7PAM2021-0105-2024 - Machine Learning and Neural Networks

University of Hertfordshire UH

DENOSING DIFFUSION PROBABILISTIC MODEL

TUTORIAL FOR BEGINNERS

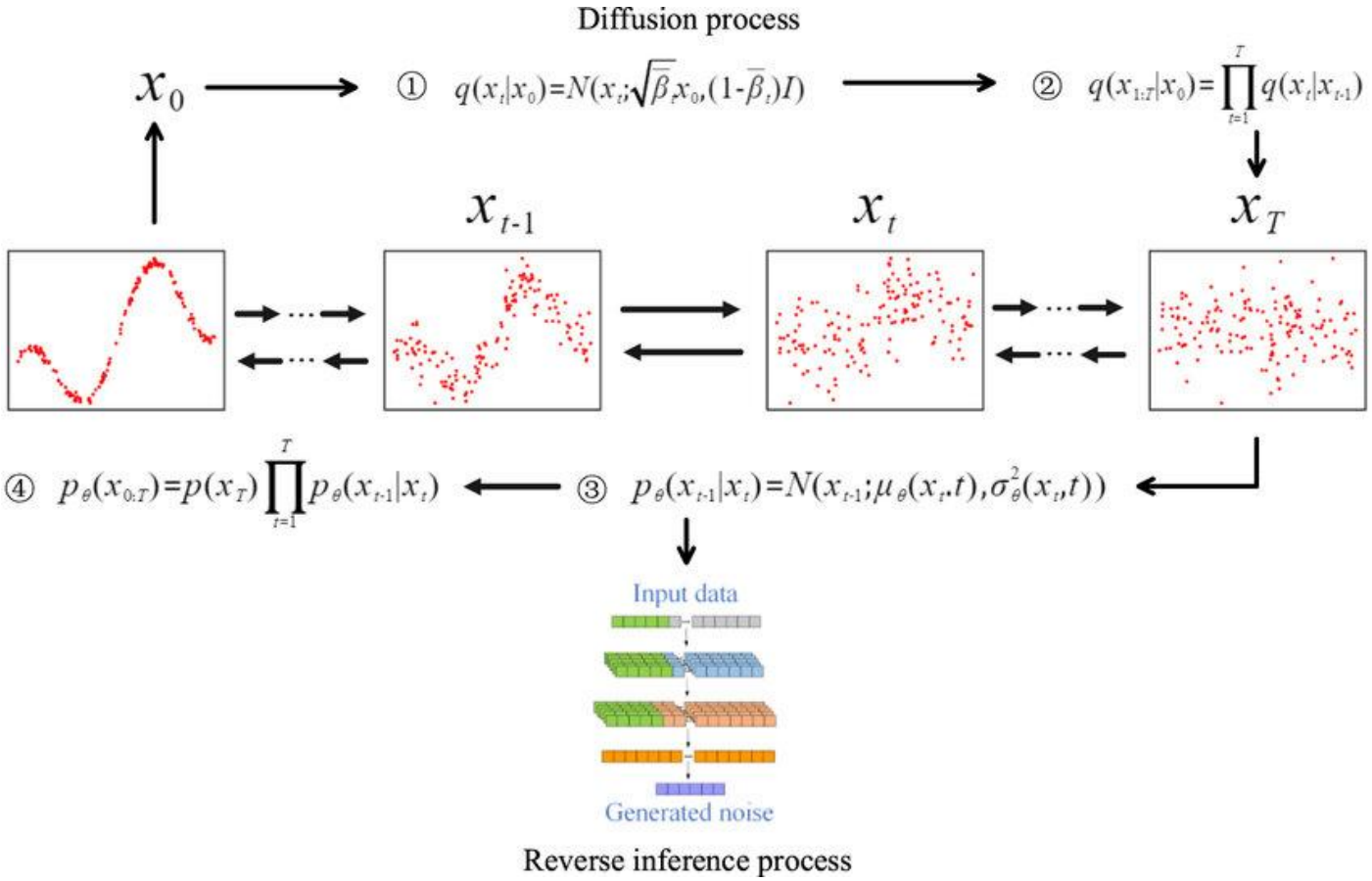
SUBMITTED BY

KEERTHIVASAN KANNAN

(23052162)

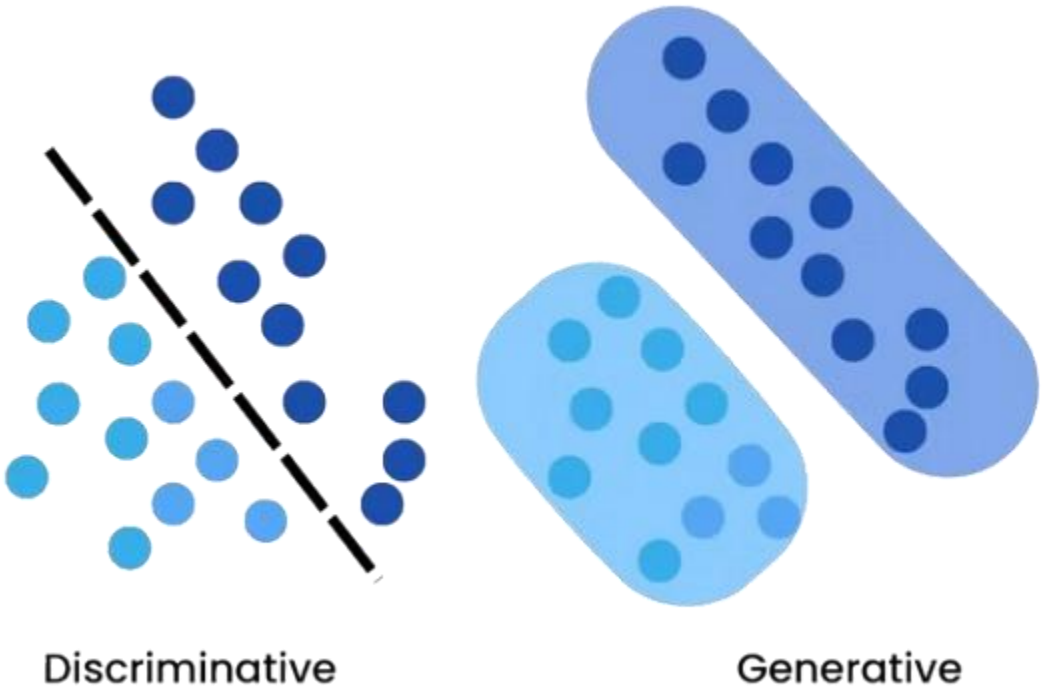
INTRODUCTION

Denoising Diffusion Probabilistic Models (DDPMs) present an evolutionary AI technique which converts random noise into realistic high-detail images along with complex auditory outputs and various other data types. DDPMs execute this transformation by designing a step-based method which starts with noise addition followed by noise removal until unclear information turns into exact meaningful results. It differentiate themselves through their integration of deep learning approaches alongside probabilistic framework which leads to dependable lifelike output results. Also, it implements a basic yet influential concept by addition of structured data noise increments followed by a reverse procedure. The process represents how people naturally operate through focusing a blurry photo or readjusting out-of-focus lenses until everything appears clear. DDPMs utilize incremental learning procedures that help them determine the original data under noisy conditions. The simple design of DDPM models together with their precise outcome quality makes them suitable for applications that include realistic image generation as well as audio creation and AI creative tasks. The model delivers specific guided modeling which makes it operationally adaptable and efficient for implementation across practical and creative fields.



WHAT ARE GENERATIVE MODELS?

AI algorithms under the classification of generative models produce original content through their ability to learn patterns that exist within datasets. The core difference between discriminative models and generative models appears in their operation methods. Because, discriminative models determine classifications while generative models generate new data entries that replicate source data features. The analysis of huge data sets produces patterns which the system translates into new authentic results. Generative models function like comparable real-life models can illustrate their operation. A system that analyzes multiple arts pieces from one artist will learn to recognize and retain fundamental elements such as the usage of specific colors and brush movements and thematic materials. The trained generative model develops capacity to produce entirely new paintings with distinctive qualities of a specific artist so vivid that they match his genuine artistic style even though these original paintings never actually existed. The system's functionality reaches beyond visual elements to produce music and literature together with realistic human vocalization. These models demonstrate an innovative capability which sets them apart from predictive and analytical AI approaches because they generate new data creatively instead of processing only existing information. Vast potential exists in fields such as entertainment and design along with art and synthetic data generation for training other AI models because of generative models.



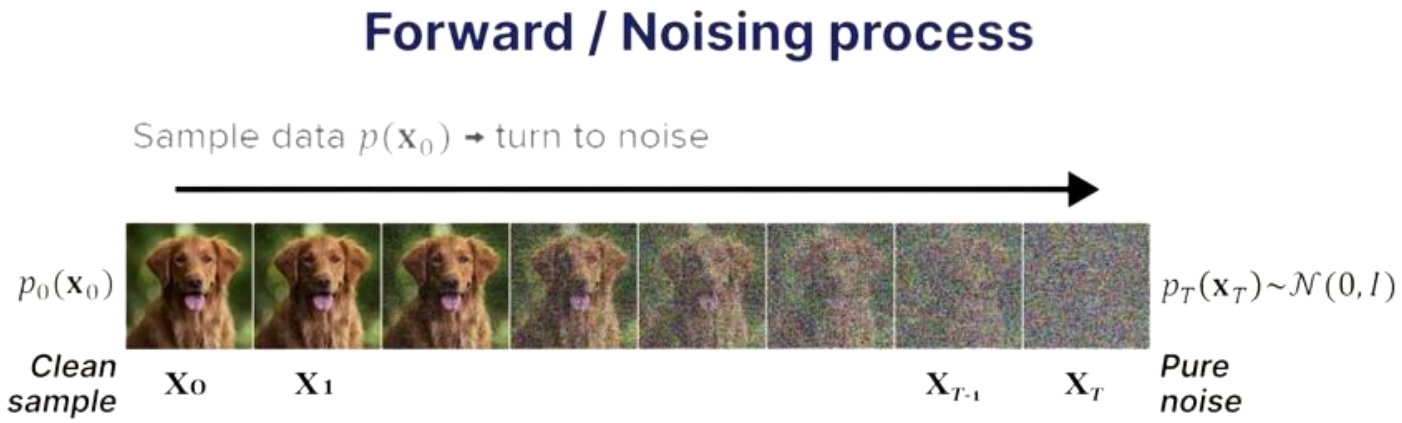
DIFFUSION MODELS

Diffusion models apply a straightforward yet powerful natural manifestation of physical diffusion processes found in nature to create their mechanism. Data structured information goes through a gradual process of randomization that allows recovering original content via reversal. The process of ink mixing with water provides a helpful illustration where organized ink patterns exist during initial stages. Timely exposure while stirring water causes the ink to spread uniformly until it transforms into an unstructured homogenous state. While working with diffusion models the structured data which includes clear images or audio signals undergoes carefully controlled gradual addition of noise. Successive steps guide the data from readable patterns through an incremental process until it ends in total randomness. The main advantage of diffusion models consists of their accurate ability to undo the process of data diffusion. A systematic reconstruction of original data becomes possible because these models receive detailed training about the methods that noise has affected the source information. The model uses reversed diffusion techniques to eliminate noise step by step until it returns data to its structured original form from random patterns. The step-by-step elimination of imperfections that diffusion models conduct follows natural editing techniques used by humans which demonstrates their ability to operate intuitively. These models generate highly realistic and high-quality results for different generative tasks because of their exceptional effectiveness.

DDPM PROCESSES

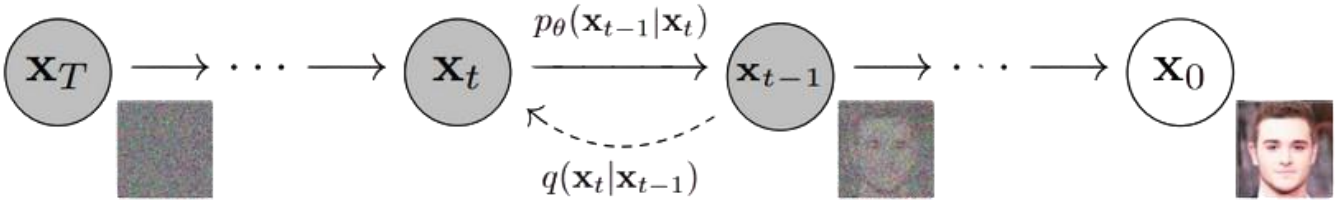
FORWARD DIFFUSION PROCESS:

Through its forward diffusion mechanism the process adds controlled randomization to clean data which leads to the complete loss of original content in both images and structured information. The introduction of minor random noises occurs with fine control to progressively obscure the original data at each step of the procedure. The original data structure sustains at first but details progressively disappear as the process runs until the data shares no distinguishable features from randomly generated noise. The addition of noise occurs gradually based on mathematical control that determines the speed and intensity of noise introduction. Controlling noise emissions via this process creates accurate outcomes because it shows predictable behavior during each production cycle. The structured data transforms into random data through the forward diffusion process which delivers a systematic workflow.



REVERSE DIFFUSION PROCESS:

Neural networks perform the reverse diffusion process step by step through training for denoising. A neural network receives entirely random data which it transforms gradually into the original structure by repeating noise subtraction between each iteration. The network undergoes intensive training which allows it to identify precise patterns needed for reversing the diffusion process and achieve successful data restoration. Each reverse step uses previously acquired knowledge about data appearance from the more white-noisy step. Multiple iterations of operation allow the neural network to eliminate noise step by step until it reveals the original data with clarity. The neural network technology's outstanding pattern detection abilities make this recovery method effective in producing authentic representations of original information.



MATH BEHIND DDPM

The mathematical framework of DDPM delivers operational power for this fundamental generative model to function. Mathematical probabilistic equations explain the forward and reverse diffusion procedures. The forward diffusion step is defined by a conditional probability distribution that systematically adds noise:

$$q(x_t|x_{t-1}) = \aleph(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I$$

The data at timestep t is denoted by x_t and the earlier data point at timestep t is shown as x_{t-1}. The parameter β_t affects the amount of noise introduced in successive increments as it determines the speed of data transformation from original to random values. Smooth gradual evolution in the diffusion process becomes possible due to this controlled method.

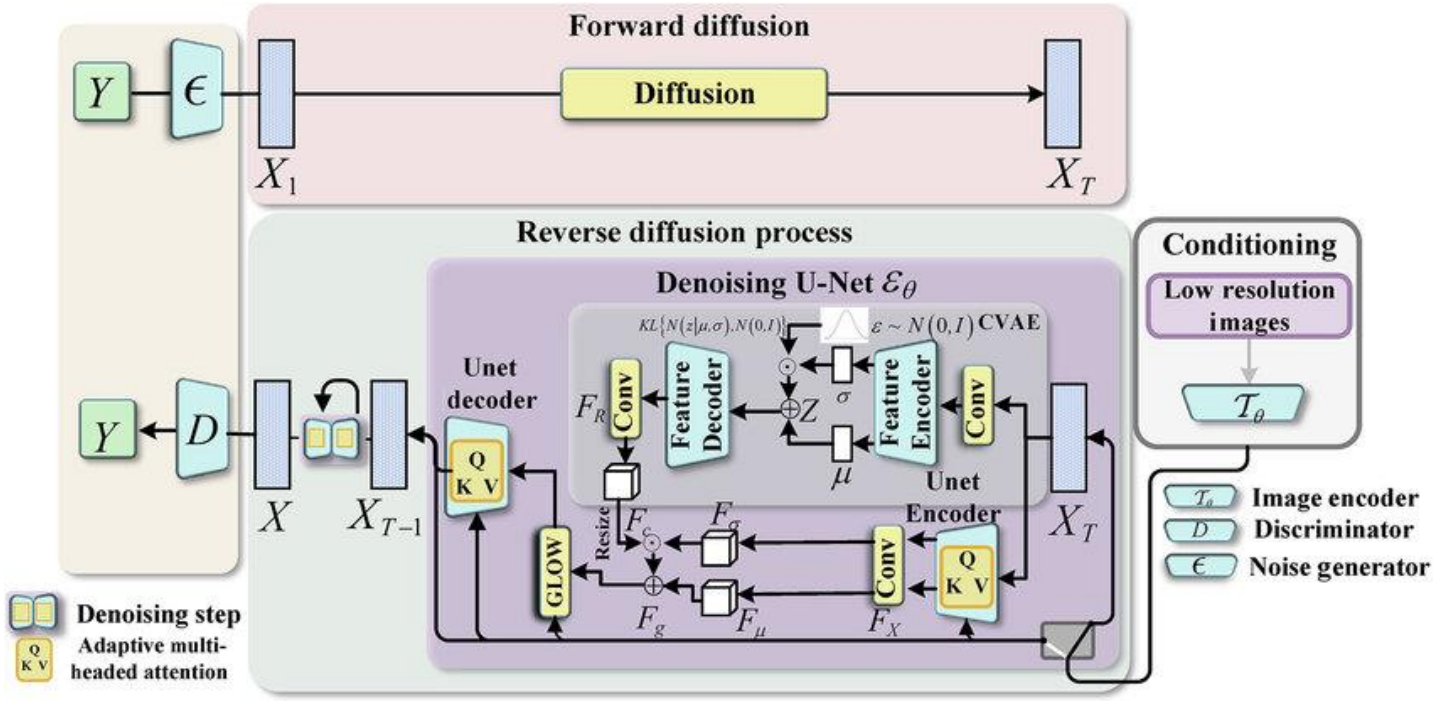
During reverse diffusion the process attempts to convert noisy data elements back into their clear original state. The reverse calculation follows another probability distribution which can be represented mathematically:

$$p_\theta(x_{t-1}|x_t) = \aleph(x_{t-1};\ \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

The previously predicted clean data mean value is denoted by $\mu_\theta(x_t, t)$ while the predicted variance value is indicated by $\Sigma_\theta(x_t, t)$ through a trained neural network. Through its predictions the model can successfully de-noise data through an organized noise removal process.

NEURAL NETWORKS IN DDPM

DDPMs utilize neural networks as their fundamental component that executes the important task of noise reduction from data throughout their reverse diffusion operations. Within DDPMs neural networks operate as advanced detectors that examine deteriorated information to perform system-level defect elimination. The DDPM neural network develops the capability to forecast statistical restoration parameters necessary for noise reduction in data by understanding the mean and variance distributions at various diffusion stages. Through numerous training sessions with big data the neural network develops profound expertise about various types of information that lie beneath varying degrees of noise. The network trains its restoration abilities through this process until it reaches a point of effectiveness for identifying and recovering clear structural information during the noise reversal process. The success of DDPM heavily depends on selecting an appropriate neural network topology. Two principal model architectures exist for DDPM generation including convolutional neural networks for images and transformer models for textual and audio operations. The architecture selection defines the strengths of different systems that ensure effective pattern extraction for complex denoising operations. The predictive accuracy of networks depends directly on proper design and training practices since they determine the quality level of generated final outputs by DDPM.



IMPLEMENTATION OF DDPM

Building a DDPM consists of educating neural networks to progressively extract noise from data samples. The training process for DDPM will be dissected through clear steps which are supported by basic Python code examples.

STEP 1: DATASET PREPARATION

The initial process gathers a clean dataset with numerous high-quality examples including images, audio files or textual data. Our analysis implies image data as the sole dataset for this discussion.

```
import torchvision
import torchvision.transforms as transforms
from torch.utils.data import DataLoader

transform = transforms.Compose(
    [transforms.Resize((64, 64)),
     transforms.ToTensor(),])
dataset = torchvision.datasets.CIFAR10(
    root='./data', train=True,
    download=True, transform=transform)
data_loader = DataLoader(dataset,
            batch_size=128, shuffle=True)
```

Before training occurs the model performs image preprocessing by converting images to tensors while adjusting their size.

STEP 2: ADDING NOISE

In each progression of forward diffusion steps the introduction of noise increases. Gaussian noise is introduced by small amounts at each progression step for images.

```
def add_noise(x, t, betas):
    sqrt_alpha = torch.sqrt(1 - betas[t])
    sqrt_beta = torch.sqrt(betas[t])
    noise = torch.randn_like(x)
    noisy_x = sqrt_alpha * x + sqrt_beta * noise
    return noisy_x, noise
```

x is the original clean image. **t** represents the timestep or diffusion step. **betas** is a predefined schedule controlling the amount of noise added at each step.

STEP 3: DEFINING THE NEURAL NETWORK

The neural system of DDPM operates to forecast the introduced noise data. Images benefit from the effective architecture of a convolutional neural network (CNN).

```
class SimpleDenoiseCNN(nn.Module):
    def __init__(self):
        super(SimpleDenoiseCNN, self).__init__()
        self.network = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=3, padding=1),
            nn.ReLU(),
            nn.Conv2d(64, 64, kernel_size=3, padding=1),
            nn.ReLU(),
            nn.Conv2d(64, 3, kernel_size=3, padding=1),
        )

    def forward(self, x, t):
        return self.network(x)
```

This network takes noisy images (x) as input and outputs a prediction of the noise originally added.

STEP 4: TRAINING THE NEURAL NETWORK

The model consumes distorted images during training sessions that require it to forecast the corruption introduced to the images. A calculation of loss occurs through the comparison of predicted against actual noise values.

```
import torch.optim as optim
model = SimpleDenoiseCNN()
optimizer = optim.Adam(model.parameters(), lr=1e-4)
criterion = nn.MSELoss()
T = 1000
betas = torch.linspace(0.0001, 0.02, T)
epochs = 10
for epoch in range(epochs):
    for images, _ in data_loader:
        optimizer.zero_grad()
        t = torch.randint(0, T, (images.size(0),), dtype=torch.long)
        noisy_images, noise = add_noise(images, t, betas)
        predicted_noise = model(noisy_images, t)
        loss = criterion(predicted_noise, noise)
        loss.backward()
        optimizer.step()
    print(f"Epoch {epoch+1}/{epochs}, Loss: {loss.item():.4f}")
```

The system distributes every image batch into randomly selected time increments called timesteps. The generated noise matches the specification of timestep t. Prediction of this noise occurs during the model processing of noisy images. Loss calculations between predicted noise and actual noise serve as the main direction for the model to enhance its ability to remove noise from inputs.

STEP 5: SAMPLING NEW IMAGES

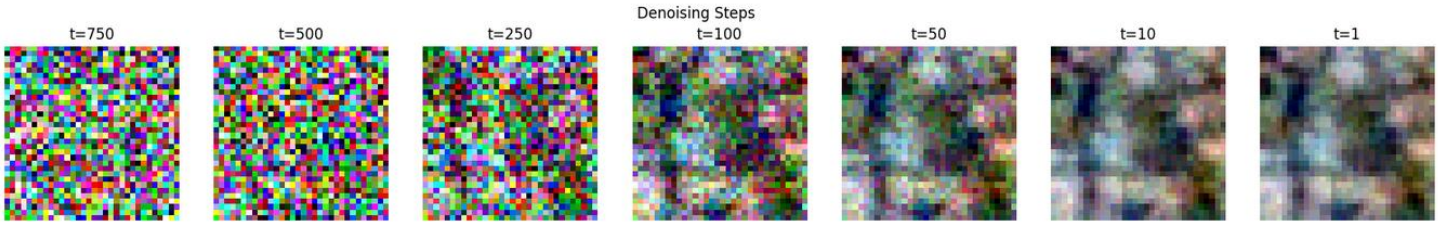
Once the model reaches completion its reverse diffusion process creates new data by successively eliminating noise from random input.

```python
@torch.no_grad()
def sample_image(model, betas, image_shape):
    model.eval()
    x = torch.randn(image_shape)  # Start from pure noise
    T = len(betas)

    for t in reversed(range(T)):
        predicted_noise = model(x, torch.tensor([t]))
        alpha = 1 - betas[t]
        x = (x - betas[t] / torch.sqrt(1 - alpha) * predicted_noise) / torch.sqrt(alpha)
        if t > 0:
            noise = torch.randn_like(x)
            x += torch.sqrt(betas[t]) * noise

    return x

generated_image = sample_image(model, betas, (1, 3, 64, 64))
```



REAL WORLD APPLICATIONS

Accepting DDPMs as powerful tools to solve various complex real-world challenges in numerous fields has occurred quickly. They demonstrate substantial practical versatility because they enable realistic media creation and help with critical medical diagnosis.

- Image and video generation
- Medical Imaging
- Text-to-Image Synthesis
- Enhancing Data Privacy
- Audio Generation and Enhancement
- Anomaly Detection

OVERALL UNDERSTANDING YOU CAN GET

- Understanding the systematic procedure of noise annotation and removal in DDPM produces a complete understanding regarding their revolutionary approach to generative modeling.
- Having a clear differentiation between DDPM and GANs and VAEs by explaining DDPM's probabilistic framework with its denoising process.
- The ability to comprehend mathematical aspects of diffusion processes including forward and reverse diffusion models improves understanding of their internal operational dynamics.
- An understanding of neural networks in DDPM framework helps researchers make better design decisions for the models and their noise prediction systems.
- The real world applications of insight demonstrate how GPT-3 functions across diverse fields which include picture and audio production and medical diagnosis and anomaly identification and privacy protection.
- Experiential learning of DDPM implementation required students to perform data preparation and noise scheduling and network training and sampled new data from trained models.
- The ability to evaluate both advantages and disadvantages of DDPMs brings informed decision making when developing new solutions and implementing models for various situations.

REFERENCES

- *Denoising Diffusion Probabilistic Models*Jonathan Ho, Ajay Jain, Pieter Abbeel (2020)
- Official documentation for PyTorch framework
- Torchvision Documentation: For datasets, models, and transforms
- NVIDIA CUDA documentation for enabling GPU support
- Alex Krizhevsky, Learning Multiple Layers of Features from Tiny Images (2009)
- *U-Net: Convolutional Networks for Biomedical Image Segmentation* Olaf Ronneberger, Philipp Fischer, Thomas Brox (2015)
- Hugging Face Diffusers Library
- Andreas Lugmayr's DDPM Implementation

GITHUB LINK : **INDIVIDUAL ASSIGNMENT MLNN**