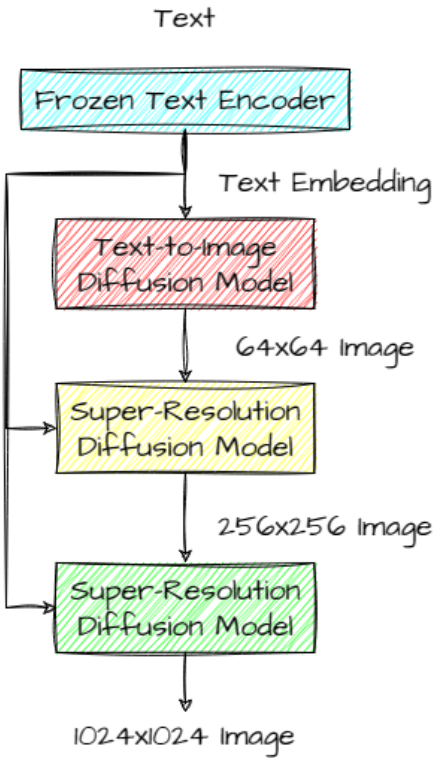


Figure 4. The structure of the Imagen model. Graph is reprinted from Saharia et al. (2022)

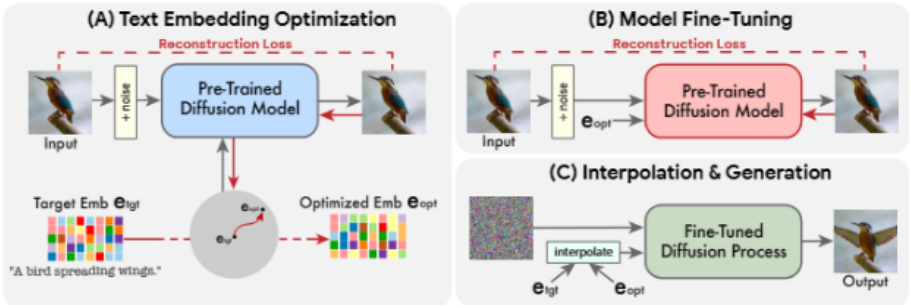


Figure 5. The structure of the Imagic model. Sketch is based on Kawar et al. (2022)

In the article by Rombach et al. (2022), the stable diffusion model is introduced. The model employs an encoder to compress the image, performs the diffusion operation on the latent space, and then utilizes the decoder to restore the image. The method proposed in the article significantly lowers the computational complexity of the diffusion process on the latent space. Additionally, a cross-attention approach is suggested to achieve multimodal training. This method successfully completed several image-generation tasks, such as class-condition image generation and text-to-image generation.

3. Methodology

3.1 Dataset and tools

The most commonly used datasets for Chest X-ray analysis include CheXpert (Irvin et al., 2019), MIMIC-CXR (Johnson et al., 2019) and ChestXray 14 (Summers, 2019) datasets. We decided to use the CheXpert dataset in our experiment finally. First, to obtain the MIMIC-CXR dataset, it is necessary to complete relevant courses and obtain authorization from MIT beforehand. It is hard for us to get the authorization in such short time. Second, compared to the CheXpert dataset, the ChestXray dataset includes many errors and misclassified images [1].

The CheXpert dataset is a large collection of X-ray images and their labels which are created by their paired radiology reports. All the X-ray image data in this dataset is obtained from Stanford Hospital. This dataset consists of 224,316 X-ray images (both front and lateral images) obtained from 65,240 patients, with each X-ray image labeled as 'Positive', 'Negative', 'Uncertain', or 'Not Mentioned' for each of the 14 observations (Table 1). An automatic labeling system was developed to reduce the cost of asking experts to label. The Labeling system's principle is to extract features manually and label them using text recognition and semantic analysis on radiology reports and paired images. The validation dataset has 200 chest X-ray images of 200 patients. Three experts labeled these images instead of the Labeling system used for training. The labels given should be either present, uncertain likely, uncertain unlikely, or absent. In the given labels, present and uncertain likely should be considered positive, whereas absent and uncertain unlikely should be considered negative. The test dataset consists of 500 chest X-ray images labeled by five experts, three of whom have already labeled the validation dataset. Observations with a positive mention in the report get a positive (1) label. Those with an uncertain mention, but no positive ones, are labeled uncertain (u). If there's a negative mention, it's labeled negative (0). Absent mentions are labeled as not mentioned (NM). An observation of "No Finding" is labeled positive (1) only if no pathologies are classified as positive or uncertain. To determine the feasibility of observation extraction, the authors manually reviewed a set of 1000 reports (Irvin et al., 2019), which were evaluated by a board-certified radiologist.

[1] https://laurenoakdenrayner.com/2017/12/18/the-chestxray14-dataset-problems/

Pathology	Positive (%)	Uncertain(%)	Negative(%)	NM (%)
No Finding	22,381 (10.02)	0 (0.00)	0 (0.00)	201,033 (89.98)
Enlarged Cardiom	10,798 (4.83)	12,403 (5.55)	21,638 (9.69)	178,575 (79.93)
Cardiomegaly	27,000 (12.09)	8,087 (3.62)	11,116 (4.98)	177,211 (79.31)
Lung Opacity	105,581 (47.26)	5,598 (2.51)	6,599 (2.95)	105,636 (47.28)
Lung Lesion	9,186 (4.11)	1,488 (0.67)	1,270 (0.57)	211,470 (94.65)
Edema	52,246 (23.38)	12,984 (5.81)	20,726 (9.28)	137,458 (61.53)
Consolidation	28,097 (12.58)	27,742 (12,42)	14,783 (6.61)	152,792 (68.39)
Pneumonia	6,039 (2.70)	18,770 (8.40)	2,799 (1.25)	195,806 (87.65)
Atelectasis	33,376 (14.94)	33,739 (15.10)	1,328 (0.59)	154,971 (69.37)
Pneumothorax	19,448 (8.70)	3,145 (1.41)	56,341 (25.22)	144,480 (64.67)
Pleural Effusion	86,187 (38.58)	11,628 (5.20)	35,396 (15.84)	90,203 (40.38)
Pleural Other	3,523 (1.58)	2,653 (1.19)	316 (0.14)	216,922 (97.09)
Fracture	9,040 (4.05)	642 (0.29)	2,512 (1.12)	211,220 (94.54)
Support Devices	116,001 (51.92)	1,079 (0.48)	6,137 (2.75)	100,197 (44.85)

Table 1. Data distribution (number of images and their percentage of the whole dataset) of CheXpert dataset

Regarding the label of uncertainty. The authors of this dataset proposed five different methods for dealing with it:

- **U-Ignore model**: A simple approach to handling uncertainty is to ignore the *u* labels during training, which serves as a baseline to compare approaches that explicitly incorporate the uncertainty labels.
- **U-Zeros and U-ones**: These two methods map all the instances of *u* to 0 (U-Zeroes model) or 1 (U-Ones model) respectively. If the uncertainty label does provide semantically useful information to the classifier, this approach may perturb the decision-making of classifiers and impair their performance.
- **U-SelfTrained**: This method initially trains a model to convergence using the U-Ignore approach, which ignores the u labels during training. Subsequently, the model is employed to predict and relabel each of the uncertainty labels with the probability prediction produced by the model. The method does not replace any instances of 1s or 0s. Then, on the relabeled examples, a loss function is set up as the mean of the binary cross-entropy losses over the observations.

- **U-MultiClass**: This method tries to consider the *u* label as its own class rather than mapping it to a binary label for each of the 14 observations. This approach outputs the probability of each of the 3 possible classes $\{p_0, p_1, p_u\} \in [0, 1]$, $p_0 + p_1 + p_u = 1$. This method also sets the loss as the mean of the multi-class cross-entropy losses over the observations.

For the original dataset, there are only training dataset and validation dataset, we followed the previous method Yuan et al. (2021) to split the training dataset into a training dataset and a test dataset with a ratio of 0.8 (179453): 0.2 (44863) first. Then, we used the splited training dataset and real test dataset on GitHub [2] for our experiment. The presence of NM labels and -1 labels in the test and validation sets will introduce a large bias in the final classification results. So, only positive and negative labels are in these datasets. Among all the 14 label classes, we chose pneumothorax and pleural effusion for the experiment since they are similar diseases and pneumothorax data is quite unbalanced while pleural effusion's labels are far more balanced. The sample distributions of positive and negative types in the test set and validation set are shown in the Table 2 below:

Disease	Pneumothorax		Pleural Effusion	
Dataset	Test	Validation	Test	Validation
Negative	658	226	548	167
Positive	10	8	120	67

Table 2. Number of images for final test and validation dataset

3.2 Data augmentation model

3.2.1 Traditional data augmentation

There are different types of conventional approaches to data augmentation:

- **Geometric Transformation:**
  - *Rotation:* The image can be rotated by a fixed angle, and the excess portions of the image are cropped while the empty portions are filled.
  - *Shearing:* Essentially, this is the distortion of an image so that the shape of the image appears to be skewed in a certain direction. This distortion is achieved by shifting each point in the image, either fixed or variable, based on a specified shear factor, while leaving the coordinates of an axis unchanged.

[2] https://github.com/rajpurkarlab/cheXpert-test-set-labels/tree/main

- *Translation:* This method refers to the process of moving an image either vertically, horizontally, or both, within a defined frame or canvas. This movement results in a displacement of the image content, while the empty space left by the displacement can be filled in various ways, often with zeros (making it black for grayscale images) or by wrapping the image.
- *Flipping:* Flipping involves flipping the image horizontally or vertically. This technique can cause problems with medical image datasets. For example, flipping a CT scan of the brain could result in loss of anatomical orientation, which could adversely affect model performance.
- *Cropping:* Cropping is a method used to eliminate the peripheral parts of an image, leaving only the central part or a random section.
- *Scaling:* Images can be enlarged (zoom in) or reduced (zoom out). If the image is enlarged, the final size will be larger than the original size, which requires us to crop the image. If the image is reduced, we need to add padding.

All the data augmentation methods above can be implemented by the function in `torchvision.transforms`. [3]

- **Noise Injection:**
  - *Gaussian noise:* Adding random variations to an image or signal with a Gaussian distribution is a commonly employed method to incorporate randomness to the data or to replicate real-world conditions, where noise is an intrinsic feature. This method is commonly used to incorporate randomness into the data or to simulate actual conditions where noise is inherent. In Python, the numpy library allows adding Gaussian noise to an image or signal, using the mean value and standard deviation value.
  - *Salt and pepper noise:* Impulse noise is caused by errors in image acquisition, transmission, or storage. Salt and pepper noise reduces the quality of the image by setting some pixels to their maximum or minimum intensity values. The modeling of this noise can be achieved by randomly changing some pixels to the highest or lowest intensity values in the image. The numpy library can be used for implementing this function in Python.
- **Filtering:**
  - *Bilateral filtering:* Bilateral filtering is a non-linear image smoothing technique that reduces noise while preserving edges. The main goal of bilateral filtering is to replace the intensity values of each pixel with a weighted average of the intensity values of the surrounding pixels. These weights are calculated based on two factors: the spatial distance between pixels and the difference in intensity between pixels. The proximity between pixels is taken into account to give higher weights to neighboring pixels, while the intensity difference

[3] https://pytorch.org/vision/stable/transforms.html

ensures that pixels with similar intensities are given more preference in the average calculation. To perform bilateral filtering on an image, the OpenCV library offers a function called `cv2.bilateralFilter` that can be used. [4]

3.2.2 Diffusion model

We utilized Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020)and Denoising Diffusion Implicit Models (DDIM) (Song et al., 2020) for the Diffusion models parts. State-of-the-art models such as GLIDE (Nichol et al., 2021) and DALLE-2 (Ramesh et al., 2022) are not implemented in our experiment due to their high computational demands during training.

- **Denoising Diffusion Probabilistic Models:** The model can be divided mainly into two parts: the forward diffusion process and the reverse denoising process (see Figure 6).

  Forward diffusion process: Given a data point sampled from a real data distribution

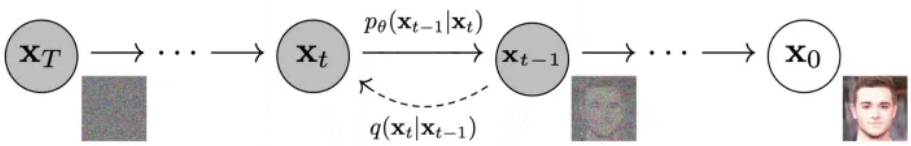


Figure 6. The Markov chain of forward (reverse) diffusion process of generating a sample. Graph is reprinted from Ho et al. (2020)

$\mathbf{x}_0 \sim q(\mathbf{x})$, add a small amount of Gaussian noise to the sample in T steps, producing a sequence of noisy samples $\mathbf{x}_1, \ldots, \mathbf{x}_T$. The step sizes are controlled by a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^T$. The Equations are shown as follows:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3.1}$$

The data sample $\mathbf{x}_0$ gradually loses its distinguishable features as step t becomes larger. Eventually when $T \to \infty$, $\mathbf{x}_T$ is equivalent to an isotropic Gaussian distribution.

Reverse denoising process: If we are able to gradually obtain the reversed distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, we can restore the original distribution $\mathbf{x}_0$ from the complete standard Gaussian distribution$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. It has been demonstrated that if $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ satisfies a Gaussian distribution and β_t is small enough, then $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ remains a Gaussian distribution. However, we cannot simply deduce $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, thus we

[4] https://docs.opencv.org/4.x/index.html

utilize a deep learning model with parameters θ to predict the inverse distribution p_θ.

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{3.2}$$

We are unable to obtain the reversed distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, but if $\mathbf{x}_0$ is known, $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ can be derived through the Bayesian formula.

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \tag{3.3}$$

So, for each time step, we use $\mathbf{x}_t$ and t to predict the Gaussian noise $\mathbf{z}_\theta(\mathbf{x}_t, t)$, and get the mean value based on the equation 3.4. Since the variance $\Sigma_t(x_t, t)$ in DDPM is equal to $\tilde{\beta}_t$ and $\tilde{\beta}_t$ is equal to $\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ which is similar to β_t, we can use the equation 3.1 to get $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, and use the reparameterization trick to get the $\mathbf{x}_{t-1}$.(Figure 7)

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{z}_\theta(\mathbf{x}_t, t)\right) \tag{3.4}$$

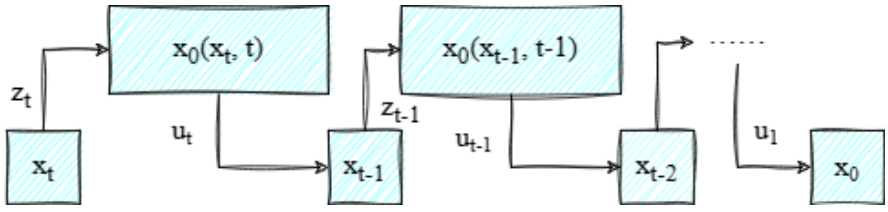


Figure 7. Calculation process of the reverse denoising process

The primary concept of the diffusion model is training a model to predict noise. Because the noise and original data share the same dimension, an AutoEncoder architecture may be chosen as the noise prediction model. DDPM adopts a U-Net model utilizing residual blocks and attention blocks (Figure 8). The U-Net is a

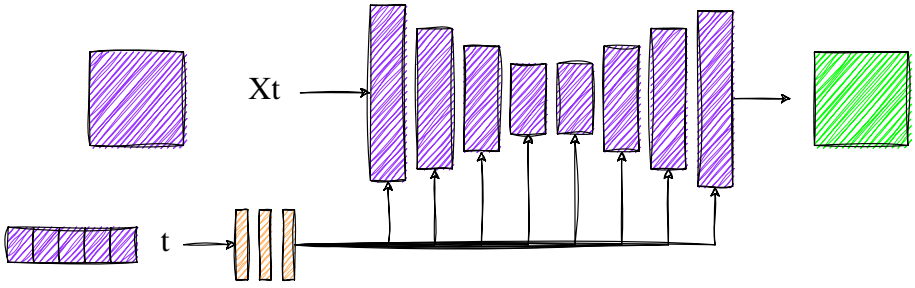


Figure 8. Structure of the U-Net in the DDPM

type of encoder-decoder architecture. The encoder is divided into different stages, each with a down-sampling module that reduces feature size. The decoder reverses this process, gradually restoring compressed features from the encoder. The U-Net

also includes skip connections in the decoder module, which concatenates features of the same dimensionality obtained from the encoder. This design is beneficial for network optimization. The U-Net has two residual blocks in each stage and self-attention modules in some stages to improve the network's global modeling ability. Additionally, the diffusion model requires T noise prediction models. In practice, we can use a time embedding (similar to the position embedding in the transformer) to encode the timestep in each residual block. The training process is shown in Figure 9.

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \mathbf{z}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Figure 9. Training and sampling process of the DDPM. Graph is reprinted from Ho et al. (2020)

- **Classifier-free Diffusion Models:** Instead of sampling in the direction of the gradient of an image classifier, classifier-free guidance mixes the score estimates of a conditional diffusion model and a jointly trained unconditional diffusion model. For the classifier-guidance model, the diffusion score $\epsilon_\theta(z_\theta, c) \approx -\sigma_\lambda \bigtriangledown_{z_\lambda} log\, p(z_\lambda|c)$is changed to $\epsilon_\theta(z_\theta, c) \approx -\sigma_\lambda \bigtriangledown_{z_\lambda} (log\, p(z_\lambda|c) + w log\, p_\theta(c|z_\lambda))$. For the DDPM model, the mean value of the Gaussian distribution is computed primarily from the results of the noise estimation model $\epsilon_\theta(x_t)$. We also include additional input conditions in the noise estimation model as $\epsilon_\theta(x_t, y)$. When training the diffusion model, both conditional and unconditional training methods are combined, with the condition y set to zero for unconditional training. This results in a model that supports both conditional and unconditional noise estimation. The advantage of this approach is that it incorporates additional input y during the training process, and in theory, the more input information, the easier it is to train. However, its disadvantage is also the introduction of additional input y during the training process, which means that any signal control requires retraining the entire diffusion model. The training and sampling process is shown in Figures 10 and 11.

3.3 Classification model

As we have described in Section 2.1, commonly used supervised deep learning classification models include DenseNet (Huang et al., 2017), ResNet (He et al., 2016), InceptionNet (Szegedy et al., 2015), U-Net (Ronneberger et al., 2015). We implemented the DenseNet, Inception-ResNet and CheXNet (Rajpurkar et al., 2017) in our experiment and compared the outcome of the classification task.

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: p_{uncond}: probability of unconditional training
repeat
  $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$ ▷ Sample data with conditioning from the dataset
  $\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond} ▷ Randomly discard conditioning to train unconditionally
  $\lambda \sim p(\lambda)$ ▷ Sample log SNR value
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$ ▷ Corrupt data to the sampled log SNR value
  Take gradient step on $\nabla_\theta \left\| \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon} \right\|^2$ ▷ Optimization of denoising model
until converged

Figure 10. Training process of the Classifier-free model. Graph is reprinted from Ho and Salimans (2022)

Algorithm 2 Conditional sampling with classifier-free guidance

Require: w: guidance strength
Require: $\mathbf{c}$: conditioning information for conditional sampling
Require: $\lambda_1, \ldots, \lambda_T$: increasing log SNR sequence with $\lambda_1 = \lambda_{\min}$, $\lambda_T = \lambda_{\max}$
$\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = 1, \ldots, T$ **do**
  ▷ Form the classifier-free guided score at log SNR λ_t
  $\tilde{\boldsymbol{\epsilon}}_t = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t)$
  ▷ Sampling step (could be replaced by another sampler, e.g. DDIM)
  $\tilde{\mathbf{x}}_t = (\mathbf{z}_t - \sigma_{\lambda_t} \tilde{\boldsymbol{\epsilon}}_t)/\alpha_{\lambda_t}$
  $\mathbf{z}_{t+1} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda_{t+1}|\lambda_t}(\mathbf{z}_t, \tilde{\mathbf{x}}_t), (\tilde{\sigma}^2_{\lambda_{t+1}|\lambda_t})^{1-v}(\sigma^2_{\lambda_t|\lambda_{t+1}})^v)$ if $t < T$ else $\mathbf{z}_{t+1} = \tilde{\mathbf{x}}_t$
end for
return $\mathbf{z}_{T+1}$

Figure 11. Sampling process of the Classifier-free model. Graph is reprinted from Ho and Salimans (2022)

- **DenseNet:** The basic idea of DenseNet (Figure 12) is consistent with that of ResNet (Figure 13), and its two features are: 1. Establishing dense connections between all the front layers and all the back layers. 2. Realizing feature reuse through feature connections in the channel. The output of the traditional network at layer l is $x_l = H_l(x_{l-1})$. For the ResNet, it adds the identity function of the previous layer input $x_l = H_l(x_{l-1}) + x_{l-1}$. For the DenseNet, all previous layers are connected as input $x_l = H_l([x_0, x_1, \ldots, x_{l-1}])$.

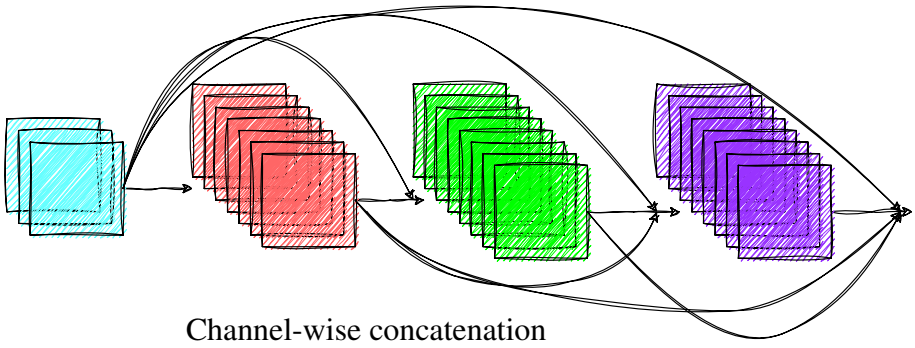


Figure 12. Dense connections structure of the DenseNet

The DenseNet network architecture primarily consists of the DenseBlock and the Transition. The feature map size of each layer in the DenseBlock is consistent

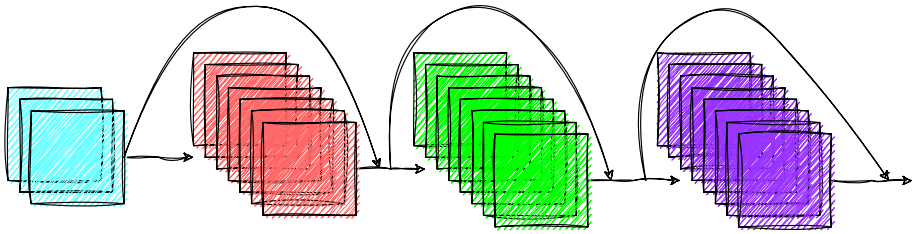


Figure 13. Connections structure of the ResNet

and can be connected along the channel dimension. The non-linear combination function $H(\cdot)$ in the DenseNet architecture is composed of Batch Normalization (BN), Rectified Linear Unit (ReLU), and a 3x3 Convolutional block (Figure 14). Contrary to ResNet, the number of output feature maps resulting from the convolutional operation in the DenseBlock is the growth rate parameter k. If the channel number of the feature map in the input layer is k_0, then the input channel number of the l-th layer equals $k_0 + k(l-1)$. Despite setting k to a low value, the input of the DenseBlock increases significantly as the number of layers grows. To reduce computation, the DenseBlock can incorporate a bottleneck block, which involves adding a 1x1 convolutional block to the original structure. Thus, the updated structure is composed of the following elements: BN, ReLU, 1x1 Convolutional block, BN, ReLU, and 3x3 Convolutional block (Figure 15).

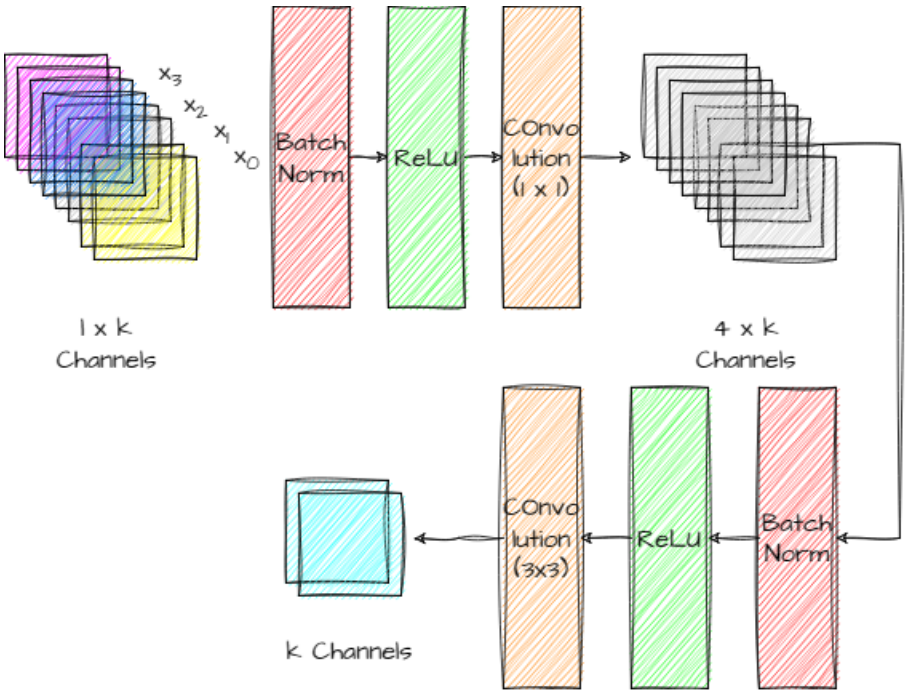


Figure 14. Structure of the DenseNet which includes the dense block

The Transition Layer connects two contiguous DenseBlocks and reduces the size of

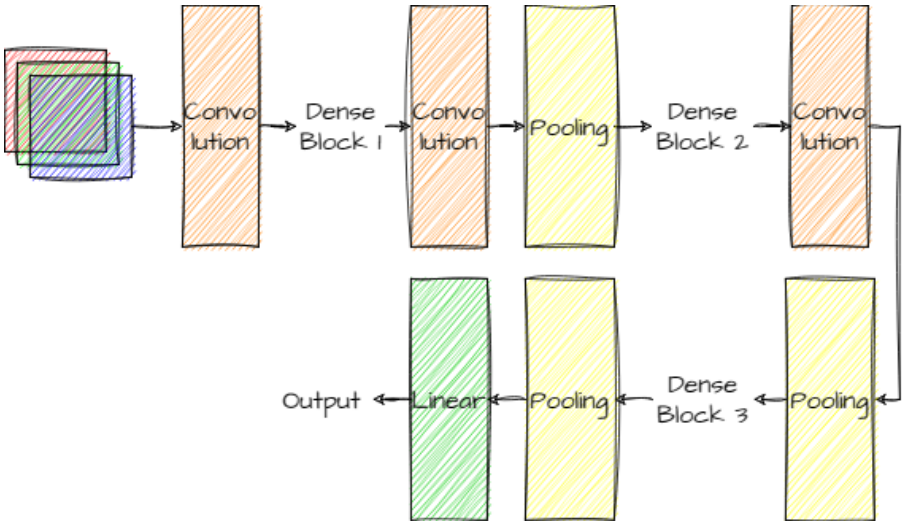


Figure 15. Structure of the DenseNet which includes the improved dense block

the feature maps. The layer consists of BN followed by ReLU, a 1x1 Convolutional block, and a 2x2 Average Pooling Block. Moreover, the Transition Layer compresses the model by reducing the number of feature maps when the compression rate θ is set.

- **Inception-ResNet:** The network combines residual connections and the Inception-v4 architecture and introduces residual scaling to stabilize the training process. The Inception-v4 architecture is formed by combining the Inception-v3 network with the Stem module, as depicted in Figure 16.

  The Inception-ResNet network differs from the original Inception network in the following three ways: Firstly, the structure of the Inception block used in the Inception-ResNet network is simpler. Secondly, in the Inception-ResNet network, a filter-expansion layer (1x1 convolution layer) is used to increase the number of channels that may be lost due to the Inception block. This is important as excessive dimension reduction can lead to representational bottleneck and loss of information. Thirdly, The Inception-ResNet architecture only implemented Batch Normalization in the stem module to reduce its storage consumption. Moreover, with this enhanced network, the residual network becomes unstable when the number of convolution kernels surpasses 1000. This instability is manifested in the output of zeros in the last layer before the average pooling layer, after several thousand iterations, ultimately leading to the network's failure. Neither reducing the learning rate nor increasing the Batch Normalization can resolve this issue. As a result, the network includes a residual scaling operation (Figure 17). This operation uses a reduction constant between 0.1 and 0.3 to decrease the Inception network output variance, improve its stability, and prevent model to be overfitting.
- **CheXNet:** The main structure of the CheXNet is the DenseNet121. However, the final fully connected layer of the network is replaced with a binary output and a

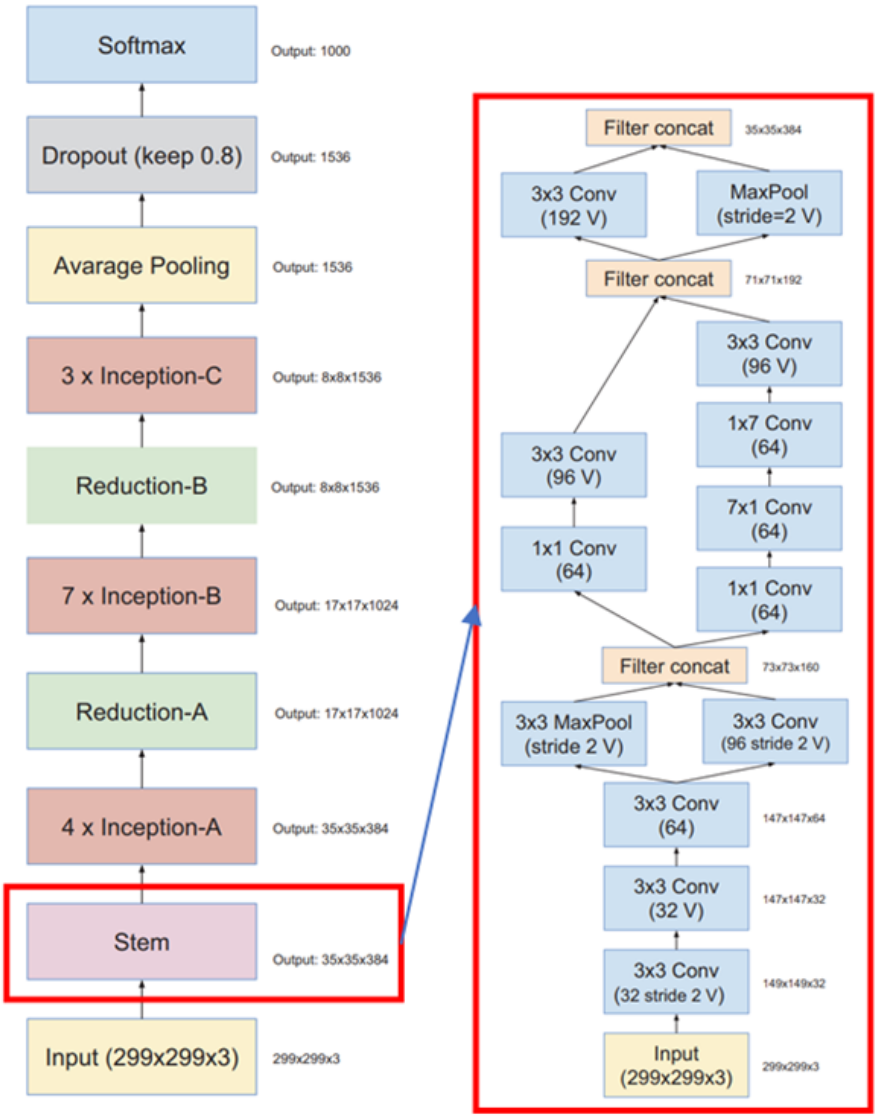


Figure 16. Structure of the Inception-ResNet-v2. Graph is reprinted from Szegedy et al. (2017)

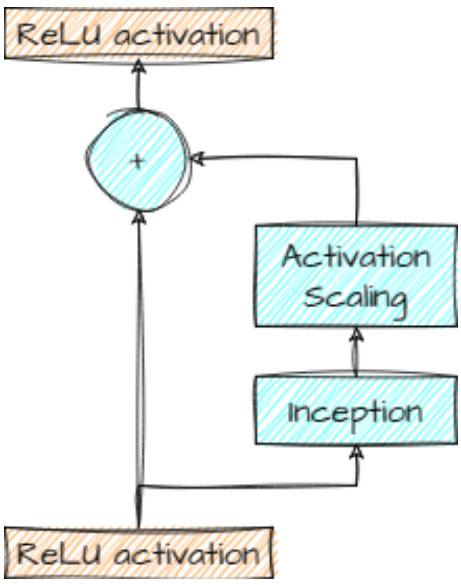


Figure 17. Structure of the residual scaling block. Sketch is based on Szegedy et al. (2017)

Sigmoid unit is connected to output the probability values. The CheXNet also uses a mini-batch of 16 and Adam gradient descent. Sometimes, the network extends the single output to a 14-dimensional output to get the prediction of different diseases simultaneously.

3.4 Evaluation

3.4.1 Image generation evaluation metric

There are three types of evaluation metrics that can be used in image generation: distribution-based metrics, pixel-based quality metrics and human-subjective evaluation metrics. We input the generated dataset and the real dataset into the evaluation metric function. And all the generated images are matched one by one with the corresponding real images to obtain the final value to calculate the average value.

- **Distribution-based metrics:** This metric type is based on the distribution, assessing the statistical properties of the produced images compared to a set of actual images. It compares the statistical properties of the produced images with those of actual images. Instead of emphasizing pixel-level details, these metrics analyze the higher-level characteristics and properties of the images. They aim to capture the distribution of overall features, such as color, object shapes, and textures. Distribution-based metrics frequently entail extracting features using pre-trained neural networks, followed by statistical analysis to compare distributions of real and generated images. Examples of distribution-based metrics are Fréchet Inception Distance (FID) and Inception Score (IS).
  - **IS (Inception Score)[5]:** The score specifically focuses on the output class label of the input data. The Inception score is made up of two components: the marginal likelihood of generated images and the diversity of the generated data. In order to optimize the first component, the probability distribution entropy value should be minimized. A lower value indicates a higher likelihood that the generated image belongs to a particular category and that the image quality is high. This value is determined by calculating Equation 3.5. The optimal value for the second component would be the largest possible average probability distribution entropy value. Equation 3.6 can be used to calculate this value.

$$E_{x\sim p_G}(H(p(y|x))) = \sum_{x\in G} P(x) \sum_{i=1}^{1000} P(y_i|x) \log \frac{1}{P(y_i|x)} \tag{3.5}$$

[5]https://github.com/openai/improved-gan/blob/master/inception_score/model.py

$$H(E_{x\sim p_G}(p(y|x))) = \sum_{x\in G} P(x) \sum_{i=1}^{1000} P(y_i|x) \log \frac{1}{P(y_i)} \tag{3.6}$$

When we combine both equations and obtain the equation for the Inception score (Equation 3.7), we can determine the quality of the generated image from the IS value. The higher the value of IS, the higher the quality of the generated image.

$$\text{IS} = \exp \sum_{x\in G} P(x) \sum_{i=1}^{1000} P(y_i|x) \log \frac{P(y_i|x)}{P(y_i)} = \exp E_{x\sim p_G} KL(p(y|x)||p(y)) \tag{3.7}$$

- **FID (Fréchet Inception Distance)[6]:** The FID metric compares feature distributions of real and generated images using a pre-trained Inception neural network in the scope of generative models. Feature vectors from images are extracted using the Inception neural network, and these vectors are then compared to assess the similarity between the distribution of real and generated images.

$$\text{FID}(x, g) = \|\mu_x - \mu_g\| + \text{Tr}\left(\Sigma_x + \Sigma_g - 2\sqrt{\Sigma_x \Sigma_g}\right) \tag{3.8}$$

A low FID score indicates that the generated images are of high quality and are similar to real images. Conversely, a high FID score indicates that the generated images are of low quality or significantly differ from real images.

- **Pixel-based quality metrics:** This type refers to the reconstruction metrics. This metric evaluates the quality of image reconstruction. These metrics compare the generated image to a reference or ground truth image, measuring the similarity between them. Typically, they focus on pixel-level or structural comparisons by evaluating factors such as luminance, contrast, edges, and textures. Popularly known reconstruction metrics include Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). These metrics provide a quantitative assessment of the level of similarity between the generated image and the reference image in terms of visual appearance.
  - **PSNR (Peak Signal to Noise Ratio)[7]:** The score is calculated by comparing the maximum possible power of a signal to the power of noise that corrupts the signal's quality. Peak Signal-to-Noise Ratio (PSNR) is defined as the ratio of the maximum power of a signal to the power of corrupting noise, expressed in decibels (dB).

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right) \tag{3.9}$$

[6] https://github.com/bioinf-jku/TTUR/blob/master/FIDvsINC/fid.py

[7] https://scikit-image.org/docs/stable/api/skimage.metrics.html#skimage.metrics.peak_signal_noise_ratio

MAX refers to the image's maximum pixel value, while MSE represents the mean squared error between the reconstructed and original images.
A high PSNR value indicates that the reconstructed image is of high quality and similar to the original image. Conversely, a low PSNR value means that the reconstructed image is of low quality and significantly different from the original image.

- **SSIM (Structure Similarity Index Measure)[8]:** The perceived quality of an image is related to the structural information present in the image, based on empirical observations. The SSIM index determines the structural similarity of two images by analyzing the luminance (Equation 3.10), contrast (Equation 3.11), and structure (Equation 3.12) information that they contain. One calculates the SSIM index by taking the average of SSIM values across multiple windows in the image.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{3.10}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{3.11}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{3.12}$$

The range of SSIM index values is between -1 and 1. A value of 1 indicates perfect similarity between the two images, while a value close to -1 shows significant differences between them. The SSIM index offers a reliable measure of how humans perceive image quality thanks to its consideration of the image's structural information, which is significant for human perception.

- **Human-subjective evaluation metrics:** The third category is not part of the mathematical evaluation criteria for deep learning. However, as with the datasets we employ, we have to ask experts to assess the final output results directly. For larger datasets, requiring experts to evaluate the results obtained from the test set or training set would consume a significant amount of time and money. Therefore, in our experiments, we did not consider using this evaluation method.

3.4.2 Image classification evaluation metric

The augmented dataset will be used for the classification task. The binary classification mainly employs the pneumothorax and pleural effusion label. Accuracy, precision, recall, F1 score, and ROC curve are typical evaluation metrics (Luque et al., 2019).

[8] https://scikit-image.org/docs/stable/api/skimage.metrics.html#skimage.metrics.structural_similarity

- **Confusion matrix:** The confusion matrix visually represents the performance of a classifier by depicting the number of predicted samples for each class. For a binary problem, it shows the true positive, true negative, false positive, and false negative predictions (Table 3).

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Table 3. Table of confusion matrix

- **Accuracy:** Accuracy refers to the ratio of correct predictions made by the classifier. For binary classification problems, accuracy can be calculated as:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \\ &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \end{aligned} \tag{3.13}$$

  The accuracy metric provides a comprehensive assessment of the classifier's performance; however, it may not always be a reliable metric, particularly if there is an imbalanced class distribution.
- **Precision:** Precision is a measure that evaluates the ratio of true positive predictions to all positive predictions made by a classifier. It indicates the number of positive predictions that the classifier makes that are actually correct. Precision is calculated using the following formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.14}$$

- **Recall:** Recall is a metric that quantifies the proportion of correctly predicted actual positive cases by a classifier. To calculate Recall, use the following formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.15}$$

- **F1 score:** The F1 score combines precision and recall into a unified metric that balances both measures. Precisely, the F1 score is calculated as the harmonic mean of precision and recall. Interpreted as the balance between a classifier's precision and recall, the F1 score is widely used as a single number metric to compare classifiers. The F1 score is especially valuable for severe class imbalance since it gives equal importance to both precision and recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.16}$$

- **ROC curve:** The ROC curve displays the performance of a binary classifier by plotting the true positive rate against the false positive rate at various thresholds. It's particularly useful for assessing classifiers in imbalanced datasets. The AUC provides a summarized numerical score of the ROC curve, often used for comparing classifiers.
- **Cohen's Kappa:** Cohen's Kappa score measures the degree of agreement between two raters beyond what might happen by chance. Ranging from -1 to 1, a higher score indicates better agreement. The formula is:

$$\text{Kappa} = \frac{p_o - p_e}{1 - p_e} \tag{3.17}$$

  where p_o is the observed agreement and p_e is the expected agreement by chance, calculated using the distribution of each rater's ratings.

3.4.3 Final evaluation pipeline

In our experiment, requesting the expert to clarify the outcome is costly. Initially, we evaluate the generated images mainly by using the FID, PSNR, and SSIM scores, followed by the evaluation of the classification task result using the F1, AUC, and Cohen's Kappa scores. We will run the McNemar significance test (McNemar, 1947)to determine the statistical significance of the obtained classification results. The McNemar significance test is a statistical method employed to determine whether there exists a significant association between two categorical variables in a binary classification task. In this regard, the McNemar test assists in evaluating whether the observed distribution of predicted class labels significantly deviates from the expected distribution. In binary classification, there are two categorical variables: the true class labels (also called the ground truth) and the predicted class labels generated by the classification model. The McNemar significance test enables us to assess the significance of the association between the predicted class labels and the true class labels, or if their distribution varies from the anticipated distribution by chance.

4. Experiments

4.1 Label imputation

The dataset includes four types of picture annotations: uncertain (labeled as -1), negative (labeled as 0), positive (labeled as 1), and not mentioned (labeled as NM), respectively. As NM cannot be used as a label for training and the label occupies such a large proportion in the dataset that it cannot be ignored. Two methods can be used to replace it with numbers. The first imputation method involves filling NM with 0, while the second involves filling NM with 1. It should be noted that the training method for binary classification tasks differs from that used for multi-label classification tasks if the NM and uncertain labels are involved. The original paper (Irvin et al., 2019)we followed did not mention how to fill NM, but it did demonstrate the AUC score results for the five different methods we mentioned in Section 3.1 used to address the uncertain label.

Their results indicate that different methods perform best for different diseases. In general, U-Selftrained, U-MultiClass, and U-Ones exhibit satisfactory performance. Due to the relatively complicated process of the first two methods in the baseline model, we chose to use either U-Ones or U-Zeros to deal with uncertain labels. We subsequently conducted experiments that led to the development of four data preprocessing methods, using two approaches to replace NM values in combination with either U-Ones or U-Zeros. Furthermore, we analyzed the classification performance in terms of F1 score and AUC score based on the validation dataset with both pneumothorax labels and pleural effusion labels with the DenseNet.

4.2 Classification model

We used the DenseNet121, Inception-ResNet-v2, and CheXNet as described in Section 3.3 respectively to compare the classification results. The best-performing model in terms of AUC and F1 scores was selected as the classifier in the following process. The specific hyperparameter settings are as follows.

- **DenseNet161:**

```
growth_rate = 32
num_init_features = 64
bn_size = 4
drop_rate = 0.2
```

```
num_layers = (6, 12, 36, 24)
transition_num = 3
learning_rate = 0.001
batch_size = 16
```

- **Inception-ResNet-v2:**

```
bn_size = 11
drop_rate = 0.2
learning_rate = 0.01
batch_size = 4
```

- **CheXNet:**

```
learning_rate = 0.001
batch_size = 16
```

After deciding the classification network, we used the different types of images to perform the classification task and evaluate their results on both F1 and AUC scores.

4.3 Image masking

We employed various masks to mask images and evaluate the classification performance based on the F1 score and AUC score. Our aim is to confirm that the model focuses on disease features rather than unique markers or edge features. If we only include the borders and markers of an image and exclude the body and class features, we expect to get a poor classification result. If the classification result is better after using the mask to cover the edge or marker features, this indicates that those features can introduce bias. Those masking methods consist of the separate mask method, square mask method, joint mask method, and inverted joint mask method (Figure 18).

- **Square mask:** The square mask approach is almost the same as the method in Maguolo and Nanni (2021). We resized all images to (389,320) and set the square mask's size to (300,300). Moreover, the mask's center coincides with the center of the image. We found that when a square mask is used, the mask in the processed image covers all parts of the image that are not used for lung disease detection, so we decided to use a separate mask to cover only the lung area.
- **Separate mask:** As for the separate mask approach, we used two symmetric masks to cover each lung area in the image. In detail, we analyzed 1000 images in the

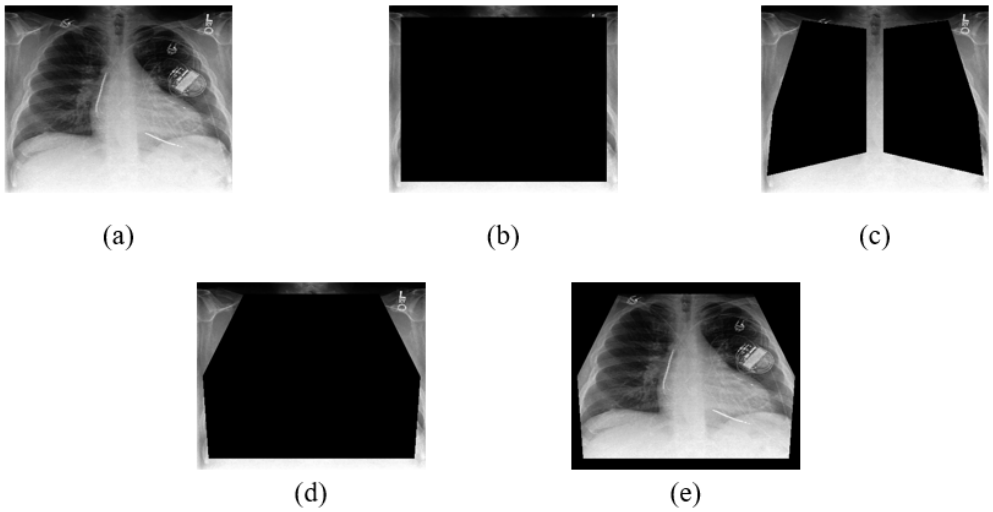


Figure 18. Original images and different images after using different mask methods. (a): original example image. (b): example image with the square mask. (c) example image with the separate mask. (d) example image with the joint mask. (e) example image with the inverted joint mask.

training dataset and manually identified the coordinates of every point on the two lung masks' polygons to obtain the coordinates that can make up the largest mask as shown in Figure 19.

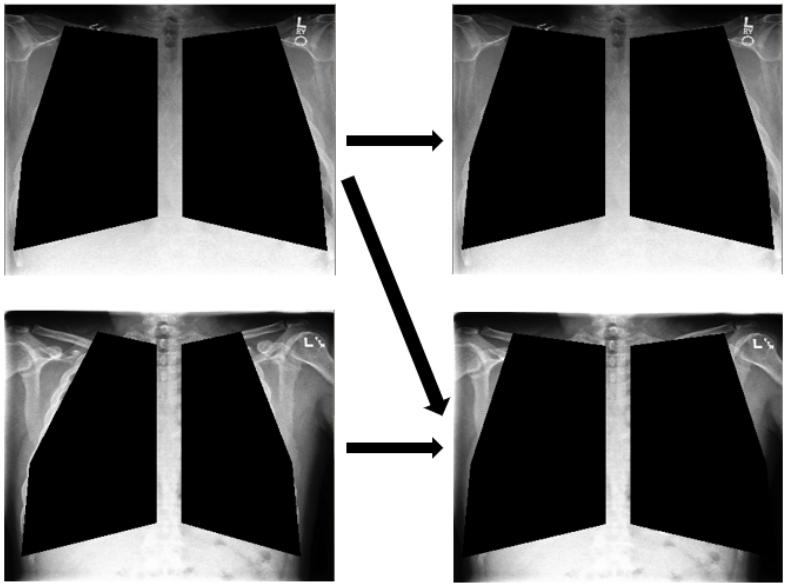


Figure 19. Schematic diagram for generating an image with a separate mask. First row: Image 1 and Image 1 with the separate mask. Second row: Image 2 and Image 2 with the separate mask.

- **Joint mask:** We assumed that some images may not have the patient positioned exactly in the middle without any rotation and that the location of the patient's spine may be seen as a potential feature. Then, we created the joint mask. We merged two masks by connecting them to form an axial-symmetric polygon.
- **Inverted joint mask:** After using the previously mentioned masks on the pneu-

mothorax dataset, we covered the feature area to determine the significance of edge features. If these features are confirmed to influence the classification outcome, we implemented the "inverted joint mask" approach. This approach entails revealing the section masked in the joint mask while masking other regions.

4.4 Traditional augmentation

Initially, we performed the augmentation individually using eight traditional methods. We then compared the results. Subsequently, we choose the techniques that positively impact the classification performance and compose them into a final augmentation method. The methods used include rotation, shearing, shifting, flipping, noise addition, coloring, cropping, and blurring. We list below the hyper-parameters of different augmentation methods:

- **Rotation:** We used the nearest method for the filling method and the rotation angle was chosen randomly between minus 90 and 90 degrees.
- **Flipping:** Given that there is not much difference between up-down flipping and left-right flipping in this dataset, we set a probability of 0.5 to perform a vertical flip or to perform a horizontal flip in each data augmentation process.
- **Translation:** Use the constant value which is 0.0 for the filling method, and the maximum shifting distance is 0.2 $\times$ image height and 0.2 $\times$ image width.
- **Shearing:** We used the nearest method for the filling method and chose a random integer between 0 and 30 as the intensity number which means the shearing angle is between 0 and 15 degrees.
- **Cropping:** We decided to use the random cropping method instead of the center cropping and random resized cropping method. And we set the image size after the cropping as (256, 256).
- **Adding noise**: We added Gaussian noise or Pepper noise with the same probability of 0.5. For the Gaussian noise, the mean value is equal to 0 while the variance value is 0.5 and the amplitude value is a random value within 30. For the pepper noise, there are no hyperparameters we need to change.
- **Blurring:** We set the size of the blurring Gaussian kernel to (11,11) and set the standard deviation to 10.
- **Coloring**: There is no saturation in the gray image. In the coloring method, we finetuned two hyperparameters that control the brightness and contrast. The initial value of the two hyperparameters is 1 which means keep the original image. We set those hyperparameters to random values between 1 and 2 which means increasing the brightness and contrast a little bit.

4.5 Augmentation Strategies

Two augmentation strategies are used to test traditional augmentation methods related to the Pneumothorax disease at first. The first method, Experiment Unbalanced (Table 4), involves selecting all 18074 images belonging to the positive class from the original dataset and augmenting them 11 times per image before proceeding further. For each augmentation time, there is a 0.5 probability of executing each individual traditional augmentation method or directly using the diffusion model for an enhancement. Subsequently, one-third of the negative images are chosen, and their augmentation is performed only one time per image. In the second method, Experiment Balanced (Table 5), all of the 18074 positive class images from the original dataset are selected first, followed by the selection of the same number of images (18074) from the negative class (200821). Thereafter, all selected images are augmented four times each and subsequently, amalgamated with all the original training images into the final training datasets.

	Original Training Dataset	Augmentated Training Dataset		Unbalanced Test Dataset
	Real Images	Real Images	Fake Images	Real Images
Negative	200821	160657	160657 * 1/3 = 53552	40164
Positive	22593	18074	18074 * 11 = 198814	4519

Table 4. Number of images for Experiment Unbalanced with pneumothorax

	Original Training Dataset	Augmentated Training Dataset		Unbalanced Test Dataset
	Real Images	Real Images	Fake Images	Real Images
Negative	200821	18074	18074 * 4 = 72296	40164
Positive	22593	18074	18074 * 4 = 72296	4519

Table 5. Number of images for Experiment Balanced with pneumothorax

After doing the classification experiment, we found that neither augmentation strategy was particularly good, so we used the Principal Component Analysis (PCA) method to extract the first two dimensions in the output vector (1x1024) of the last layer of DenseNet and used the data of the first dimension as the x coordinate, and the data of the second dimension as the y coordinate, so as to replace the original image in the coordinate system. The 2000 randomly sampled images (1000 positive and 1000 negative) after the PCA

method are shown in Figure 20. We calculated the distance of all the downscaled data points from the downscaled decision boundary and computed the average value. After that we selected all the images that were misclassified and their distance from the decision boundary was less than the mean value and performed image augmentation operation on them. This augmentation method is called the misclassified augmentation. All the final augmented dataset should be the almost balanced dataset. We first augmented each of

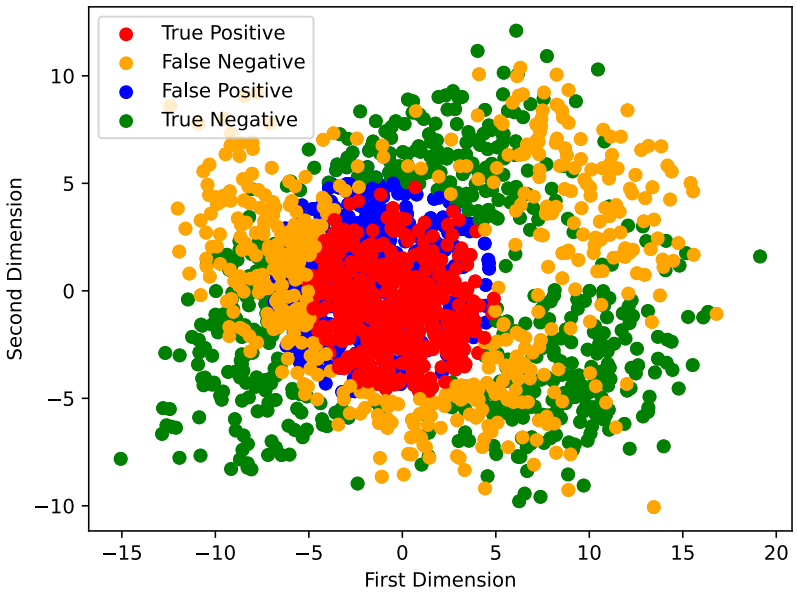


Figure 20. Visualization of 2000 random training images with the disease of Pneumothorax after the PCA method.

the traditional data augmentation methods using both unbalanced and balanced methods and compared the final results. For the final comparison between the traditional data augmentation methods and diffusion models, We used the unbalanced, balanced and misclassified augmentation methods to augment and compare the results.

4.6 Evaluation of synthetic images from diffusion model

Apart from the traditional image data augmentation methods, we also use some diffusion models to generate synthetic images. The methods we used include the DDPM model (Ho et al., 2020), the DDIM model (Song et al., 2020), stable diffusion with the pre-trained weight on the LAION-2B dataset, and stable diffusion with LoRa finetuning. The specific parameters are as follows:

- **DDPM:** For the diffusion model, we mainly use the same code as the tutorial file

on the 'Hugging Face' website. [1] The noise scheduler and the U-Net model we use are the DDPMscheduler and Unet2Dmodel in the diffuser library, respectively. The detailed parameters are as follows:

```
in_channels = 1,
out_channels = 1,
layers_per_block = 2,
block_out_channels = (128,128,256,256,512,512),
down_block_types=(
    "DownBlock2D",
    "DownBlock2D",
    "DownBlock2D",
    "DownBlock2D",
    "AttnDownBlock2D",
    "DownBlock2D",
),
up_block_types=(
    "UpBlock2D",
    "AttnUpBlock2D",
    "UpBlock2D",
    "UpBlock2D",
    "UpBlock2D",
    "UpBlock2D",
),
```

In addition, a linear noise schedule with a range of [0.0001, 0.02] is used, and the total number of diffusion steps is 1000 by default. We use the different diffusion steps and epochs to do the experiment and we find out the number of epochs does not obviously affect the final result, but the number of timesteps has a significant influence. We get the best result when the timestep is equal to 4000 and we set the number of epochs equal to 40.

- **DDIM:** We mainly use the code from the GitHub project. [2] The detailed hyperparameter settings are almost the same as the DDPM network:

```
in_channels = 1,
out_channels = 1,
layers_per_block = 2,
beta_start = 0.0001
```

[1] https://huggingface.co/docs/diffusers/tutorials/basic training
[2] https://github.com/ermongroup/ddim

```
beta_end = 0.02
num_diffusion_timesteps = 4000
batch_size = 16
n_epochs = 8000
n_iters = 4000000
learning_rate = 0.0001
```

- **Stable Model 1:** For the stable diffusion model with pretrained weight, the pretrained weight is trained on the LAION-2B dataset, The prompt we used is "Chest X-ray images of people with Pneumothorax" and "Chest X-ray images of healthy people"
- **Stable Model 2:** For the stable diffusion model with finetuning method. First, we use the fixed sentences' structure and label of each patient to generate the sentence of each image. The sentence can be "Chest X-ray images of people with disease 1, disease 2." or "Chest X-ray images of healthy people". Then we use the tools of LoRA to finetune the model and get a new weight. The parameters of the LoRA tools are shown as follows:

```
train_batch_size = 16,
epoch = 20,
mixed_percision = fp16,
save_percision = fp16,
Learning_rate = 0.0001,
LR_Scheduler = cosine,
LR_warmup = 10%,
Optimizer = AdamW,
Text_Encoder_learning_rate = 5e-5,
Unet_learning_rate = 0.001,
Network_Rank = 8,
Network_Alpha = 1,
Max_resolution = 256,256.
```

For the DDIM model, we analyzed the impact of using RGB and grayscale images and we evaluated the SSIM, PSNR and FID metrics on the validation dataset. We also compared the performance of different diffusion models on those metrics on the validation dataset. Finally, we compared the classification results of the best diffusion model and traditional data augmentation methods on the test dataset.

5. Experimental Results

5.1 Label imputation

After using the four combined methods to process the -1 and NM labels in the dataset, the data distribution of the positive and negative samples of pneumothorax and pleural effusion are shown in Tables 6 and 7.

	1 (Positive)	0 (Negative)	-1 (Uncertain)	NM
Original	19448 (8.7%)	56341 (25.2%)	3145 (1.4%)	144480 (64.7%)
FillNM(0) U-ones	22593 (10.1%)	200821 (89.9%)	—-	—-
FillNM(0) U-zeros	19448 (8.7%)	203966 (91.3%)	—-	—-
FillNM(1) U-ones	167073 (74.8%)	56341 (25.2%)	—-	—-
FillNM(1) U-zeros	163928 (73.4%)	59486 (26.6%)	—-	—-

Table 6. Pneumothorax label distribution of different preprocessing methods for the training dataset

	1 (Positive)	0 (Negative)	-1 (Uncertain)	NM
Original	86187 (38.6%)	35396 (15.8%)	11628 (5.2%)	90203 (40.4%)
FillNM(0) U-ones	97815 (43.8%)	125599 (56.2%)	—-	—-
FillNM(0) U-zeros	86187 (38.6%)	137227 (61.4%)	—-	—-
FillNM(1) U-ones	188018 (84.2%)	35396 (15.8%)	—-	—-
FillNM(1) U-zeros	176390 (79.0%)	47024 (21.0%)	—-	—-

Table 7. Pleural effusion label distribution of different preprocessing methods for the training dataset

The classification outcomes of applying different methods to do the filling NM and uncertain are presented in the following Table 8. Based on the result, We chose to fill the NM with 0 and use the U-ones method to replace all the uncertain labels in the dataset. For the Pneumothorax dataset, the number of negative samples accounts for the vast majority of the total number of samples while the positive and negative samples are almost balanced for the pleural effusion.

Method Names	F1 score Pneumothorax	AUC score Pneumothorax	F1 score Pleural Effusion	AUC score Pleural Effusion
FillNM(0) U-ones	**0.2331**	**0.6486**	**0.7646**	**0.8570**
FillNM(0) U-zeros	0.2064	0.5429	0.7023	0.7229
FillNM(1) U-ones	0.2217	0.6323	0.7575	0.8478
FillNM(1) U-zeros	0.2085	0.6291	0.7421	0.8427

Table 8. Final classification result of using different dataset preprocessing methods on validation dataset. Bold font: The best score under certain labels and certain metrics.

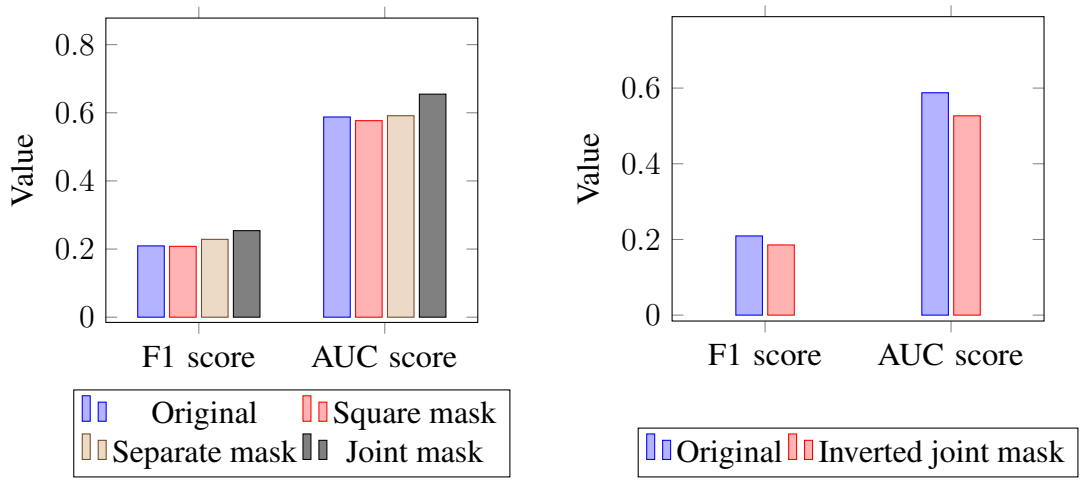
5.2 Classification model

Our final results using the three classification networks on both pneumothorax and pleural effusion labels on the two metrics of AUC and F1 are shown in Table 9. For the pneumothorax label, DenseNet performs best on either F1 score and AUC score. For the pleural effusion label, CheXNet performs best on either F1 score and AUC score. The Inception-ResNet performs worst on every metric for every label. For the result of different image types, dataset with only frontal images performs best on either metrics and diseases (Table 10). But we finally chose to use the original dataset which contains both lateral images and frontal images for our experiment.

Classification Network	F1 score Pneumothorax	AUC score Pneumothorax	F1 score Pleural Effusion	AUC score Pleural Effusion
Inception-ResNet	0.1842	0.5545	0.6924	0.7981
DenseNet	**0.2093**	**0.5877**	0.7154	0.8209
CheXNet	0.1914	0.5799	**0.7236**	**0.8315**

Table 9. Classification model's performance for different image types and diseases on validation dataset. Bold font: The best score under certain labels and certain metrics.

.



(a) (b)

Figure 21. Final result of using different mask methods. (a): Different masking experiments for verifying that edge features and markers have an effect on final results. (b): Experiment of using inverted joint mask.

Image datasets	F1 score Pneumothorax	AUC score Pneumothorax	F1 score Pleural Effusion	AUC score Pleural Effusion
Lateral	0.1878	0.5613	0.6458	0.7474
Frontal	**0.2142**	**0.5998**	**0.7207**	**0.8311**
Combined	0.2093	0.5877	0.7154	0.8209

Table 10. Classification model's performance for different classification networks and diseases on validation dataset. Bold font: The best score under certain labels and certain metrics.

.

5.3 Image masking

The evaluation presented in Figure 21a showed the performance impact of various mask modifications to the original model, specifically the introduction of square, separate, and joint masks. Notably, the model enhanced with a joint mask exhibited superior performance, achieving the highest scores in both F1 and AUC metrics. Conversely, the unmodified original model was observed to be the least effective. The results of the baseline model and those after adding an inverted joint task are shown in Figure 21b. The results show that both the F1 and AUC scores decrease after adding the inverted joint mask.

5.4 Traditional augmentation

Different images after the eight traditional augmentation methods are shown in Figure 22. It shows that shearing, cropping and coloring methods change the original image a lot.

The final classification results with different data augmentation methods and augmentation strategies on the pneumothorax dataset are shown in Table 11. For the augmentation methods, all methods can improve the accuracy but only rotation, flipping, shearing, adding noise and blurring improve the result on other evaluation metrics. Among them, the rotation method performs best. For the augmentation strategy, almost all the unbalanced experiments perform better than the balanced experiments except the cropping method. In general, those five methods can increase the performance compared to the pneumothorax baseline model. However, the performance of any augmented dataset is still worse than the performance of the pleural effusion dataset which is naturally balanced. We chose the rotation, shearing, flipping, adding noise, and blurring for the final combined traditional augmentation methods, the result is shown in Table 12, we can observe that compared with the results of the pneumothorax baseline model, traditional augmentation does improve the performance of classification based on the accuracy, F1 score, AUC score and Cohen's Kappa score. We also use the McNemar test for the original pneumothorax classification model and combined model. The final p-value is 0.0012 while the value of the test is equal to 3963.15 which shows significant improvements.

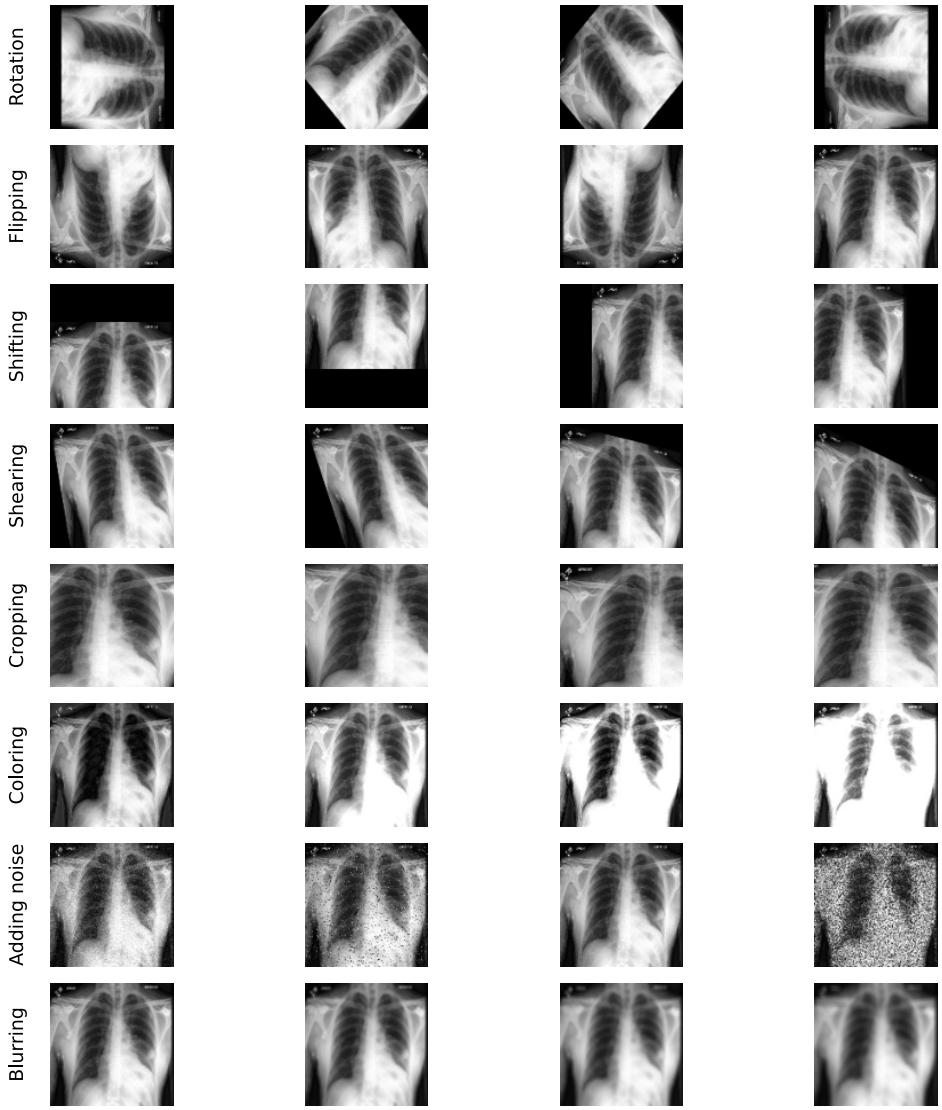


Figure 22. Example images after different traditional augmentation methods

Dataset/Method	Strategy	Accuracy	F1 score	AUC	Cohen's Kappa
Pneumothorax	—-	0.3579	0.2093	0.5877	0.0666
Pleural Effusion	—-	0.7758	0.7154	0.8209	0.5263
Rotation	**Unbalanced**	**0.8075**	**0.4097**	**0.6372**	**0.3139**
	Balanced	0.7687	0.2138	0.6159	0.2387
Flipping	Unbalanced	—-	—-	—-	—-
	Balanced	**0.7838**	**0.3710**	**0.6242**	**0.2662**
Shifting	Unbalanced	0.6870	0.1812	0.5568	0.0381
	Balanced	0.6585	0.1407	0.5412	0.0316
Shearing	**Unbalanced**	**0.7316**	**0.2738**	**0.5998**	**0.1487**
	Balanced	0.6977	0.2551	0.5916	0.1427
Cropping	Unbalanced	0.6684	0.1829	0.5478	0.0365
	Balanced	0.6829	0.1720	0.5316	0.0272
Coloring	Unbalanced	0.7545	0.1679	0.5682	0.0404
	Balanced	0.7334	0.1762	0.5698	0.0432
Adding noise	**Unbalanced**	**0.6949**	**0.2167**	**0.5923**	**0.0787**
	Balanced	0.6857	0.2122	0.5887	0.0721
Blurring	**Unbalanced**	**0.7238**	**0.2681**	**0.5981**	**0.1409**
	Balanced	0.7078	0.2475	0.5831	0.1151

Table 11. Classification result of different traditional augmentation methods and strategies on the pneumothorax test dataset. Bold: traditional image augmentation methods and augmentation strategies with relatively improved results compared to the baseline model with pneumothorax labels.

Dataset/Method	Strategy	Accuracy	F1 score	AUC	Cohen's Kappa
Pneumothorax	—-	0.3579	0.2093	0.5877	0.0666
Pleural Effusion	—-	0.7758	0.7154	0.8209	0.5263
Combined	**Unbalanced**	**0.7714**	**0.4177**	**0.7589**	**0.3149**
	Balanced	0.7518	0.3834	0.6392	0.2733

Table 12. Classification result of combined traditional augmentation methods on the pneumothorax test dataset. Bold: augmentation methods and strategies with best results.

5.5 Augmentation strategies

Our results of using unbalanced and balanced augmentation strategies on the traditional data augmentation methods are shown in Table 11 and 12 in the last section 5.4. For the misclassified augmentation strategy, the result is shown in Table 15 in section 5.7.

5.6 Evaluation of synthetic images from diffusion model

The results of the DDIM model with RGB images input and grayscale images input are shown in Table 13. Both approaches exhibit similar performance in terms of reconstruction metrics and FID scores and the result with grayscale images is a little higher than the result with RGB images. So, we chose the model for the one-channel image to stand for the DDIM model. Images after using different diffusion models to augment are shown in Figures 23. We use multiple criteria for generated images to evaluate images generated by different diffusion models. As the result shows in Tables 14, the DDIM model has the best performance.

	SSIM	PSNR (dB)	FID
DDIM Model with RGB images	0.5833 ± 0.07	22.35 ± 0.68	71.55
DDIM Model with grayscale images	0.5914 ± 0.08	22.49 ± 0.65	71.17

Table 13. Generation metric results of different diffusion models. For SSIM and PSNR, the larger the value, the better the performance. For the FID score, the lower the value, the better the performance.

	SSIM	PSNR (dB)	FID
Traditional Augmentation	0.3310 ± 0.08	12.57 ± 0.65	90.44
DDPM Model with grayscale images	0.4378 ± 0.07	20.08 ± 0.72	80.29
DDIM Model with grayscale images	**0.5914 ± 0.08**	**22.49 ± 0.65**	**71.17**
SD Model with pre-trained datasets	0.1056 ± 0.12	8.77 ± 0.46	325.74
SD Model with finetuning	0.3525 ± 0.05	12.00 ± 0.74	119.84

Table 14. Reconstruction generation metric results of different diffusion models. Bold font: The best generation score for the original dataset .

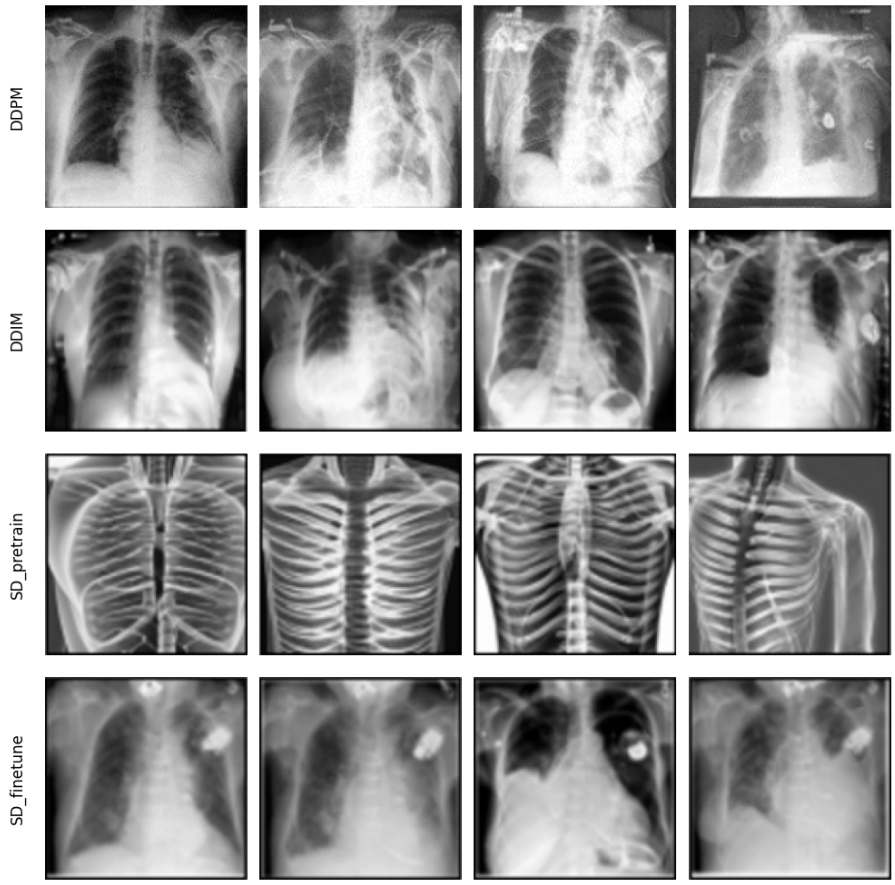


Figure 23. Negative images generated by different diffusion models

5.7 Traditionional versus diffusion-based augmented classification

We used the DDIM model and traditional data augmentation methods to generate images with three different augmentation strategies and evaluate their performance with various classification metrics (Table 15). It shows that DDIM did perform better not just than the original experiment, but also better than the traditional augmentation methods. We compared the result of the baseline experiment for the pneumothorax and the traditional augmentation method, the p-value of the significance test is 0.001 and the p-value of the test between the baseline experiment and the DDIM augmentation method is 0.005. Both augmentation methods help the classifier improve much compared to the baseline model, especially on the Cohen's Kappa metric which indicates the imbalance problem for the classification task. Likewise, we did a significance test between the result of the traditional augmentation method and the DDIM method, the p-value is 0.03, which means there are still significant improvements for the DDIM model compared to the traditional method.

Dataset	Experiment	Accuracy	F1 score	AUC	Cohen's Kappa
Pneumothorax	—-	0.3579	0.1855	0.5268	0.0666
Pleural Effusion	—-	0.7758	0.7154	0.8209	0.5263
Traditional Augmentation Methods	Unbalanced	0.7714	0.4177	0.7589	0.3149
	Balanced	0.7518	0.3834	0.6392	0.2733
	Misclassified augmentation	0.7923	0.4516	0.7942	0.3248
DDIM	Unbalanced	0.8221	0.4467	0.7924	0.3398
	Balanced	0.8014	0.4257	0.7694	0.3216
	Misclassified augmentation	**0.8427**	**0.4629**	**0.8142**	**0.3574**

Table 15. Classification results of different augmentation methods and strategies on pneumothorax test dataset. Bold font: The best score with certain methods and strategies.

6. Discussion

Medical image classification suffers from a common challenge: the data imbalance between positive and negative samples. The goal of our research was to address this problem by evaluating the effectiveness of traditional image augmentation techniques and diffusion models. The correct classification of medical images is crucial as it has the potential to improve diagnosis and treatment outcomes for patients. Failure to address the issue of sample imbalance could influence the accuracy of diagnostic tools, subsequently impacting patient care. While traditional image augmentation methods can alleviate the issue of unbalanced datasets to some extent, the introduction of diffusion models can further improve the quality of generated images, leading to better results in the final downstream tasks. This is because traditional image augmentation methods simply add some distortions to the original images without generating any meaningful new image information.

Our primary discovery was the noteworthy effectiveness of diffusion models, particularly DDPM and DDIM, in addressing sample imbalance. By generating lifelike images, these models significantly enhanced task performance, outpacing traditional augmentation techniques in our experiments. While Sundaram and Hulkund (2021) primarily explored GAN networks alongside traditional methods, our study provides fresh insights by emphasizing the potential of diffusion models.

For the label imputation methods, we expanded upon the foundational work of Irvin et al. (2019). We experimented with different strategies for replacing multiple labels with a single label. Our methodological choices were driven by the pursuit of optimal results, which we incorporated into our pipeline. Additional findings spotlight the influence of external markers and annotations in X-ray images. Our research underscored the significance of meticulous masking techniques to eliminate potential distractions. Moreover, DenseNet proves more effective for the binary classification task in our dataset than both Inception-ResNet and CheXNet, thanks to its precise convolutional kernel and dense features.

While our research offered several insights, there were inherent limitations. The decision to oversimplify uncertain data by categorizing it as positive or negative might have introduced biases. In terms of the stable diffusion model, the lack of corresponding radiology reports and various finetuning methods indicates the intricacy of multimodal image generation in medical image classification area.

In our exploration of label imputation, the significant presence of "Not Mentioned" la-

bels poses a challenge. Simply substituting either positive or negative labels can skew the results. This highlights the need for a novel substitution strategy to strengthen the classification ability of the final model. In addition, our choice of inverted joint mask was relatively rudimentary. Future work could explore the potential of deep learning-based image segmentation followed by masking to eliminate edge features and specific markers. Moreover, future research could benefit from experiments with stable diffusion models and text data derived from radiological reports.

7. Conclusions

In conclusion, our research reveals the potential and challenges of using diffusion models in medical image classification. While traditional image augmentation methods remain valuable, diffusion models stand out as a promising alternative due to their minimal requirements for fine-tuning and their ability to produce high-quality images. Despite our advancements, not only in the application of more rigorous label imputation and image masking methods but also in proposing better data augmentation strategies, as evidenced by our experiments with the stable diffusion model, we should do more experiments for diffusion models to propose better solutions in detailed areas like multimodal image generation.

Bibliography

Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. (2019). Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, 38(8):1788–1800.

Bansal, M. A., Sharma, D. R., and Kathuria, D. M. (2022). A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (CSUR)*, 54(10s):1–29.

Barber, D. and Hose, D. (2005). Automatic segmentation of medical images using image registration: diagnostic and simulation applications. *Journal of medical engineering & technology*, 29(2):53–63.

Bearman, A., Russakovsky, O., Ferrari, V., and Fei-Fei, L. (2016). What's the point: Semantic segmentation with point supervision. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 549–565. Springer.

Bhalodia, R., Elhabian, S. Y., Kavan, L., and Whitaker, R. T. (2018). Deepssm: a deep learning framework for statistical shape modeling from raw images. In *Shape in Medical Imaging: International Workshop, ShapeMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings*, pages 244–257. Springer.

Bria, A., Marrocco, C., and Tortorella, F. (2020). Addressing class imbalance in deep learning for small lesion detection on medical images. *Computers in biology and medicine*, 120:103735.

Burt, P. J. (1983). Edward, and eh adelson. the laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(532-540):340.

Chen, H., Dou, Q., Yu, L., Qin, J., and Heng, P.-A. (2018). Voxresnet: Deep voxelwise residual networks for brain segmentation from 3d mr images. *NeuroImage*, 170:446–455.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.

Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., and Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59.

Corral Acero, J., Zacur, E., Xu, H., Ariga, R., Bueno-Orovio, A., Lamata, P., and Grau, V. (2019). Smod-data augmentation based on statistical models of deformation to enhance segmentation in 2d cine cardiac mri. In *Functional Imaging and Modeling of the Heart: 10th International Conference, FIMH 2019, Bordeaux, France, June 6–8, 2019, Proceedings 10*, pages 361–369. Springer.

Dalca, A. V., Yu, E., Golland, P., Fischl, B., Sabuncu, M. R., and Eugenio Iglesias, J. (2019). Unsupervised deep learning for bayesian brain mri segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*, pages 356–365. Springer.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Huang, P., Liu, X., and Huang, Y. (2021). Data augmentation for medical mr image using generative adversarial networks. *arXiv preprint arXiv:2111.14297*.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. (2019). Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597.

Javaid, U., Dasnoy, D., and Lee, J. A. (2019). Semantic segmentation of computed tomography for radiotherapy with deep learning: compensating insufficient annotation quality using contour augmentation. In *Medical Imaging 2019: Image Processing*, volume 10949, pages 682–694. SPIE.

Johnson, A. E., Pollard, T. J., Berkowitz, S. J., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Mark, R. G., and Horng, S. (2019). Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54.

Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., and Irani, M. (2022). Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*.

Kim, S., An, S., Chikontwe, P., and Park, S. H. (2021). Bidirectional rnn-based few shot learning for 3d medical image segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1808–1816.

Kim, Y.-C., Kim, K. R., Choi, K., Kim, M., Chung, Y., and Choe, Y. H. (2019). Evcmr: a tool for the quantitative evaluation and visualization of cardiac mri data. *Computers in Biology and Medicine*, 111:103334.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kompanek, M., Tamajka, M., and Benesova, W. (2019). Volumetrie data augmentation as an effective tool in mri classification using 3d convolutional neural network. In *2019 international conference on systems, signals and image processing (IWSSIP)*, pages 115–119. IEEE.

Kora, P., Ooi, C. P., Faust, O., Raghavendra, U., Gudigar, A., Chan, W. Y., Meenakshi, K., Swaraja, K., Plawiak, P., and Acharya, U. R. (2022). Transfer learning techniques for medical image analysis: A review. *Biocybernetics and Biomedical Engineering*, 42(1):79–107.

Kozlov, V. N. (2000). Visual pattern and geometric transformations of images. *Pattern Recognition and Image Analysis*, 10(3):321–342.

Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.-W., and Heng, P.-A. (2018). H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Luque, A., Carrasco, A., Martín, A., and de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231.

Maguolo, G. and Nanni, L. (2021). A critic evaluation of methods for covid-19 automatic detection from x-ray images. *Information Fusion*, 76:1–7.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR.

Nishio, M., Noguchi, S., and Fujimoto, K. (2020). Automatic pancreas segmentation using coarse-scaled 2d model of deep learning: usefulness of data augmentation and deep u-net. *Applied Sciences*, 10(10):3360.

Novosad, P., Fonov, V., Collins, D. L., and Initiative†, A. D. N. (2020). Accurate and robust segmentation of neuroanatomy in t1-weighted mri by combining spatial priors with deep convolutional neural networks. *Human brain mapping*, 41(2):309–327.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., et al. (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Pang, T., Wong, J. H. D., Ng, W. L., and Chan, C. S. (2021). Semi-supervised gan-based radiomics model for data augmentation in breast ultrasound mass classification. *Computer Methods and Programs in Biomedicine*, 203:106018.

Perez, F., Vasconcelos, C., Avila, S., and Valle, E. (2018). Data augmentation for skin lesion analysis. In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis: First International Workshop, OR 2.0 2018, 5th International Workshop, CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16 and 20, 2018, Proceedings 5*, pages 303–311. Springer.

Pesteie, M., Abolmaesumi, P., and Rohling, R. N. (2019). Adaptive augmentation of medical data using independently conditional variational auto-encoders. *IEEE transactions on medical imaging*, 38(12):2807–2820.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

Rigaud, B., Anderson, B. M., Zhiqian, H. Y., Gobeli, M., Cazoulat, G., Söderberg, J., Samuelsson, E., Lidberg, D., Ward, C., Taku, N., et al. (2021). Automatic segmentation using deep learning to enable online dose optimization during adaptive radiation therapy of cervical cancer. *International Journal of Radiation Oncology* Biology* Physics*, 109(4):1096–1110.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Sandfort, V., Yan, K., Pickhardt, P. J., and Summers, R. M. (2019). Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 9(1):16884.

Schoenberg, I. J. (1964). Spline interpolation and best quadrature formulae.

Shen, D., Wu, G., and Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248.

Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. Edinburgh.

Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Stefan, L.-D., Cid, Y. D., del Toro, O. A. J., Ionescu, B., and Müller, H. (2017). Finding and classifying tuberculosis types for a targeted treatment: Medgift-upb participation in the imageclef 2017 tuberculosis task. In *CLEF (Working Notes)*.

Summers, R. (2019). Nih chest x-ray dataset of 14 common thorax disease categories.

Sundaram, S. and Hulkund, N. (2021). Gan-based data augmentation for chest x-ray classification. *arXiv preprint arXiv:2107.02970*.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., and Pinheiro, P. R. (2020). Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection. *Ieee Access*, 8:91916–91923.

Wang, S., Li, C., Wang, R., Liu, Z., Wang, M., Tan, H., Wu, Y., Liu, X., Sun, H., Yang, R., et al. (2021). Annotation-efficient deep learning for automatic medical image segmentation. *Nature communications*, 12(1):5915.

Wodzinski, M., Banzato, T., Atzori, M., Andrearczyk, V., Cid, Y. D., and Muller, H. (2020). Training deep neural networks for small and highly heterogeneous mri datasets for cancer grading. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1758–1761. IEEE.

Yuan, Z., Yan, Y., Sonka, M., and Yang, T. (2021). Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3040–3049.

Zeng, T., Li, R., Mukkamala, R., Ye, J., and Ji, S. (2015). Deep convolutional neural networks for annotating gene expression patterns in the mouse brain. *BMC bioinformatics*, 16:1–10.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, L., Wang, X., Yang, D., Sanford, T., Harmon, S., Turkbey, B., Wood, B. J., Roth, H., Myronenko, A., Xu, D., et al. (2020). Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE transactions on medical imaging*, 39(7):2531–2540.

Zhang, Y., Wu, J., Liu, Y., Chen, Y., Chen, W., Wu, E. X., Li, C., and Tang, X. (2021). A deep learning framework for pancreas segmentation with multi-atlas registration and 3d level-set. *Medical Image Analysis*, 68:101884.

Zhao, A., Balakrishnan, G., Durand, F., Guttag, J. V., and Dalca, A. V. (2019). Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8543–8553.

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., and Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer.

Zhu, B., Liu, J. Z., Cauley, S. F., Rosen, B. R., and Rosen, M. S. (2018). Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492.