R.M.K GROUP OF ENGINEERING INSTITUTIONS

R.M.K
GROUP OF INSTITUTIONS

R.M.K
GROUP OF
INSTITUTIONS

R.M.K GROUP OF INSTITUTIONS

Please read this disclaimer before proceeding:

This document is confidential and intended solely for the educational purpose of RMK Group of Educational Institutions. If you have received this document through email in error, please notify the system manager. This document contains proprietary information and is intended only to the respective group / learning community as intended. If you are not the addressee you should not disseminate, distribute or copy through e-mail. Please notify the sender immediately by e-mail if you have received this document by mistake and delete this document from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.

Digital Course Material
22AI005
Introduction to Generative AI

Department : Information Technology

Batch/Year : 2022-2026/IV

Created by : Ms.G.K.Monica

Date : 20-08-2025

R.M.K GROUP OF INSTITUTIONS

TABLE OF CONTENTS

COURSE OBJECTIVES

The Course will enable learners to:

- To understand the basic concepts of Generative AI.
- To build Generative AI systems to generate images.
- To understand the concept used in Generative AI Models.
- To use various Generative AI models.
- To compare and use the various Large Language Models.
- To understand the basics of Prompt Engineering.

PRE REQUISITES

- **PRE-REQUISITE CHART**

22AI005 INTRODUCTION TO GENERATIVE AI

INTRODUCTION TO GENERATIVE AI

L T P C

3 0 0 3

OBJECTIVES:

- To understand the basic concepts of Generative AI.
- To build Generative AI systems to generate images.
- To understand the concept used in Generative AI Models.
- To use various Generative AI models.
- To compare and use the various Large Language Models.
- To understand the basics of Prompt Engineering.

UNIT I INTRODUCTION **9**

Generative Models – Image transformation – Challenges - Deep Neural Networks – Perceptron – back propagation – CNN – RNN – Optimizer.

UNIT II IMAGE GENERATION **9**

Creating encodings of images – variational objective – Inverse Autoregressive flow – Importing CIFAR – Creating the network from TensorFlow 2.

UNIT III GENERATIVE ADVERSARIAL NETWORKS **9**

Generative Adversarial Networks – Vanilla GAN – Improved GANs – Progressive GAN – Challenges – Paired style transfer – Unpaired style transfer – Deepfakes – Modes of operation – key feature set – High level flow – Replacement – Re-enactment.

UNIT IV LARGE LANGUAGE MODELS **9**

Overview of LLMs - Transformers – GPT – Types of LLMs – Key concepts – other Transformers – T5 – Generative Pre-Training Models – Multi-modal Models – DALL.E 2

UNIT V PROMPT ENGINEERING **9**

Basics – In-Context Learning – In-Context Prompting – Techniques – Image Prompting – Prompt Hijacking – Challenges.

TOTAL: 45 PERIODS

COURSE OUTCOME

Course Code	Course Outcome Statement	Cognitive/ Affective Level of the Course Outcome	Expected Level of Attainment
Course Outcome Statements in Cognitive Domain			
C305.1	Elaborate the basic concepts of Generative AI	Understand K2	60%
C305.2	Build Generative AI systems to generate images	Analyse K4	60%
C305.3	Apply the concepts used in Generative AI Models	Apply K3	60%
C305.4	Use various Generative AI models.	Apply K3	60%
C305.5	Compare and use the various Large Language Models	Analyse K4	60%
C305.6	Analyze the basics of Prompt Engineering.	Apply K3	60%
Course Outcome Statements in Affective domain			
C305.7	Attend the classes regularly	Respond (A2)	95%
C305.8	Submit the Assignments regularly.	Respond (A2)	95%
C305.9	Participation in Seminar/Quiz/ Group Discussion/ Collaborative learning and content beyond syllabus	Valuing (A3)	95%

CO-PO/PSO MAPPING

Correlation Matrix of the Course Outcomes to Programme Outcomes and Programme Specific Outcomes Including Course Enrichment Activities

Course Outcomes (Cos)		Programme Outcomes (POs), Programme Specific Outcomes (PSOs)														
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
		K3	K4	K5	K5	K3/K5	A2	A3	A3	A3	A3	A3	A2	K3	K3	K3
C305.1	K3	3	2	1	1	3						3		3	3	3
C305.2	K4	3	3	2	2	3						3		3	3	3
C305.3	K2	2	1									2		3	3	3
C305.4	K3	3	2	1	1	3						3		3	3	3
C305.5	K4	3	3	2	2	3						3		3	3	3
C305.6	K3	3	2	1	1	3						3		2	2	2
C305.7	A2												3			
C305.8	A2								2	2	2		3			
C305.9	A3						3	3		3	3		3			
C305		3	3	2	2	3	1	1	1	3	3	3	3	3	3	3

UNIT III
GENERATIVE ADVERSARIAL NETWORKS

LECTURE PLAN – UNIT III

	UNIT III GENERATIVE ADVERSARIAL NETWORKS						
Sl. No	TOPIC	NO OF PERIODS	PROPOSED LECTURE PERIOD	ACTUAL LECTURE PERIOD	PERTAINING CO(s)	TAXONOMY LEVEL	MODE OF DELIVERY
1	Generative Adversarial Networks	1			CO1	K2	PPT
2	Vanilla GAN – Improved GANs	1			CO1	K2	PPT
3	Progressive GAN – Challenges	1			CO1	K2	PPT
4	Paired style transfer – Unpaired style transfer	1			CO1	K3	PPT
5	Deepfakes	1			CO1	K3	PPT
6	Modes of operation – key feature set	1			CO1	K2	PPT
7	High level flow	1			CO1	K2	PPT
8	Replacement	1			CO1	K2	PPT
9	Re-enactment.	1			CO1	K2	PPT

LECTURE PLAN – UNIT III

ASSESSMENT COMPONENTS

- AC 1. Unit Test AC 2. Assignment
- AC 3. Course Seminar AC 4. Course Quiz
- AC 5. Case Study
- AC 6. Record Work
- AC 7. Lab / Mini Project
- AC 8. Lab Model Exam AC 9. Project Review

MODE OF DELEIVERY

MD 1. Oral presentation MD 2. Tutorial
MD 3. Seminar
MD 4 Hands On MD 5. Videos MD 6. Field Visit

TEST YOURSELF

1. What is the primary objective of a Generative Adversarial Network (GAN)?

a)Image classification

b)Image generation

c)Text summarization

d)Text translation

2. What are the two main components of a GAN?

a)Generator and encoder

b)Discriminator and encoder

c)Generator and discriminator

d)Encoder and discriminator

3. What is the training process in a GAN called?

a)Supervised learning

b)Reinforcement learning

c)Unsupervised learning

d)Adversarial learning

4. Which loss function is commonly used in GANs?

a)Cross-entropy loss

b)Mean squared error loss

c)Binary logistic loss

d)Kullback-Leibler divergence

5. Which type of GAN is designed to generate samples conditioned on specific input information?

a)Unconditional GAN

b)Wasserstein GAN

c)Progressive GAN

d)Conditional GAN

3.1.GAN(Generative Adversarial Network)

GAN(Generative Adversarial Network) represents a cutting-edge approach to generative modeling within deep learning, often leveraging architectures like convolutional neural networks. The goal of generative modeling is to autonomously identify patterns in input data, enabling the model to produce new examples that feasibly resemble the original dataset.

What is a Generative Adversarial Network?

Generative Adversarial Networks (GANs) are a powerful class of neural networks that are used for an unsupervised learning. GANs are made up of two neural networks, **a discriminator and a generator.** They use adversarial training to produce artificial data that is identical to actual data.

- The Generator attempts to fool the Discriminator, which is tasked with accurately distinguishing between produced and genuine data, by producing random noise samples.
- Realistic, high-quality samples are produced as a result of this competitive interaction, which drives both networks toward advancement.
- GANs are proving to be highly versatile artificial intelligence tools, as evidenced by their extensive use in image synthesis, style transfer, and text-to-image synthesis.
- They have also revolutionized generative modeling.

Through adversarial training, these models engage in a competitive interplay until the generator becomes adept at creating realistic samples, fooling the discriminator approximately half the time. Generative Adversarial Networks (GANs) can be broken down into three parts:

- **Generative:** To learn a generative model, which describes how data is generated in terms of a probabilistic model.

- **Adversarial:** The word adversarial refers to setting one thing up against another. This means that, in the context of GANs, the generative result is compared with the actual images in the data set. A mechanism known as a discriminator is used to apply a model that attempts to distinguish between real and fake images.
- **Networks:** Use deep neural networks as artificial intelligence (AI) algorithms for training purposes.

3.1.1.Types of GANs

There are several types of GANs (Generative Adversarial Networks), each designed to address specific challenges or improve upon the basic GAN architecture. Here are the most common types

1. Vanilla GAN:

- This is the simplest type of GAN. Here, the Generator and the Discriminator are simple a basic multi-layer perceptrons.
- In vanilla GAN, the algorithm is really simple, it tries to optimize the mathematical equation using stochastic gradient descent.

2. Conditional GAN (CGAN):

CGAN can be described as a deep learning method in which **some conditional parameters are put into place**.

- In CGAN, an additional parameter 'y' is added to the Generator for generating the corresponding data.
- Labels are also put into the input to the Discriminator in order for the Discriminator to help distinguish the real data from the fake generated data.

3. Deep Convolutional GAN (DCGAN):

DCGAN is one of the most popular and also the most successful implementations of GAN. It is composed of ConvNets in place of multi-layer perceptrons.

- The ConvNets are implemented without max pooling, which is in fact replaced by convolutional stride.
- Also, the layers are not fully connected.

4. **Laplacian Pyramid GAN (LAPGAN):**

- The LAPGAN is a linear invertible image representation consisting of a set of band-pass images, spaced an octave apart, plus a low-frequency residual.
- This approach uses multiple numbers of Generator and Discriminator networks and different levels of the Laplacian Pyramid.
- This approach is mainly used because it produces very high-quality images. The image is down-sampled at first at each layer of the pyramid and then it is again up-scaled at each layer in a backward pass where the image acquires some noise from the Conditional GAN at these layers until it reaches its original size.

5. **Super Resolution GAN (SRGAN):**

- SRGAN as the name suggests is a way of designing a GAN in which a deep neural network is used along with an adversarial network in order to produce higher-resolution images.
- This type of GAN is particularly useful in optimally up-scaling native low-resolution images to enhance their details minimizing errors while doing so.

3.1.2.Architecture of GANs

A Generative Adversarial Network (GAN) is composed of two primary parts, which are the Generator and the Discriminator.

Generator Model

- A key element responsible for creating fresh, accurate data in a Generative Adversarial Network (GAN) is the generator model.

- The generator takes random noise as input and converts it into complex data samples, such text or images. It is commonly depicted as a deep neural network.
- The training data's underlying distribution is captured by layers of learnable parameters in its design through training.
- The generator adjusts its output to produce samples that closely mimic real data as it is being trained by using backpropagation to fine-tune its parameters.
- The generator's ability to generate high-quality, varied samples that can fool the discriminator is what makes it successful.

Generator Loss

- The objective of the generator in a GAN is to produce synthetic samples that are realistic enough to fool the discriminator.
- The generator achieves this by minimizing its loss function JG*JG*. The loss is minimized when the log probability is maximized, i.e., when the discriminator is highly likely to classify the generated samples as real.
- The following equation is given below:

$$J_G = -\frac{1}{m}\sum_{i=1}^{m} logD(G(z_i))$$

Where,

- J_G measure how well the generator is fooling the discriminator.
- $logD(G(z_i))$ represents log probability of the discriminator being correct for generated samples.
- The generator aims to minimize this loss, encouraging the production of samples that the discriminator classifies as real $logD\big(G(z_i)\big)$,close to 1.

Discriminator Model

- An artificial neural network called a discriminator model is used in Generative Adversarial Networks (GANs) to differentiate between generated and actual input.
- By evaluating input samples and allocating probability of authenticity, the discriminator functions as a binary classifier.
- Over time, the discriminator learns to differentiate between genuine data from the dataset and artificial samples created by the generator. This allows it to progressively hone its parameters and increase its level of proficiency.

Discriminator Loss

- The discriminator reduces the negative log likelihood of correctly classifying both produced and real samples.
- This loss incentivizes the discriminator to accurately categorize generated samples as fake and real samples with the following equation:

$$J_D = -\frac{1}{m}\sum_{i=1}^{m} logD(x_i) - \frac{1}{m}\sum_{i=1}^{m} \log(1-D(G(z_i)))$$

- J_D assesses the discriminator's ability to discern between produced and actual samples.
- The log likelihood that the discriminator will accurately categorize real data is represented by $logD(x_i)$
- The log chance that the discriminator would correctly categorize generated samples as fake is represented by $\log(1 - D(G(z_i)))$
- The discriminator aims to reduce this loss by accurately identifying artificial and real samples.

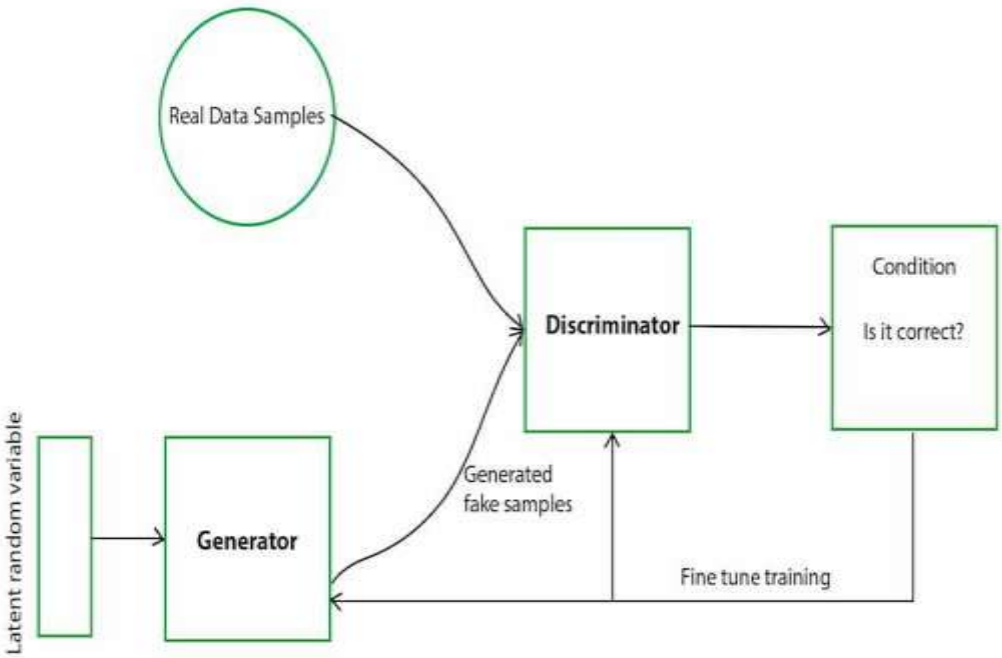
MinMax Loss

In a Generative Adversarial Network (GAN), the minimax loss formula is provided by:

$$min_G\, max_D(G, D) = E_{x \sim p_{data}}[logD(x)] + E_{Z \sim p_z(z)}\left[\log\left(1 - D\left(g(z)\right)\right)\right]$$

Where,

- G is generator network and is D is the discriminator network
- Actual data samples obtained from the true data distribution p_{data} (x) are represented by x.
- Random noise sampled from a previous distribution $p_z(z)$ (usually a normal or uniform distribution) is represented by z.
- D(x) represents the discriminator's likelihood of correctly identifying actual data as real.
- D(G(z)) is the likelihood that the discriminator will identify generated data coming from the generator as authentic.



How does a GAN work?

The steps involved in how a GAN works:

Initialization:

Two neural networks are created: a Generator (G) and a Discriminator (D).

- G is tasked with creating new data, like images or text, that closely resembles real data.
- D acts as a critic, trying to distinguish between real data (from a training dataset) and the data generated by G.

Generator's First Move:

- G takes a random noise vector as input. This noise vector contains random values and acts as the starting point for G's creation process.
- Using its internal layers and learned patterns, G transforms the noise vector into a new data sample, like a generated image.

Discriminator's Turn:

D receives two kinds of inputs:

- Real data samples from the training dataset.
- The data samples generated by G in the previous step. D's job is to analyze each input and determine whether it's real data or something G cooked up. It outputs a probability score between 0 and 1. A score of 1 indicates the data is likely real, and 0 suggests it's fake.

The Learning Process:

Now, the adversarial part comes in:

- If D correctly identifies real data as real (score close to 1) and generated data as fake (score close to 0), both G and D are rewarded to a small degree. This is because they're both doing their jobs well.

- However, the key is to continuously improve. If D consistently identifies everything correctly, it won't learn much. So, the goal is for G to eventually trick D.

Generator's Improvement:

- When D mistakenly labels G's creation as real (score close to 1), it's a sign that G is on the right track. In this case, G receives a significant positive update, while D receives a penalty for being fooled.
- This feedback helps G improve its generation process to create more realistic data.

Discriminator's Adaptation:

- Conversely, if D correctly identifies G's fake data (score close to 0), but G receives no reward, D is further strengthened in its discrimination abilities.
- This ongoing duel between G and D refines both networks over time.

As training progresses, G gets better at generating realistic data, making it harder for D to tell the difference. Ideally, G becomes so adept that D can't reliably distinguish real from fake data. At this point, G is considered well-trained and can be used to generate new, realistic data samples.

3.1.3Application Of Generative Adversarial Networks (GANs)

GANs, or Generative Adversarial Networks, have many uses in many different fields. Here are some of the widely recognized uses of GANs:

Image Synthesis and Generation : GANs are often used for picture synthesis and generation tasks, They may create fresh, lifelike pictures that mimic training data by learning the distribution that explains the dataset

Image-to-Image Translation : GANs may be used for problems involving image-to-image translation, where the objective is to convert an input picture from one domain to another while maintaining its key features. GANs may be used, for instance, to change pictures from day to night, transform drawings into realistic images, or change the creative style of an image.

Text-to-Image Synthesis : GANs have been used to create visuals from descriptions in text. GANs may produce pictures that translate to a description given a text input, such as a phrase or a caption. This application might have an impact on how realistic visual material is produced using text-based instructions.

Data Augmentation : GANs can augment present data and increase the robustness and generalizability of machine-learning models by creating synthetic data samples.

Data Generation for Training : GANs can enhance the resolution and quality of low-resolution images. By training on pairs of low-resolution and high-resolution images, GANs can generate high-resolution images from low-resolution inputs, enabling improved image quality in various applications such as medical imaging, satellite imaging, and video enhancement.

Advantages of GAN:

- Synthetic data generation
- High-quality results
- Unsupervised learning
- Versatility

Disadvantages of GAN

- Training Instability
- Computational Cost
- Overfitting
- Bias and Fairness
- Interpretability and Accountability

3.2.Improved GANs

Improved GANs refer to modifications of the original GAN architecture to address common issues such as **mode collapse**, **vanishing gradients**, and **instability** in training. Various GAN variants have been proposed to overcome these challenges:

1. Wasserstein GAN (WGAN):

- WGAN uses the Wasserstein distance to measure how close the generated data distribution is to the real data distribution.
- This modification improves training stability and prevents vanishing gradients.

2. Least Squares GAN (LSGAN):

- LSGAN replaces the binary cross-entropy loss with a least-squares objective, smoothing out the loss function and leading to more stable convergence.

3.Energy-based GAN (EBGAN):

- In EBGAN, the discriminator is treated as an energy function rather than a classifier, encouraging more stable and robust training.

Why Improvements Matter:

- Improves training convergence.
- Mitigates mode collapse (when the generator produces limited variations of output).
- Helps with optimization and preventing the discriminator from overpowering the generator.

3.3 Progressive GAN

- Progressive GANs are specifically designed to handle high-resolution image generation.
- The key innovation here is the progressive growing of layers during training.
- The process starts with low-resolution images, and new layers are gradually added to increase the image resolution step by step.
- By starting at a lower resolution, the GAN can focus on learning simple patterns first and then gradually refine its understanding as it deals with more complex patterns in higher resolutions.

Key Points:

- Used for generating high-resolution images (e.g., 1024x1024 or higher).
- Commonly applied in facial image generation.
- Training starts with low-res images, and higher-res layers are added progressively, leading to more stable training and better results.

3.4.Challenges in GANs

Training GANs is known to be difficult due to several key challenges:

Mode Collapse:

This happens when the generator produces very limited types of outputs (e.g., always generating similar images), ignoring the full diversity of the data distribution.

Training Instability:
GAN training requires a delicate balance between the generator and discriminator. If one becomes too strong compared to the other, training collapses.
Vanishing Gradients:
As the discriminator improves, it may output very confident predictions, causing gradients to vanish for the generator, making further training difficult.
Evaluation Metrics:
Measuring the quality of generated data is challenging. Common metrics include the Inception Score (IS) and the Fréchet Inception Distance (FID), but they are not perfect.

Why These Challenges Matter:
- Unstable training can prevent the GAN from learning effectively.
- Mode collapse results in lower diversity in generated data, which reduces the effectiveness of GANs for creative and generative tasks.

3.5.Paired Style Transfer

Paired Style Transfer refers to a technique in computer vision where an image's content is transformed by applying the style of another image, but with paired training data. In paired style transfer, the system is provided with two datasets: one containing the original images (content) and the other containing their corresponding styled versions (style).
This creates a supervised learning scenario, where each input image has a corresponding "styled" image, making it easier to train the model to learn the mapping between the two domains.

How Paired Style Transfer Works:

The process involves training a model (often using **convolutional neural networks**) to convert an image from the source domain (content image) into the target domain (styled image). Since the training data consists of paired examples, the model learns to understand how a particular style (e.g., Van Gogh painting style, black-and-white photo) transforms specific content (e.g., landscape photo, portrait).

Key Techniques:

Convolutional Neural Networks (CNNs):

The backbone of style transfer, CNNs can extract features from the input images and apply transformations that change the content image to match the style image.

Loss Functions:

These are used to ensure that the generated images resemble the desired outputs. In paired style transfer, both **content loss** (to preserve the content) and **style loss** (to apply the correct style) are calculated.

Image-to-Image Translation Networks:

Models like **Pix2Pix** are commonly used for paired style transfer. These networks take paired input-output data and learn the transformation using supervised learning.

Applications:

Artistic Style Transfer: Transforming photos into paintings by learning from pairs of photographs and corresponding paintings.

Photo Enhancement: Enhancing the quality of images by learning to map low-quality photos to high-quality ones.

Medical Imaging: Translating one type of medical image to another (e.g., CT scans to MRI images) by learning from paired medical data.

Examples:

Pix2Pix Model: One of the most popular architectures for paired style transfer, Pix2Pix uses a generator to create the styled image and a discriminator to evaluate whether the generated image is real or fake.

> **Example Task:** Converting a sketch to a photorealistic image by training the model on paired data where sketches correspond to their real images.

Supervised Learning: Since the model is trained on paired data, it learns explicit mappings. For instance, converting night-time images to day-time images can be achieved by training on a dataset where each night-time image has a paired day-time equivalent.

Advantage:

High Accuracy: The paired data provides explicit guidance to the model, ensuring that the transformation between content and style is learned effectively.

Better Control: Since the style for each image is predefined, the outcome is more predictable and accurate.

Limitations:

Data Dependency: Paired style transfer requires a large amount of paired data, which can be difficult to obtain, especially for specific styles or domains.

Supervised Training: The need for supervision limits the flexibility of the model compared to unpaired methods, which can learn mappings without explicit pairs.

3.6.Unpaired Style Transfer

Unpaired Style Transfer refers to a technique in computer vision and image processing where the goal is to transfer the style of one image to another without having explicitly paired training data. In contrast to paired style transfer, where each input image has a corresponding styled version for training, unpaired style transfer operates with two separate datasets: one containing content images and the other containing style images. There are no direct correspondences between the two datasets.

Key Concept

❖The main challenge in unpaired style transfer is learning a mapping between two domains (e.g., photographs and paintings) without having access to pairs of corresponding images.

❖This requires the system to understand the underlying distribution of both domains and find a way to convert images from one domain to the other while preserving key content features.

❖One of the most popular models for unpaired style transfer is CycleGAN, which uses cycle consistency as a core idea to achieve this goal.

How Unpaired Style Transfer Works:

The process typically involves two mappings:

Mapping A→B: This takes an image from domain A (e.g., a photo) and transforms it into domain B (e.g., a painting).

Mapping B→A: This performs the reverse transformation, converting an image from domain B back into domain A.

However, since there are no direct pairs, the model must ensure that the transformation preserves the key content characteristics of the image. To achieve this, unpaired style transfer employs **cycle consistency loss**.

Key Techniques:

Cycle Consistency: This ensures that the transformations between domains are meaningful and reversible.

Adversarial Loss: Both generators are trained adversarially, where each generator tries to fool the discriminator into believing that the generated images are real.

Content Preservation: The content of the original image is preserved while applying the stylistic features of the target domain.

Applications:

Unpaired style transfer is used in various creative and practical fields. Some popular applications include:

Artistic Style Transfer: Converting real photos into paintings without paired examples (e.g., transforming photographs into Van Gogh or Monet-like paintings).

Image Enhancement: Changing the weather or lighting conditions in images (e.g., converting daytime photos to nighttime scenes).

Domain Adaptation: Translating between different visual domains, such as converting horse images to zebra images, or turning sketches into more realistic images.

Advantage:

Flexibility
Generative Power
Adaptability

Limitations:

Quality Control
Training Complexity.
Mode CollapseNotable

3.7.Deepfakes

Deepfakes are synthetic media—typically videos or images—where someone's likeness, voice, or actions are replaced or altered using deep learning techniques. The term "deepfake" comes from combining "deep learning" and "fake," highlighting the use of artificial intelligence to create highly realistic fake content.

Deepfakes primarily involve the **manipulation of videos, images, or audio** to make it appear as though a person is saying or doing something they did not actually do. While deepfakes can be used for legitimate purposes, such as entertainment or art, they are often associated with deceptive or malicious uses, such as misinformation, fraud, or defamation.

How Deepfakes Work

Deepfakes are typically created using **Generative Adversarial Networks (GANs)** or similar neural network architectures.

These networks are trained on large datasets of images or videos of the target person, learning the nuances of their appearance, expressions, and voice.

The key steps to create a deepfake include:

Data Collection:

A large set of images, video frames, or audio clips of the person whose likeness will be faked is gathered. This training data is critical for teaching the AI model how the person looks and sounds from various angles and in different contexts.

Training a GAN or Autoencoder:

- The collected data is fed into a deep learning model, often a GAN, or an **autoencoder**. The GAN architecture consists of two networks: a **generator** and a **discriminator**.
- The generator creates fake images or videos, while the discriminator evaluates their authenticity by comparing them to real images or videos. Through this adversarial process, the generator improves its ability to create realistic fakes.

  Autoencoders are another common approach, often used for face swapping or replacement tasks. The encoder learns to compress an image into a lower-dimensional representation, and the decoder reconstructs the image from this representation. Once trained on two individuals, the decoder can swap faces in new images or videos.

Face or Voice Synthesis:

- Once trained, the model can generate synthetic media by blending or overlaying the learned representations of the target person onto another person in real-time.
- In videos, this often involves face-swapping, where the face of one person is replaced with the face of the target person, keeping the head movements and expressions synchronized.

Post-Processing:

- After generating the fake video or image, additional editing may be applied to improve the quality and smooth transitions between frames, ensuring the deepfake looks natural.
- This includes ensuring that shadows, lighting, and facial expressions match the surroundings.

Applications:

Deepfakes can be used across various domains, both for legitimate and harmful purposes. Some common applications include:

Entertainment and Art:

Deepfakes can be used to generate realistic visual effects in movies, creating younger or older versions of actors or digitally recreating deceased performers. It can also be used to animate historical figures or fictional characters for artistic purposes.

Social Media and Memes:

Deepfake technology has been used to create humorous or satirical content, where individuals' faces are swapped or altered for comedic effect. This is often seen in viral videos, where celebrities' faces are swapped into famous movie scenes or other public content.

Personalized Content:

Deepfakes can enable personalized avatars or digital assistants that look and sound like the user or another person. It has also been explored for creating personalized media, such as custom-tailored video messages featuring a celebrity's likeness.

Education and Training:

Deepfakes can be used for educational purposes, such as recreating historical figures or scenarios for teaching. For example, a historical figure's likeness could be used in a digital lecture to give students a more engaging learning experience.

3.8. Modes of Operation (GAN)

- **Modes of Operation** in the context of **Generative Adversarial Networks (GANs)** refer to the different ways GANs can be structured, modified, or extended to achieve specific tasks or improvements over the traditional Vanilla GAN model.
- Over time, several variants of GANs have been proposed to tackle limitations, improve performance, or adapt the model for particular use cases. Below is a detailed explanation of some key **GAN Variants and their Modes of Operation:**

1. Vanilla GAN (Basic GAN)

The Vanilla GAN is the original version of GANs, introduced by Ian Goodfellow in 2014. It consists of two neural networks: a generator and a discriminator, which are trained in an adversarial setting.

Generator: Learns to create realistic data from random noise.

Discriminator: Learns to distinguish between real data and fake data generated by the generator.

Mode of Operation:

The two networks are trained simultaneously. The generator tries to produce data that is indistinguishable from real data, while the discriminator tries to correctly classify real and generated data. The generator's goal is to fool the discriminator, while the discriminator's goal is to correctly identify real versus fake data.

Limitations:

Mode collapse: The generator may produce a limited variety of outputs.

Training instability: The training process is sensitive and can be difficult to balance, leading to convergence issues.

2. DCGAN (Deep Convolutional GAN)

Deep Convolutional GANs (DCGANs) are an extension of Vanilla GANs that use convolutional layers instead of fully connected layers, making them especially effective for generating images.

Mode of Operation:

DCGANs replace the fully connected layers with convolutional and transposed convolutional layers. The discriminator becomes a convolutional neural network (CNN) that classifies whether an image is real or generated. The generator uses transposed convolutional layers to create images from latent vectors (random noise).

Applications:

DCGANs are widely used for tasks such as image synthesis, super-resolution, and in-painting.

3. Conditional GAN (cGAN)

Conditional GANs extend the basic GAN framework by conditioning both the generator and discriminator on additional information (e.g., class labels or attributes). This allows control over the generated data.

Mode of Operation:

In a cGAN, the generator and discriminator both receive a condition (such as a label or a specific class) as input along with the random noise vector (for the generator) or the real/generated data (for the discriminator). The condition could be a class label, an attribute, or even another image.

Applications:

Image-to-image translation: For example, translating sketches into colored images.

Text-to-image synthesis: Generating images based on descriptive text input.

4. CycleGAN

CycleGAN is designed for unpaired image-to-image translation, where the task is to translate an image from one domain to another without having paired examples for training (e.g., converting photographs to paintings).

Mode of Operation:

CycleGAN introduces cycle consistency loss. This means that if you translate an image from domain A to domain B and then back to domain A, the resulting image should be similar to the original image. The cycle consistency ensures that important content is preserved during translation.

Applications:

Style transfer: Converting images between different artistic styles.

Cross-domain translation: Converting between different domains like photos ↔ paintings, horses ↔ zebras.

5. Progressive Growing GAN (ProGAN)

Progressive Growing GANs (ProGANs) incrementally grow the generator and discriminator architectures during training, starting with low-resolution images and progressively increasing the resolution.

Mode of Operation:

Training starts with small, low-resolution images, allowing the model to learn a rough approximation. As training progresses, new layers are added to both the generator and discriminator, allowing the model to gradually handle higher-resolution images. This reduces training instability and improves the quality of high-resolution outputs.

Applications:

Image generation: Used to create photorealistic high-resolution images.

Art creation: Used in generative art, including the creation of high-quality synthetic faces.

3.9.Key Feature Set

Generative Adversarial Networks have revolutionized the field of machine learning and AI, especially in tasks like image generation, data augmentation, and deepfakes. Here are some of the **key features** of GANs:

Adversarial Training: GANs involve two networks—a generator and a discriminator—that are trained in opposition to one another. This adversarial setup is the core idea behind GANs and leads to the generator improving as it tries to fool the discriminator, and vice versa.

Generative Modeling: GANs learn to model the distribution of data and generate new samples that resemble the training data. This ability to generate new data makes GANs unique compared to traditional machine learning models that primarily focus on classification or regression tasks.

Non-supervised Learning: GANs typically do not require labeled data (although some variants like conditional GANs may use labels). This makes them highly valuable in scenarios where labeled data is scarce or expensive to obtain.

High-Quality Synthetic Data: GANs can produce high-quality synthetic data, whether it be images, videos, or even text. For example, GANs are used to generate realistic human faces, artwork, and even music.

Flexibility: GANs are highly flexible and have been adapted to many different tasks. Variants like CycleGAN, Pix2Pix, and StyleGAN offer specialized solutions for tasks like image translation, style transfer, and high-resolution image synthesis.

Application Versatility: GANs have broad applications in various fields, such as synthetic data generation, data augmentation for machine learning models, image super-resolution, and even medical imaging.

3.10.High-Level Flow of GANs

The **high-level flow** of a GAN can be broken down into the interaction between the **generator** and **discriminator** in an adversarial setting. Here's a step-by-step breakdown:

Noise Input: The process starts with the generator receiving random noise as input. This noise is usually sampled from a latent space (often a uniform or normal distribution). The generator's task is to transform this random noise into a data sample that resembles real data.

Data Generation: The generator processes the noise input through its network layers to produce synthetic data. For instance, in image generation tasks, this could be a synthetic image.

Discriminator's Role: The discriminator, which is a binary classifier, receives both the real data from the training dataset and the synthetic data from the generator. It aims to classify whether each data point is real or generated (fake).

Feedback to Generator: The generator and discriminator are connected in a loop through the loss function. If the discriminator successfully identifies the synthetic data as fake, it provides a gradient update to the generator, effectively teaching it how to produce more realistic samples. If the discriminator fails to correctly classify the data as fake, it updates its own parameters to improve its ability to distinguish between real and fake data.

Adversarial Learning: This adversarial process continues until the generator becomes skilled enough to produce data that the discriminator cannot reliably distinguish from real data. In theory, training reaches a point where the discriminator is no better than random guessing, and the generator produces highly realistic outputs.

Convergence: The GAN training process is considered successful when the generator produces samples that are indistinguishable from the real data, as determined by the discriminator.

2.11. Replacement (GAN in Face Replacement or Object Replacement)

- Replacement refers to the use of GANs to substitute one element in an image or video with another, particularly for tasks like face swapping, object replacement, or deepfakes.
- The idea is to replace a specific part of an image (or multiple frames in a video) while maintaining other elements constant, such as the background, lighting, or context.

How GANs are used in Replacement:

Face Replacement: GANs can be trained to swap faces in images or videos. The generator network learns to synthesize a realistic face replacement by blending the new face onto the body and adjusting details like lighting and pose. Popular GAN architectures for this purpose include **StyleGAN** and **DeepFaceLab GAN**.

- **Training:** GANs for face replacement are typically trained on large datasets of the two faces to be swapped. They learn to map the facial features of one individual onto the face of another while maintaining expressions and orientations.
- **Real-Time Processing:** Recent advancements allow for real-time face replacement in videos. By using GANs, the face swap can be applied frame-by-frame, ensuring consistent and realistic transitions throughout the video.

Object Replacement: Beyond faces, GANs can be used for replacing or modifying objects in images. For example, replacing one car with another in a street scene or changing the appearance of an item in a product photo.

Challenges: Ensuring realism in replacements can be challenging, especially with varying lighting conditions, occlusions, or intricate textures. GANs must handle these variations to produce convincing replacements.

2.12. Re-enactment (GANs for Face and Body Re-enactment)

Re-enactment involves using GANs to manipulate or recreate specific movements, expressions, or gestures in an image or video, typically involving human faces or bodies. The goal is to map the movements or expressions of a source individual onto a target individual in a highly realistic manner.

How GANs are used in Re-enactment:

Face Re-enactment: GAN-based face re-enactment takes facial expressions, mouth movements, or eye gestures from a source video and transfers them onto the face of a target individual. For example, by analyzing a person's facial movements in a video, GANs can generate a new video where a different person exhibits the same expressions and lip movements, mimicking speech or reactions.

- **Training Process:** GANs are trained on datasets consisting of videos or images of the target individual. During training, the generator learns to re-enact the target's facial expressions based on the source input. The discriminator, in turn, learns to distinguish between real re-enactments and synthetic ones, improving the generator’s performance over time.
- **Application in Deepfakes:** In the context of deepfakes, re-enactment is often used to create fake videos of a person saying or doing things they did not actually say or do. For instance, re-enactment techniques can simulate a famous person speaking with the lip movements corresponding to any given audio track.

Body Re-enactment: Similar to face re-enactment, GANs can also be used for full-body re-enactment, where a person's body movements and gestures are replicated onto another person. This is especially useful in virtual reality, gaming, or motion capture for movies.

Challenges: Realistic re-enactment requires detailed modeling of how the face or body behaves in different situations. This involves capturing minute changes in expressions, lighting, and camera angles. GANs must also handle potential occlusions, such as hair or glasses covering parts of the face.

Applications:

1. **Entertainment:** Re-enactment GANs are used in films and media to recreate actors' performances or animate historical figures.
2. **Virtual Reality and Gaming:** Motion capture and re-enactment GANs can make virtual avatars behave more realistically by mapping real human expressions and movements onto digital characters.
3. **Human-Computer Interaction:** GANs can be used to develop more interactive and lifelike digital assistants by replicating human expressions and behaviors.

ASSIGNMENT – UNIT III

1. What improvements have been introduced in GAN architectures to tackle the issues of instability and mode collapse? (k3)

2. How does the layer-wise progressive growth of Progressive GANs contribute to generating high-quality, high-resolution images? What training challenges does this approach address effectively?(k4)

3. What are the common challenges in training GANs, including mode collapse and vanishing gradients? Discuss techniques used in recent research to overcome these difficulties.(k5)

4. What is the difference between Paired Style Transfer (like Pix2Pix) and Unpaired Style Transfer (like CycleGAN)? (k4)

5. How are GANs utilized to generate deepfakes, specifically in replacement and re-enactment modes? (k5)

PART A- UNIT-III

1. What is the basic architecture of a Vanilla GAN?

A Vanilla GAN consists of two neural networks: a generator, which creates fake data, and a discriminator, which classifies data as real or fake. They are trained together in a minimax game until the generator produces realistic data that the discriminator cannot distinguish from real data.

2. How does Wasserstein GAN (WGAN) differ from Vanilla GAN?

WGAN uses the Wasserstein distance (earth mover's distance) as a loss function instead of binary cross-entropy. This helps stabilize training and mitigates problems like mode collapse by providing smoother gradients for the generator.

3. What is mode collapse in GANs?

Mode collapse occurs when the generator produces limited variations of the output, ignoring the diversity in the real data distribution. This results in a lack of variety in the generated samples.

4. How does Least Squares GAN (LSGAN) improve over Vanilla GAN?

LSGAN replaces the binary cross-entropy loss with a least-squares loss, which penalizes the difference between real and generated data more smoothly, improving stability and helping to avoid vanishing gradients.

5. What is Progressive GAN, and how does it work?

Progressive GAN generates high-resolution images by gradually increasing the resolution of the generator and discriminator layers during training. This progressive approach stabilizes training and improves the quality of the generated images.

.

PART A- UNIT-III

6. What are the benefits of progressively growing GANs?

Progressive GANs reduce training instability and allow for the generation of high-resolution images, addressing issues of poor quality often seen in standard GANs when trained directly at high resolutions.

7. What is Paired Style Transfer?

Paired Style Transfer is used when there are corresponding pairs of images in two domains (e.g., an image and its sketch). Models like Pix2Pix are trained to learn direct mappings from one domain to another.

8. What is Unpaired Style Transfer?

Unpaired Style Transfer works when corresponding pairs of images from two domains are not available. CycleGAN is a model that uses cycle consistency loss to learn mappings between unpaired data, such as photos from two different styles or environments.

9. What is the main challenge addressed by CycleGAN in style transfer?

CycleGAN addresses the lack of paired training data by using cycle consistency loss, ensuring that when an image is transferred from one domain to another and back again, it remains consistent with the original image.

10. How do GANs contribute to the creation of deepfakes?

GANs are used to generate deepfakes by training the generator to create realistic facial or object images. These fake images can then be swapped (replacement) or manipulated (re-enactment) to produce highly realistic videos or photos.

11. What are the ethical concerns surrounding deepfakes?

Deepfakes can be used for misinformation, fraud, and defamation, posing significant ethical challenges. They raise concerns about privacy, consent, and the potential for malicious use in creating misleading media content.

PART A- UNIT-III

12. What is the difference between replacement and re-enactment in deepfakes?

Replacement involves substituting one face with another in an image or video, while re-enactment involves manipulating a person's facial expressions or movements in a video to create new, artificial actions.

13. What role does the discriminator play in GANs?

The discriminator evaluates the data it receives, classifying it as real or fake. Its feedback helps the generator improve by penalizing it when the generated data does not closely resemble the real data.

14. What is the generator's objective in a GAN?

The generator’s objective is to create data that is so realistic that the discriminator cannot distinguish it from the actual data, effectively "fooling" the discriminator.

15. How does WGAN help mitigate vanishing gradients in GAN training?

WGAN uses a loss function based on Wasserstein distance, which provides a more stable training signal and prevents vanishing gradients, thus allowing the generator to learn effectively even when the discriminator becomes very strong.

16. What is the purpose of cycle consistency loss in CycleGAN?

Cycle consistency loss ensures that when an image is transferred from one domain to another and back again, the output is the same as the original input. This enforces consistency and prevents the generator from creating arbitrary mappings.

17.How does Progressive GAN handle the challenge of training at high resolutions?

Progressive GANs start training at a low resolution and gradually add higher-resolution layers to both the generator and discriminator. This progressive growth helps avoid the instability

and poor performance often encountered when training directly at high resolutions.

18. What are some common applications of Paired Style Transfer?

Paired Style Transfer is commonly used in applications such as image-to-image translation, for example, converting sketches to photos, black-and-white images to color, or day-to-night transformations in photography.

19. How are GANs used in style transfer tasks?

GANs are used in style transfer to learn the mapping between two domains, such as transferring artistic styles to photographs or converting images from one domain (e.g., summer) to another (e.g., winter).

20. What are the main modes of operation in deepfakes?

The two primary modes of operation in deepfakes are replacement, where one person's face is swapped with another's, and re-enactment, where a person's expressions or movements are modified to produce new behaviors in a video.

PART B- UNIT III

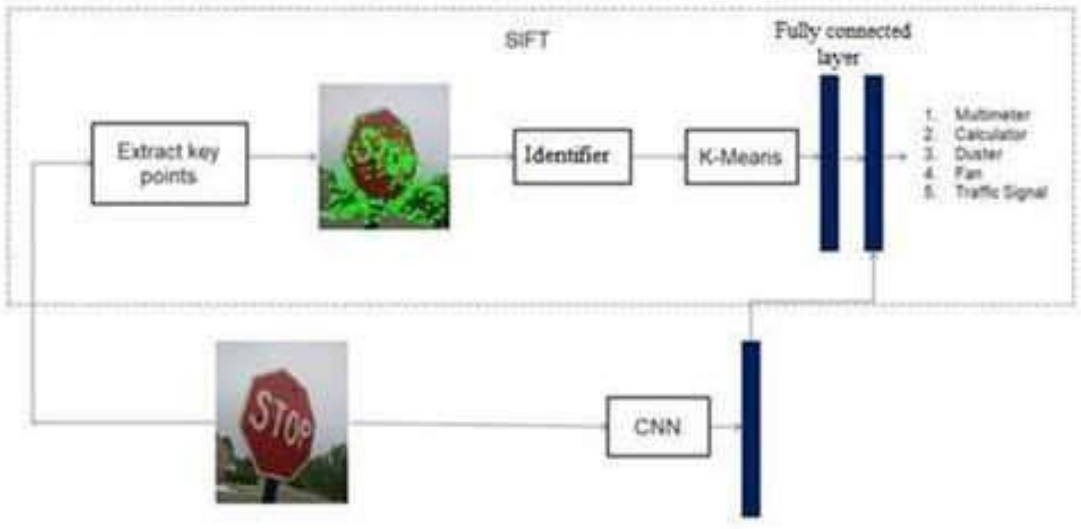
1. How does the progressive growth of layers in Progressive GANs enhance training stability and image quality? (k2)
2. Explain the use of Wasserstein distance in WGAN. How does it improve training stability and address mode collapse compared to the original GAN loss function?(k3)
3. How do Least Squares GAN (LSGAN) and Wasserstein GAN (WGAN) differ in their approach to stabilizing GAN training? What are the effects of these improvements on the diversity and quality of generated samples?(k3)
4. What are the common challenges in GAN training, such as mode collapse and vanishing gradients? How do techniques like WGAN and LSGAN attempt to solve these issues?(k5)
5. How does CycleGAN perform Unpaired Style Transfer using cycle consistency loss? What challenges and limitations does this approach encounter in handling unpaired data?(k5)
6. How do Progressive GANs enable high-resolution image generation while maintaining stability? Discuss how the progressive addition of layers contributes to the final output's detail and quality.(k2)

SUPPORTIVE ONLINE COURSES – UNIT III

- https://www.coursera.org/specializations/deep-learning
- https://www.coursera.org/specializations/generative-adversarial-networks-gans
- https://www.datacamp.com/

1.Object detection:

CNN has been applied to object recognition across images by classifying objects based on shapes and patterns found within an image. CNN models have been created that can detect a wide range of objects from everyday items such as food, celebrities, or animals to more unusual ones including dollar bills and guns. Object detection is performed using techniques such as semantic or instance segmentation. CNNs have been used to localize and identify objects within images as well as create different views of those objects such as for use in drones or self-driving cars.

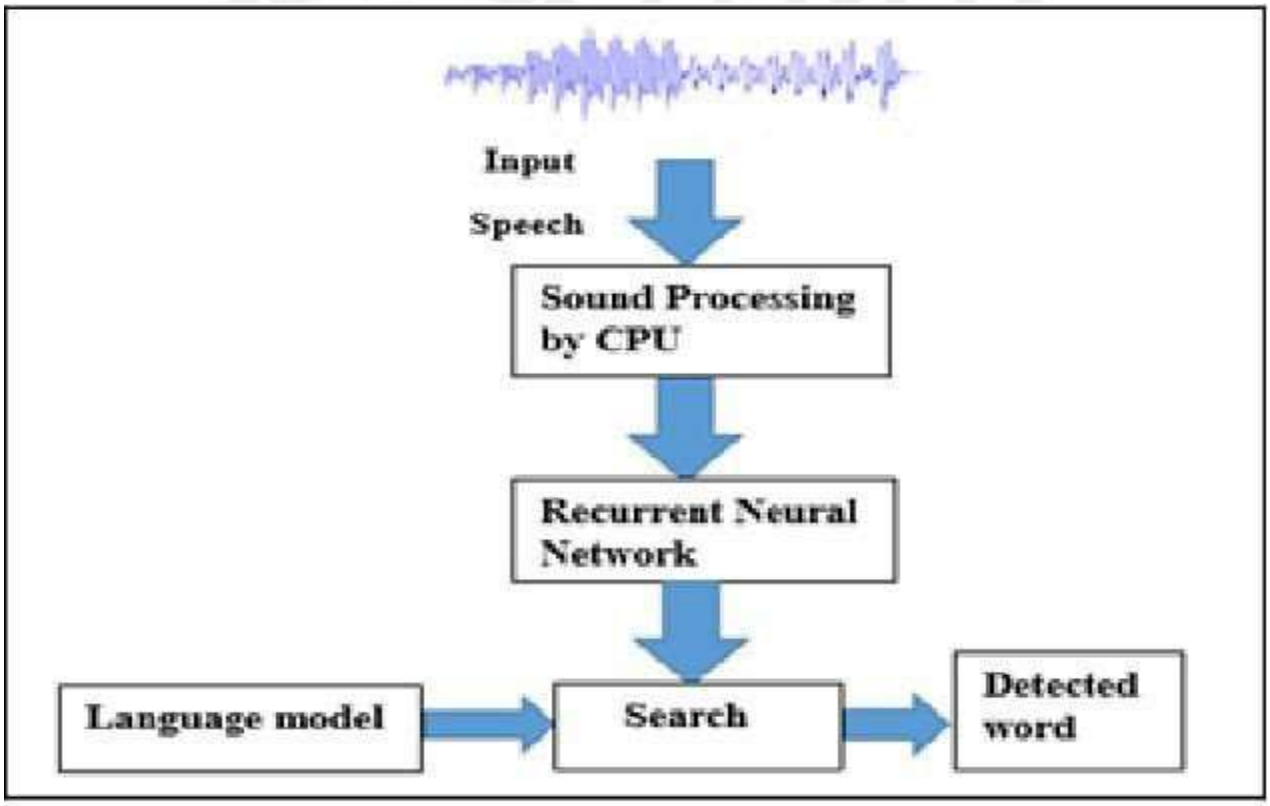


2.Speech Recognition

RNNs can be used for predicting phonetic segments considering sound waves from a medium as an input source .The set of inputs consists of phoneme or acoustic signals from an audio which are processed in a proper manner and taken as inputs. The RNN network will compute the phonemes and then produce a phonetic segment along with the likelihood of output.The steps used in speech recognition are as follows:-

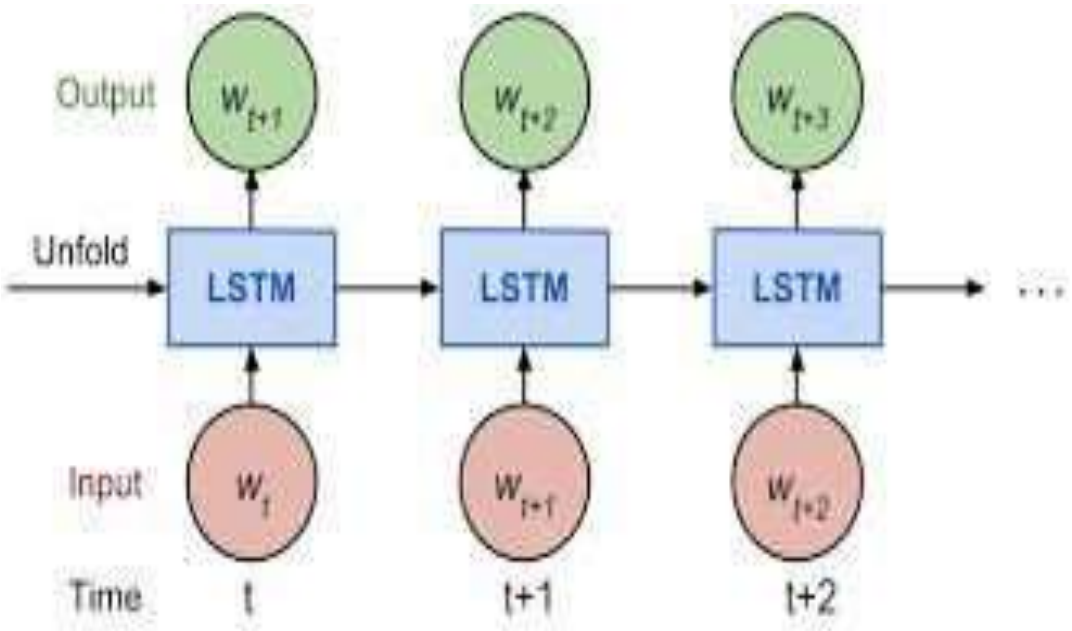
The input data is first processed and recognized through a neural network. The result consists of a varied collection of input sound waves.

The information contained in the sound wave is further classified by intent and through keuwords related to the query.



3.Language Modelling and Generating Text

Taking a sequence of words as input, we try to predict the possibility of the next word. This can be considered to be one of the most useful approaches for translation since the most likely sentence would be the one that is correct.In this method, the probability of the output of a particular time-step is used to sample the words in the next iteration.



PRESCRIBED TEXT BOOKS AND REFERENCE BOOKS

TEXT BOOKS:

1. "GANs in Action: Deep Learning with Generative Adversarial Networks" by Jakub Langr and Vladimir Bok
2. "Deep Learning with Python" by François Chollet
3. "Deep Learning with TensorFlow and Keras" by Amita Kapoor

REFERENCES:

1.Goodfellow, Ian, et al. "Generative Adversarial Nets", 2014.

2.Karras, Tero, et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation", 2018.

3.Isola, Phillip, et al. "Image-to-Image Translation with Conditional Adversarial Networks", 2017.

MINI PROJECT SUGGESTIONS

- ✓ **TEAM 1:Build a simple Vanilla GAN from scratch to generate images (e.g., handwritten digits using the MNIST dataset).**
- ✓ **TEAM 2: Progressive Growing of GANs for High-Resolution Image Generation**
- ✓ **TEAM 3: Paired and Unpaired Image-to-Image Translation using GANs (Style Transfer)**
- ✓ **TEAM 4: Deepfake Creation and Detection**
- ✓ **TEAM 5: GANs for Image Re-enactment and Replacement**

THANK YOU

Disclaimer: