



Chest X-Ray Classification with synthetic Data

By Ahmed Tarek & Ines Elgataa



TOPICS

Introduction

Dataset

Problem and solution

Conditional VAE

Enhanced Conditional VAE

DDPM

Classifier

Results

Introduction

This project leverages advanced generative modeling techniques, specifically Variational Autoencoders (**VAEs**), **Enhanced VAEs**, and Denoising Diffusion Probabilistic Models (**DDPMs**), to generate high-quality synthetic chest X-ray images. These generative models will be compared in their effectiveness for addressing the common yet challenging issue of class imbalance in medical imaging datasets..



Dataset



Chest X-ray Images (Pneumonia)

Source: [Kaggle Chest X-ray Pneumonia Dataset](#)

Description

The dataset contains chest X-ray images used for diagnosing pneumonia. It is structured into three subsets:

- Train: Primary training set.
- Test: Evaluation set for model validation.
- Val: Additional set for tuning hyperparameters and model validation.

Images are categorized into two distinct classes:

- Normal: Chest X-rays with no signs of pneumonia.
- Pneumonia: Chest X-rays exhibiting characteristics consistent with pneumonia.

The problem and our solution

Problem:

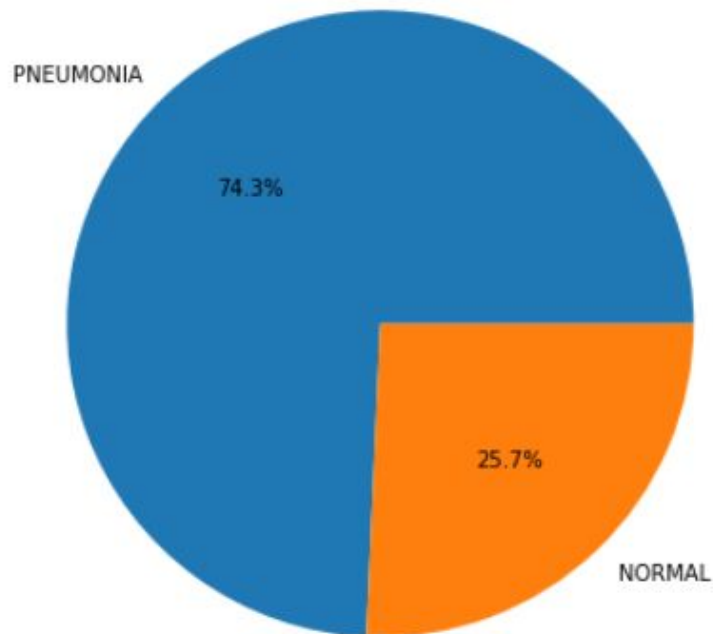
- PNEUMONIA: 3875 images
- NORMAL: 1341 images

Class imbalance in medical imaging datasets impacts diagnostic accuracy.

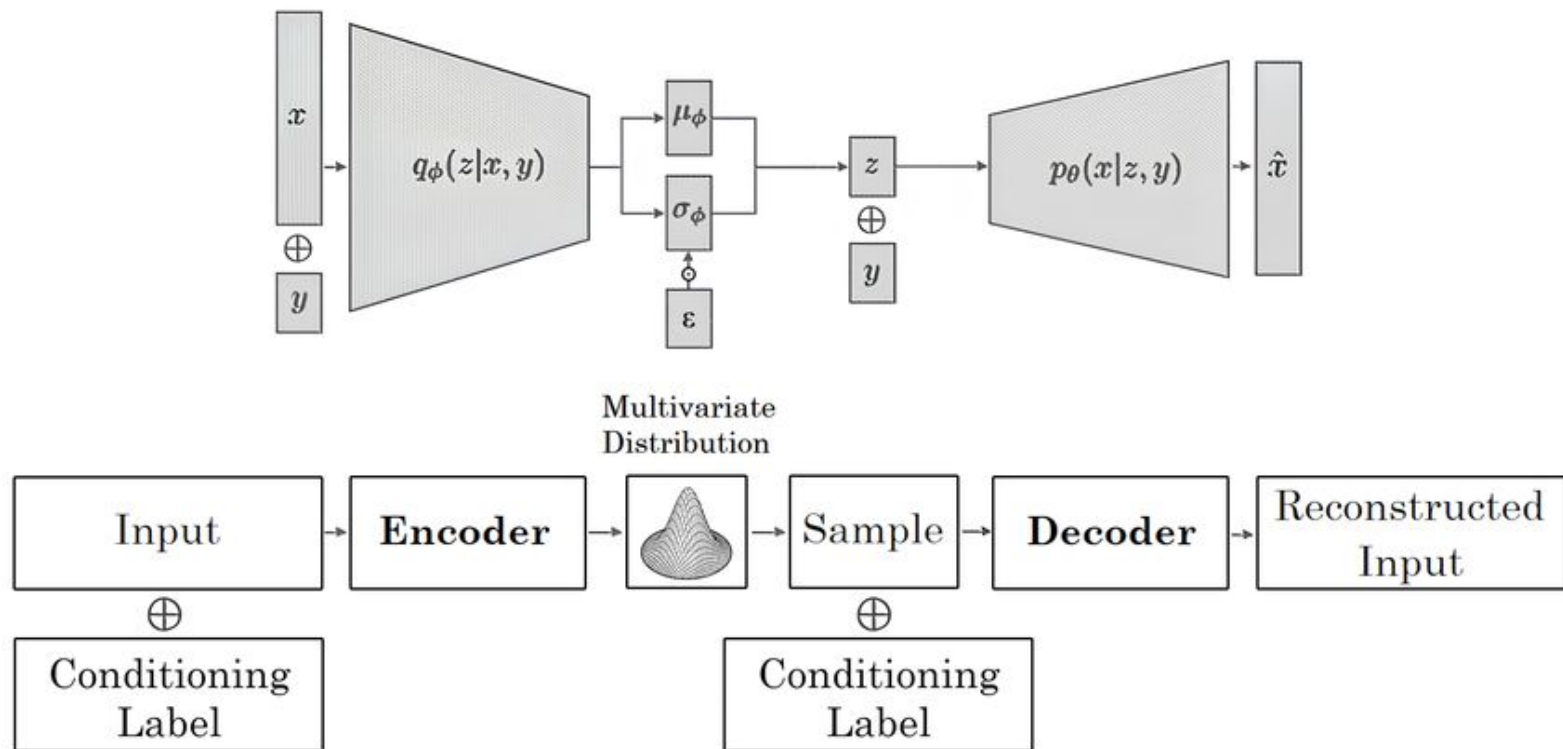
Solutions:

- Employ and compare three generative models (VAE, Enhanced VAE, and DDPM) to generate balanced synthetic chest X-rays.

Proportion of each observed category



Conditional VAE



Why Conditional VAE

Controlled Generation

By conditioning on your label y (e.g., “Normal”), your C-VAE can generate chest X-rays specific to that class, helping to balance your dataset.

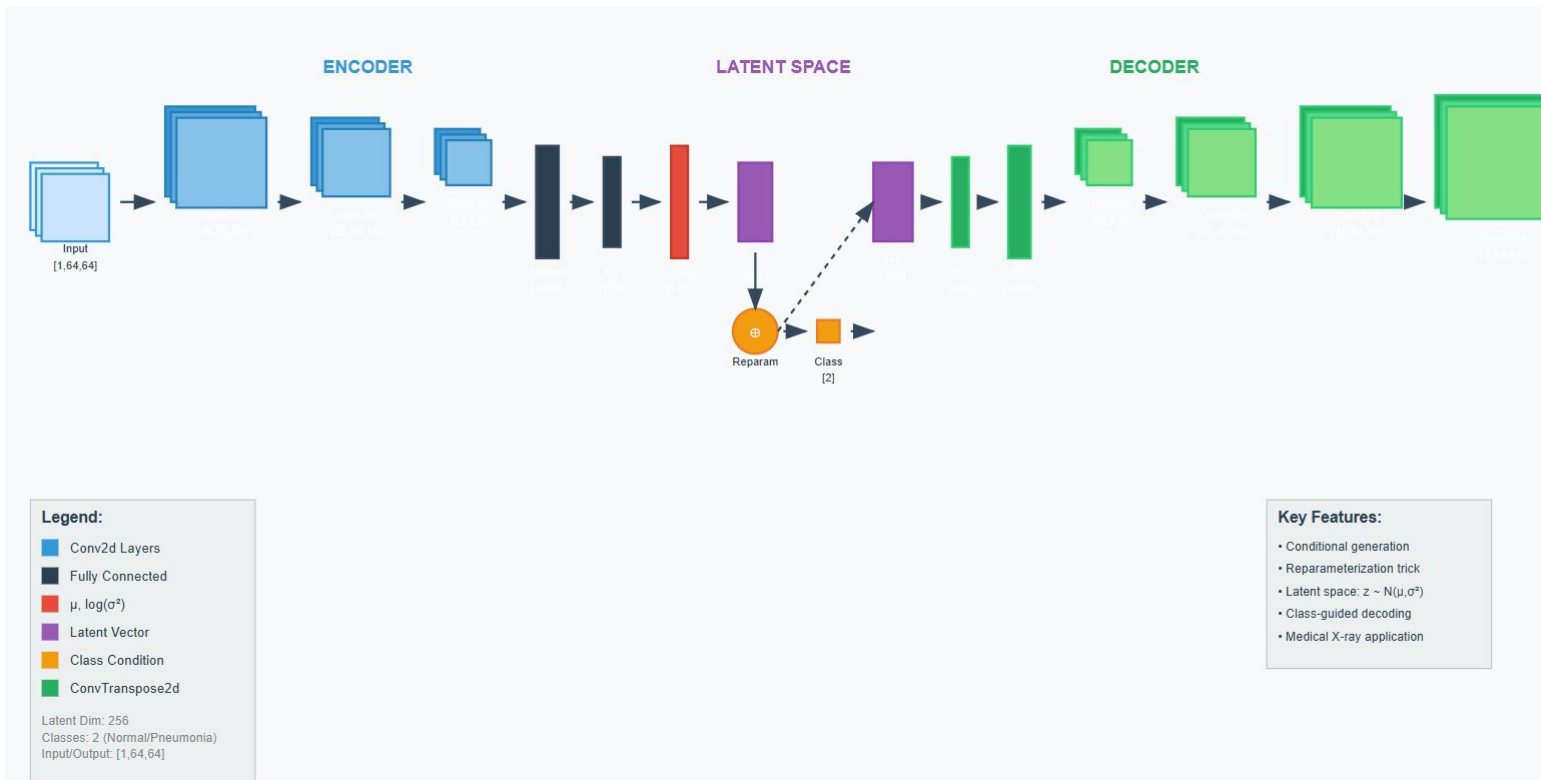
When conditioning on an additional variable y (e.g., class label), both the encoder and decoder include y :

$$\ln p(x|y) \geq \underbrace{\mathbb{E}_{q_\phi(z|x,y)}[\ln p_\theta(x|z,y)]}_{\text{Reconstruction Term}} - \underbrace{D_{\text{KL}}(q_\phi(z|x,y) \parallel p_\theta(z|y))}_{\text{Conditional Regularization}}$$

Thus the CVAE loss is:

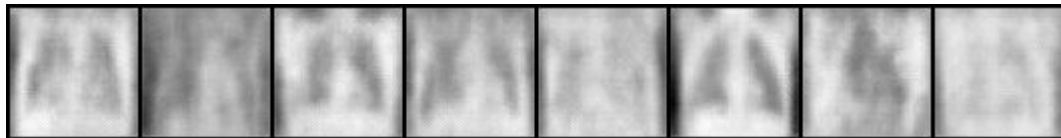
$$\mathcal{L}_{\text{CVAE}}(x, y) = -\mathbb{E}_{q_\phi(z|x,y)}[\ln p_\theta(x|z,y)] + D_{\text{KL}}(q_\phi(z|x,y) \parallel p_\theta(z|y))$$

Conditional VAE Architecture

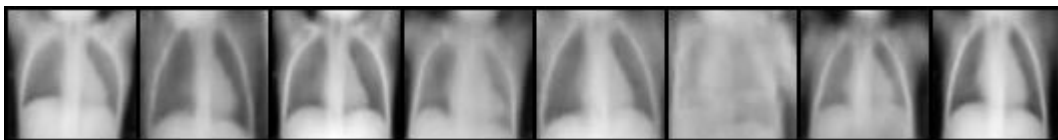


Conditional VAE Results

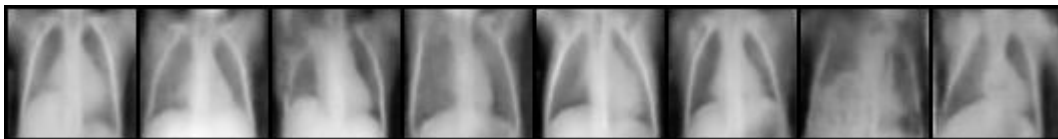
- VAE epoch 1



- VAE epoch 30



- VAE epoch 78



Enhanced Conditional VAE



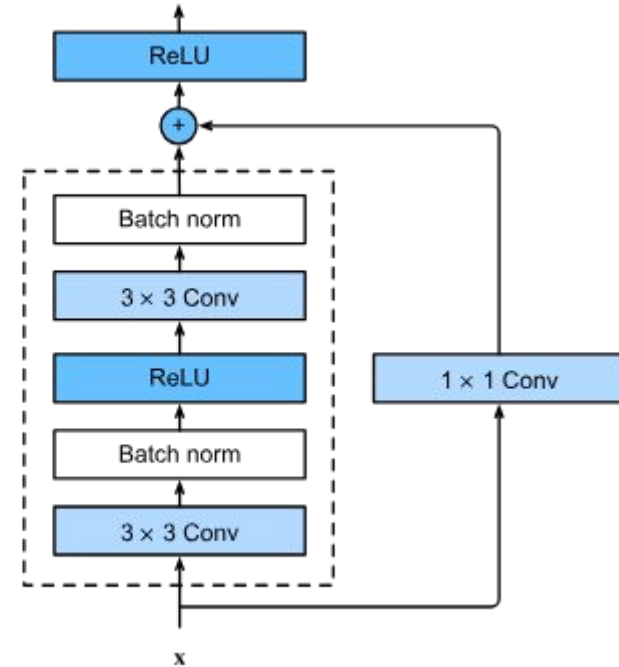
This section defines and trains a more expressive Conditional Variational Autoencoder (**VAE**) that leverages:

- **Residual blocks** in the decoder for sharper image reconstruction.
- **SSIM (Structural Similarity Index)** as a secondary loss term to preserve image structure.
- **β -VAE regularization.**
- **Early stopping**

Residual blocks

Benefits of adding residual blocks:

- Enhanced feature learning
- Prevents vanishing gradients in deep networks
- Improves reconstruction quality and detail preservation



Residual blocks in the decoder



ENHANCED DECODER



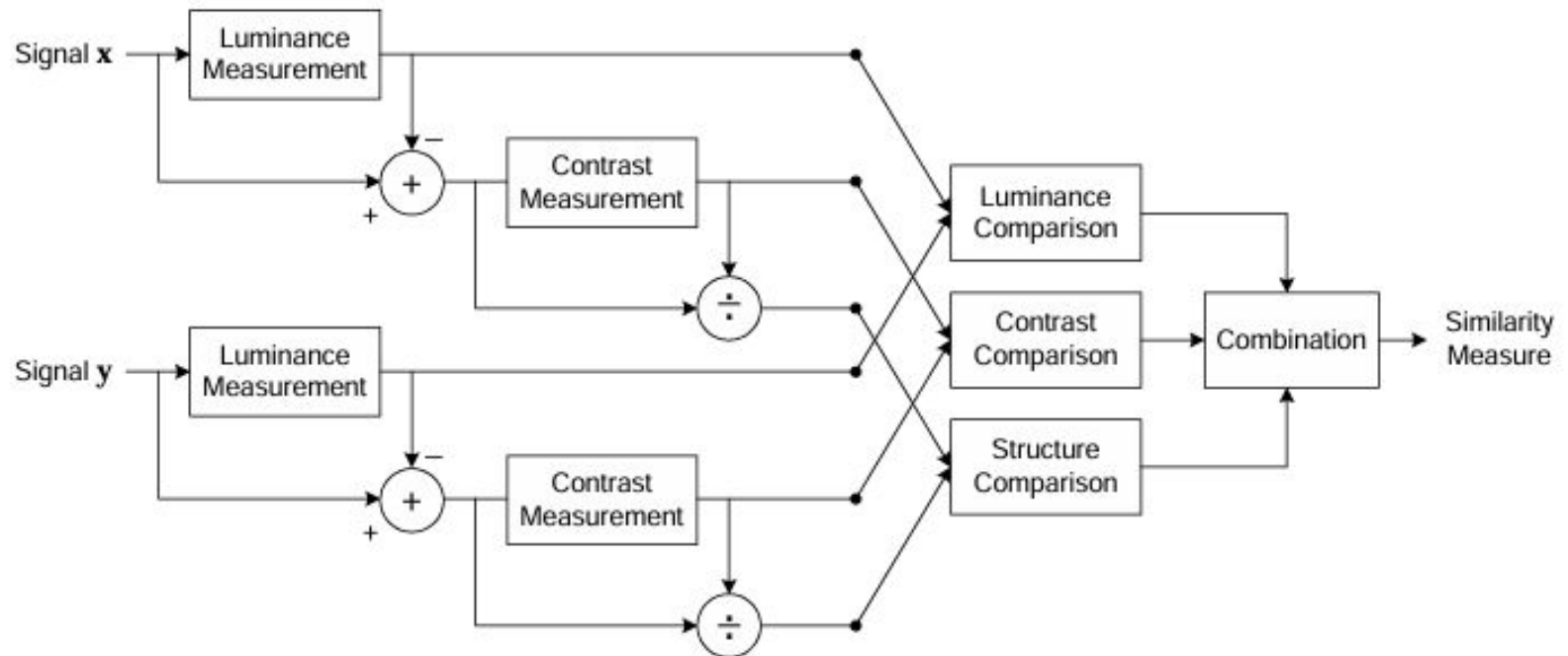
SSIM (Structural Similarity Index Measure)



- SSIM is a perceptual loss that evaluates image similarity based on **structure**, **contrast**, and **luminance** more **aligned with human visual perception** than MSE.
- Compared to **pixel-based losses**, SSIM ensures reconstructions look realistic and diagnostically useful.
- It's used to improve the structural fidelity of generated chest X-rays, especially to preserve features like **lung edges**, which are crucial for medical diagnosis.

$$\text{Loss} = \text{Reconstruction Loss (e.g., MSE)} + \text{KL} + \lambda \cdot (1 - \text{SSIM})$$

SSIM (Structural Similarity Index Measure)



β -VAE



β -VAE regularization

- β is a hyperparameter to control the strength of the regularization term (KL divergence) in the loss function and **scales the KL divergence term** in the VAE loss:

Standard VAE Loss:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q(z|x)}[-\log p(x|z)] + \text{KL}(q(z|x)||p(z))$$

β -VAE Loss:

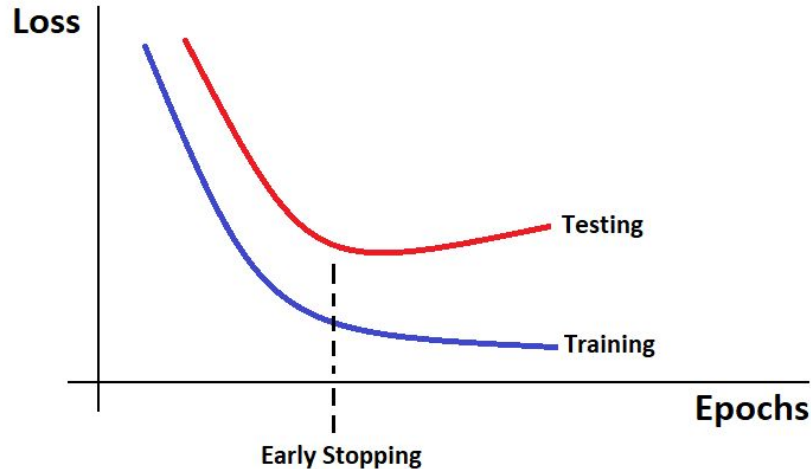
$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q(z|x)}[-\log p(x|z)] + \beta \cdot \text{KL}(q(z|x)||p(z))$$

- When $\beta = 1$, this is just a normal VAE.
- When $\beta > 1$, you increase pressure on the model to align the latent distribution $q(z|x)$ with the prior $p(z)$, usually a standard Gaussian.

Total Loss=Reconstruction Loss (e.g., MSE)+ β *KL+ λ · (1-SSIM)

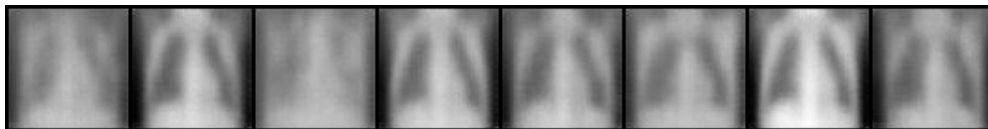
Early stopping

- Early stopping is a technique used during training to prevent overfitting and save some resources while training. It monitors the model's performance, and when performance stops improving for a specified number of epochs (patience).
- In our project we added patience = 3.

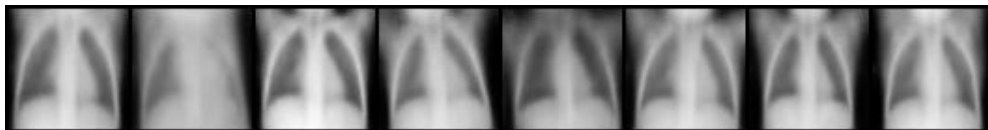


Enhanced Conditional VAE Results

- VAE epoch 1



- VAE epoch 30



- VAE epoch 78

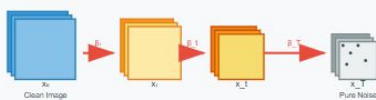


DDPM (Denoising Diffusion Probabilistic Model)

DDPM: Denoising Diffusion Probabilistic Model

T=200 timesteps | β : $1e-4 \rightarrow 0.02$ | Image Size: 128×128 | Time Embedding: 32-dim

Forward Process (Noise Addition)



Sampling Process



Reverse Process (Denoising with UNet)

DDPM Parameters & Formulas

Noise Schedule:

- $\beta_{:T}$ is linear schedule from $1e-4$ to 0.02
- $\alpha_t = 1 - \beta_t$
- $\bar{\alpha}_t = \sum_{s=1}^t \alpha_s$ for $s=1$ to t

Forward Process:

- $q(x_{t+1}|x_t) = N(x_{t+1} | x_t \sqrt{\alpha_{t+1}}, (1 - \alpha_{t+1}) \sigma_t^2)$
- $x_{t+1} = \alpha_{t+1} x_t + \sqrt{1 - \alpha_{t+1}} \epsilon$

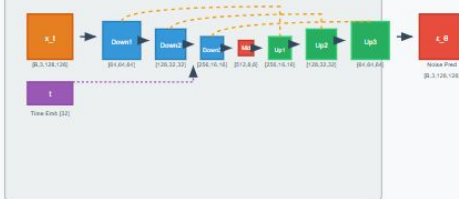
Reverse Process:

- $p_{\theta}(x_t|x_{t+1}) = N(x_t | \mu_{\theta}(x_{t+1}, t), \sigma_{\theta}^2)$
- $\mu_{\theta} = (1 - \alpha_t) \bar{\alpha}_t^{-1} (x_{t+1} - \beta_t \bar{\alpha}_t^{-1} \epsilon_{\theta}(x_{t+1}, t))$

Training Loss:

- $\mathcal{L} = E_{x_0, \epsilon, t} \| \epsilon - \epsilon_{\theta}(x_1, t) \|^2$
- Simple MSE loss between true and predicted noise

UNet Architecture



Legend & Architecture Details:

- UNet Encoder (Downsampling)
- UNet Decoder (Upsampling)
- Bottleneck / Noise Prediction
- Time Embedding (32-dim)
- Skip Connections

Key Features:

- T=200 diffusion timesteps
- Linear noise schedule $\beta \in [1e-4, 0.02]$
- UNet predicts noise $\epsilon_{\theta}(x_{t+1}, t)$
- Time-conditional generation
- Skip connections preserve details
- Sinusoidal time embeddings
- 128x128 image resolution

Training vs Sampling

Training:

- Sample clean image x_0 and timestep t
- Add noise: $x_1 = \alpha_{t+1} x_0 + \sqrt{1 - \alpha_{t+1}} \epsilon$
- Train UNet to predict ϵ : $\mathcal{L} = \| \epsilon - \epsilon_{\theta}(x_1, t) \|^2$

Key Innovation:

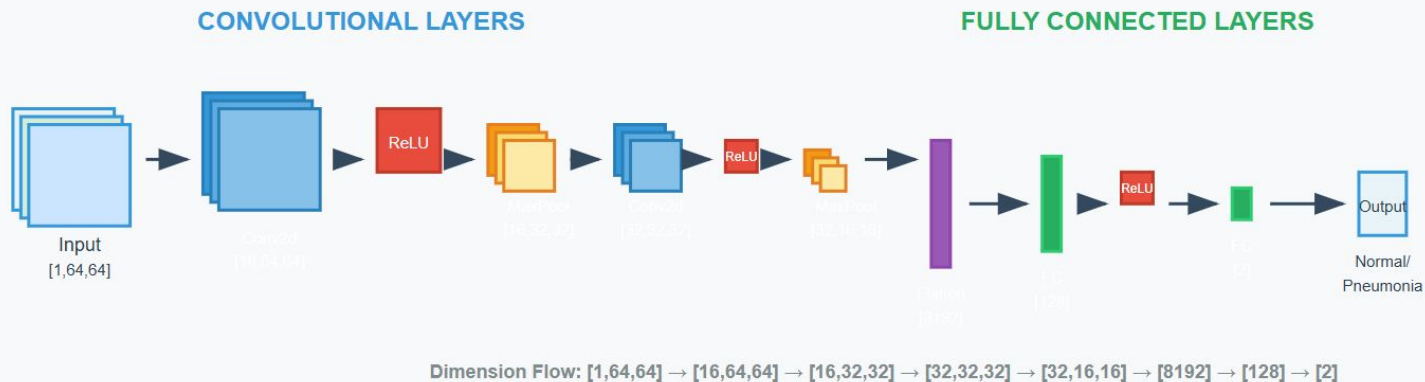
- DDPM learns to predict noise instead of the image directly, which is more stable and produces higher quality results.
- The gradual denoising process allows for fine-grained control and better sample quality than single-step generation.

Sampling:

- Start with pure noise $x_T \sim N(0, I)$
- For $t = T$ to 1: predict $\epsilon_{\theta}(x_t, t)$
- Denoise: $x_{t-1} = \text{denoising_step}(x_t, \epsilon_{\theta}(x_t, t))$
- Output clean image x_0

Complete Flow: Clean Image \rightarrow Forward Diffusion (+ Noise) \rightarrow UNet Training \rightarrow Reverse Sampling (- Noise) \rightarrow Generated Image

CNN Classifier



Legend:

- Conv2d (3×3 kernel, padding=1)
- ReLU Activation
- MaxPool2d (2×2, stride=2)
- Flatten
- Fully Connected

Task: Binary Classification
Classes: Normal/Pneumonia

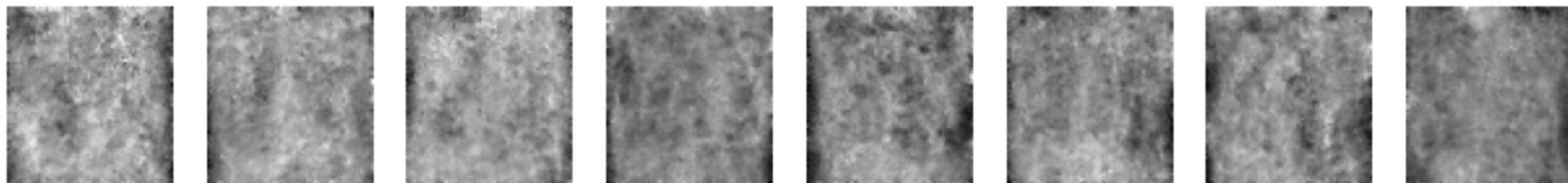
Architecture Details:

- Input: 64×64 grayscale X-ray images
- Conv1: 1→16 channels, maintains spatial size
- Conv2: 16→32 channels, maintains spatial size
- MaxPool: Reduces spatial dimensions by half
- FC1: 8192→128 neurons with ReLU
- FC2: 128→2 neurons (classification output)
- Total Parameters: ~1M parameters

Key Features:

- Simple and fast architecture
- Suitable for medical imaging
- Binary classification output
- Progressive downsampling
- ReLU activations
- Efficient parameter usage

DDMP and CNN results



```
Epoch 1 done.  
Epoch 2 done.  
Epoch 3 done.  
Accuracy: 0.7420  
Precision: 0.7101  
Recall: 0.9923  
F1: 0.8278  
Confusion matrix:  
[[ 76 158]  
 [  3 387]]
```

To address the class imbalance in the chest X-ray dataset, we used a **DDPM (Denoising Diffusion Probabilistic Model)** to generate synthetic images of the minority class ("Normal").

These synthetic images were combined with the real data to create a more balanced training set.

We then trained a **CNN classifier** on the augmented dataset (real + DDPM synthetic) and evaluated its performance on the real test set.

CNN classifier with VAE data

Training classifier on original dataset

Epoch 1/5, Loss: 0.2045

Epoch 2/5, Loss: 0.1022

Epoch 3/5, Loss: 0.0884

Epoch 4/5, Loss: 0.0681

Epoch 5/5, Loss: 0.0598

Accuracy on test set: 0.7484

Training classifier on VAE-augmented dataset

Epoch 1/5, Loss: 0.2599

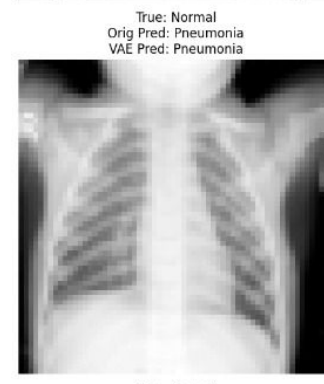
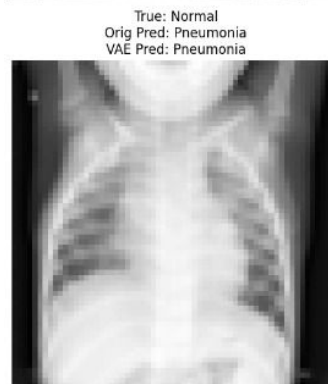
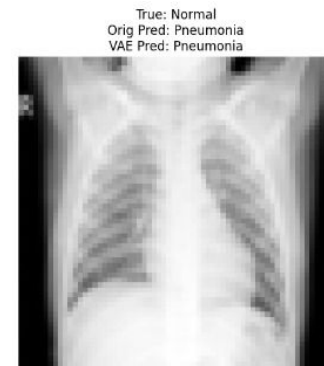
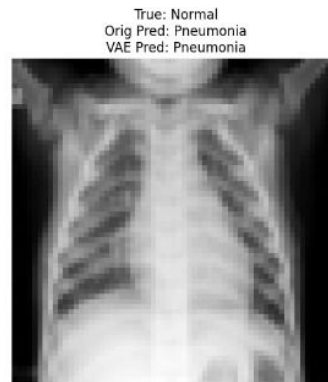
Epoch 2/5, Loss: 0.1167

Epoch 3/5, Loss: 0.0811

Epoch 4/5, Loss: 0.0598

Epoch 5/5, Loss: 0.0500

Accuracy on test set: 0.7532



CNN classifier with Enhanced VAE data

Training classifier on original dataset

Epoch 1/5, Loss: 0.2057

Epoch 2/5, Loss: 0.1214

Epoch 3/5, Loss: 0.0859

Epoch 4/5, Loss: 0.0738

Epoch 5/5, Loss: 0.0624

Original Accuracy: 0.8301

Class 0 (Normal) - Precision: 0.9507, Recall: 0.5769, F1: 0.7181

Class 1 (Pneumonia) - Precision: 0.7946, Recall: 0.9821, F1: 0.8784

Training classifier on VAE-augmented dataset

Epoch 1/5, Loss: 0.2178

Epoch 2/5, Loss: 0.0868

Epoch 3/5, Loss: 0.0569

Epoch 4/5, Loss: 0.0397

Epoch 5/5, Loss: 0.0300

VAE-Augmented Accuracy: 0.7837

Class 0 (Normal) - Precision: 0.9091, Recall: 0.4701, F1: 0.6197

Class 1 (Pneumonia) - Precision: 0.7535, Recall: 0.9718, F1: 0.8488

True: Pneumonia
Orig: Normal
VAE: Pneumonia



True: Pneumonia
Orig: Normal
VAE: Pneumonia



True: Normal
Orig: Normal
VAE: Normal



True: Normal
Orig: Normal
VAE: Normal



Challenges & Issues Encountered



- Severe Class Imbalance
- Computational Constraints
- Small Subsets for Debugging
- Quality of Synthetic Images
- Checkerboard Artifacts
- Posterior Collapse in VAEs (If the latent space was too large or the KLD loss weight too strong, the VAE decoder produced only black or blank images.)
- Hyperparameter Tuning
- No Direct Evaluation Metrics for Generators



Thank you.

