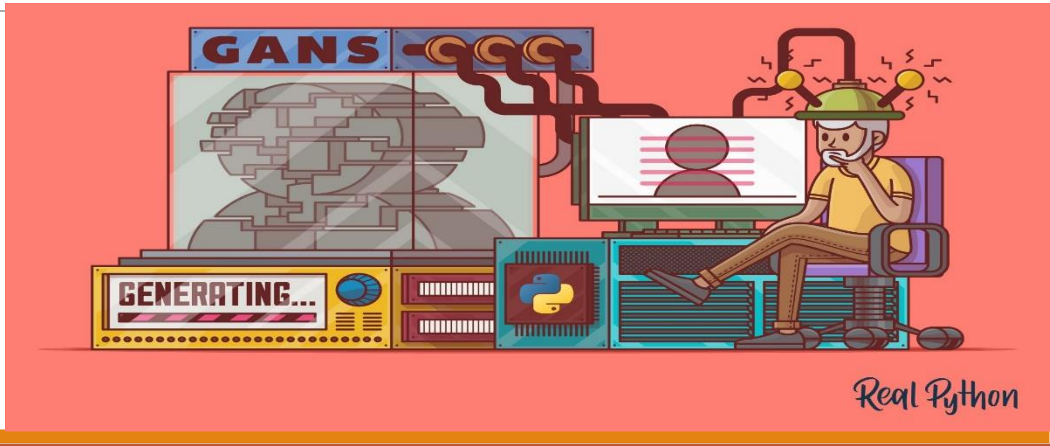
Generative Adversial Networks

SINI RAJ PULARI

BUYOFUEL –CONSULTANT

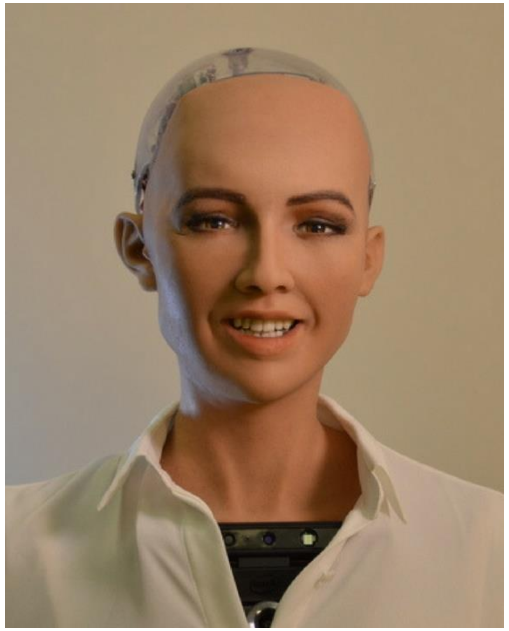
TUTOR – TEACHING AND RESEARCH

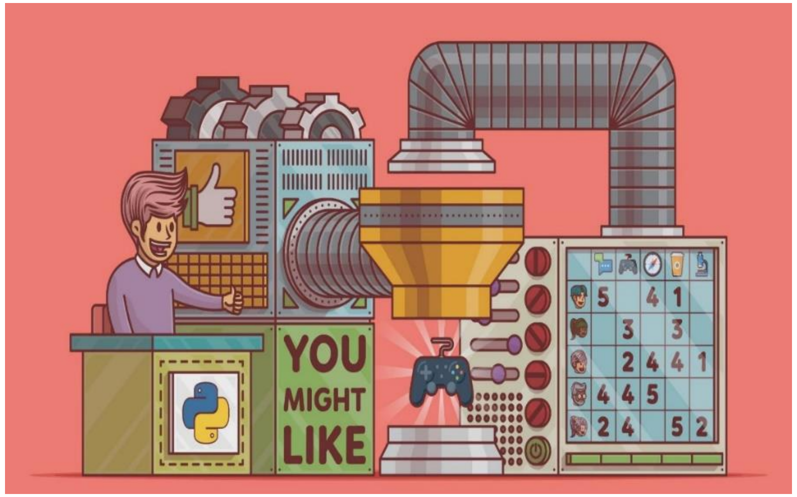
GOVT. UNIVERSITY IN THE KINGDOM OFBAHRAIN



Artificial Intelligence

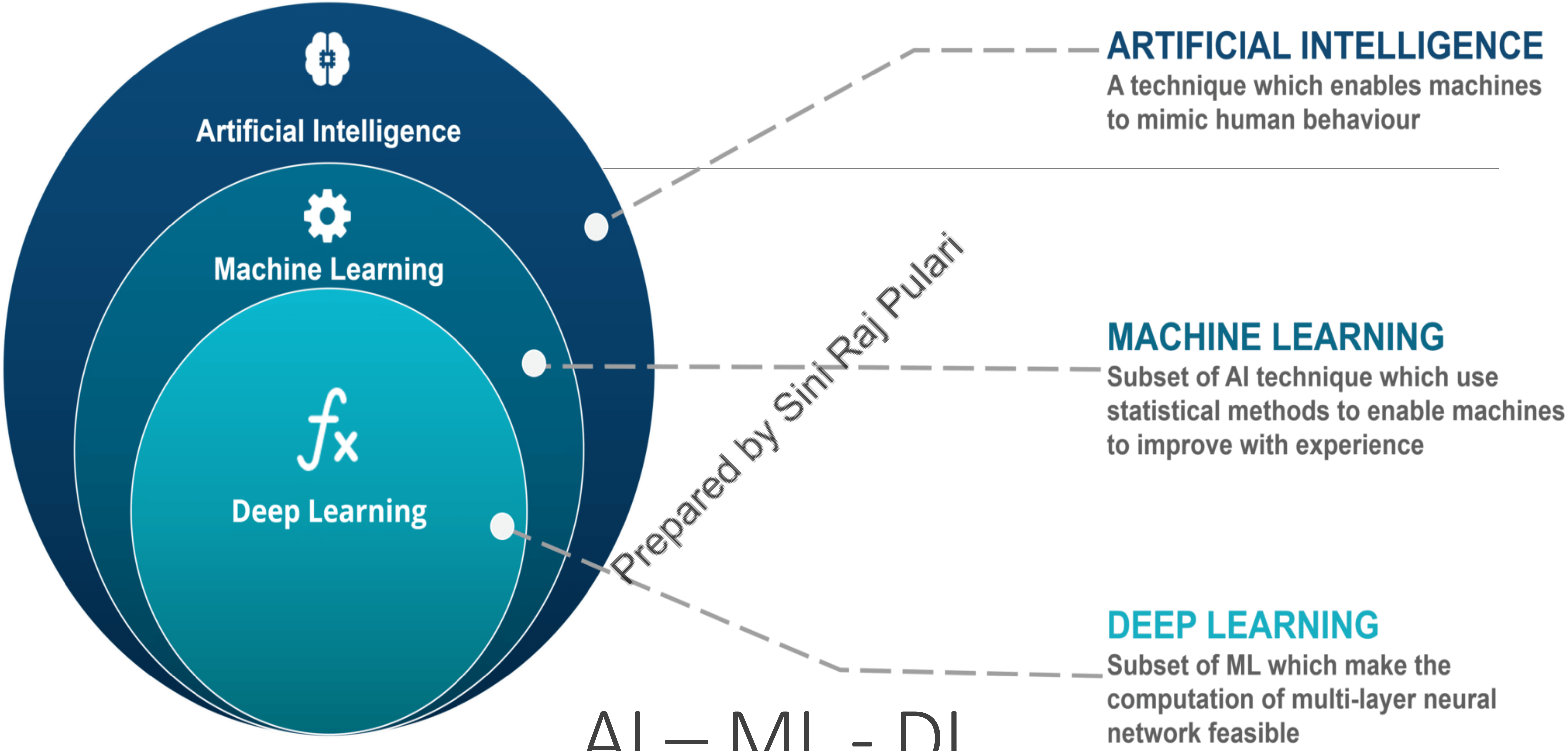
AI refers to the simulation of human intelligence processes by machines, including learning, reasoning and self-correction.



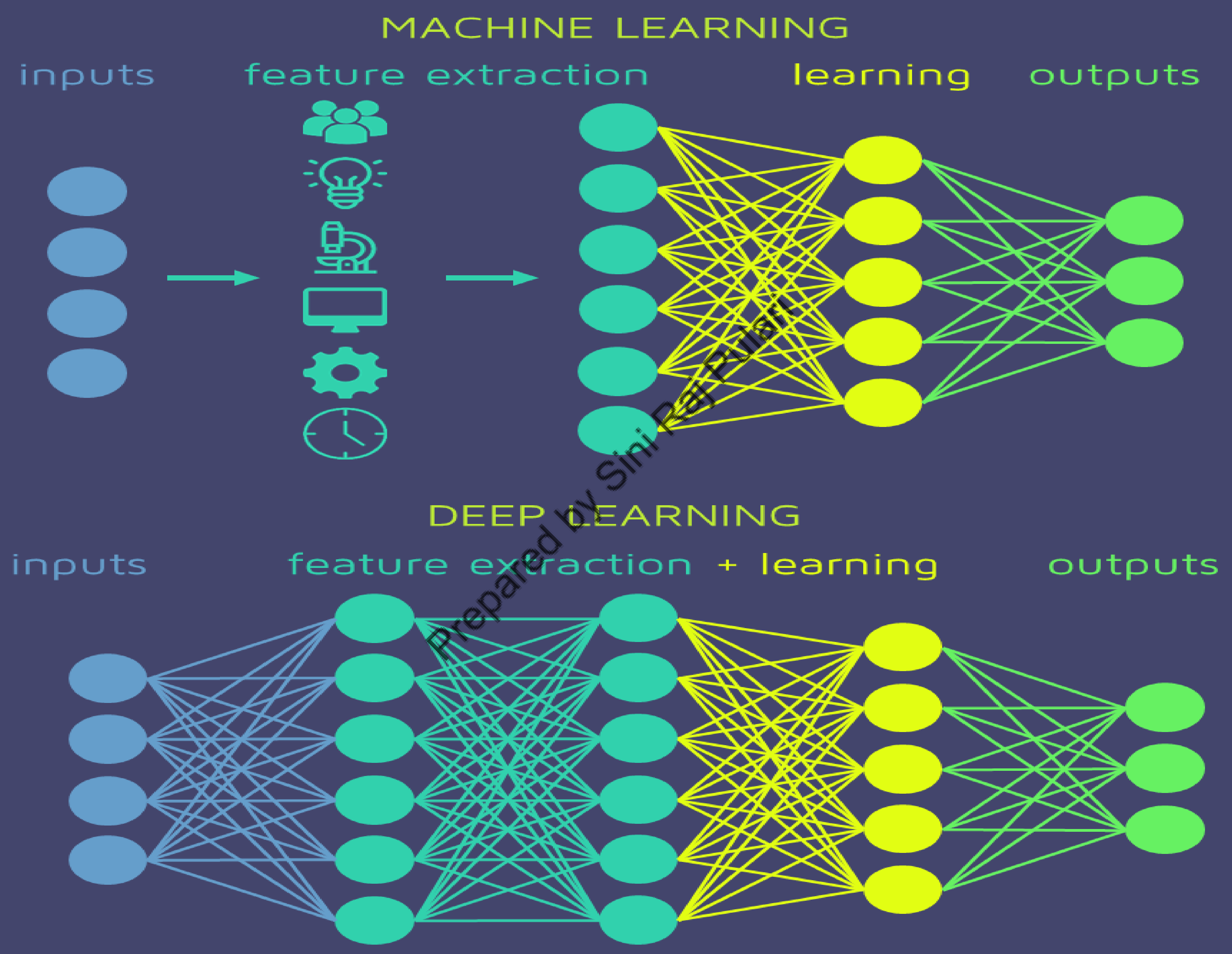


ML enables machines to learn by themselves through the use of given information. Feature extraction is manually done

Recommender systems


Artificial Intelligence
Machine Learning
fx
Deep Learning
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour
MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

AI – ML - DL


MACHINE LEARNING
inputs
feature extraction
learning
outputs
DEEP LEARNING
inputs
feature extraction + learning
outputs

DL Algorithms

Major DL algorithms

CONVOLUTIONAL NEURAL NETWORKS (CNN)

RECURRENT NEURAL NETWORKS(RNN)

AUTOENCODERS (AE)

GENERATIVE ADVERSIAL NETWORKS(GAN)

Transfer Learning

Generative Adversarial Networks

GANS


Which face is fake?
A
B
C

Check out this website !

https://www.thispersondoesnotexist.com/

IAN GOODFELLOW

Before we move on..

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn function to map
$$x \rightarrow y$$

Examples: Classification, regression, object detection, semantic segmentation, etc.

Unsupervised Learning

Data: x
x is data, no labels!

Goal: Learn the *hidden* or *underlying structure* of the data

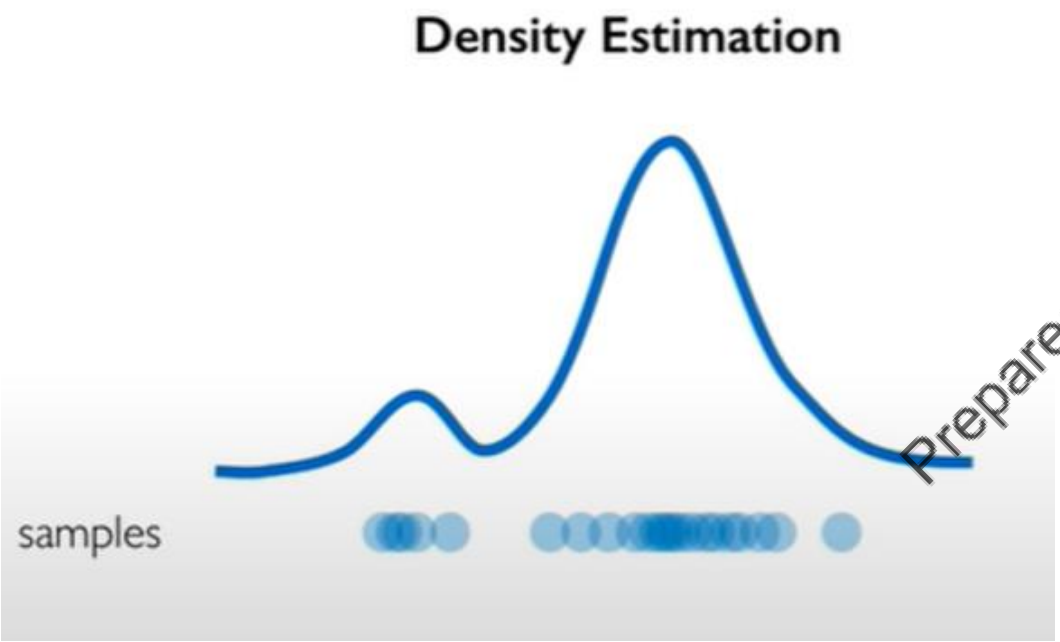
Examples: Clustering, feature or dimensionality reduction, etc.

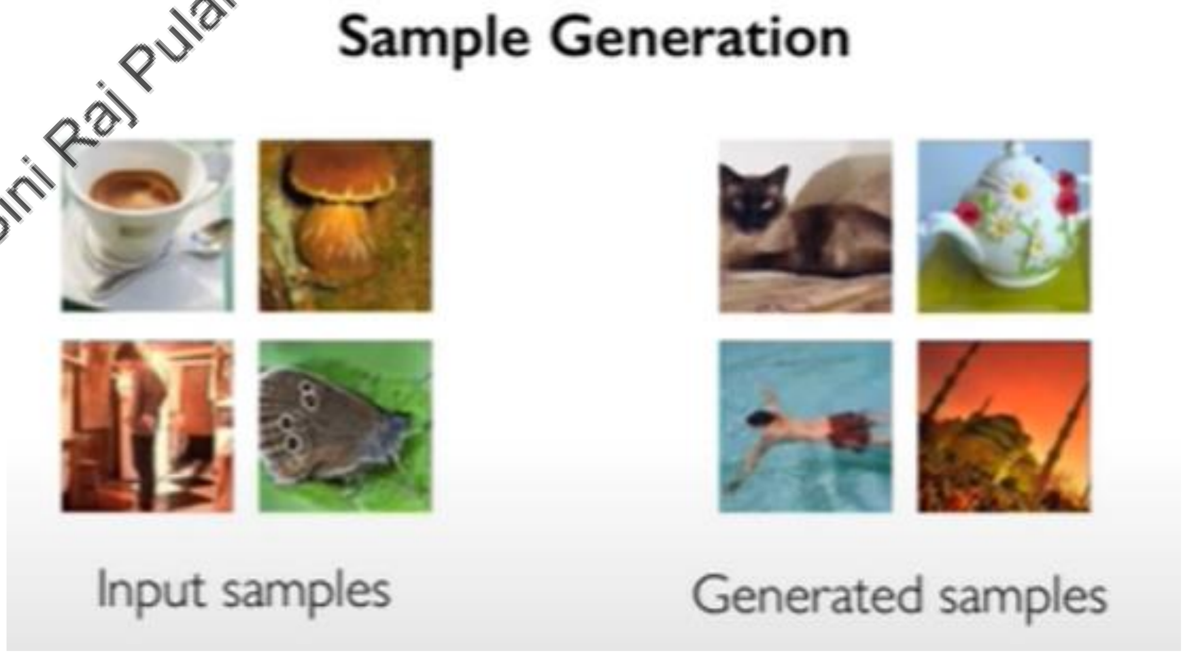
- Helps to identify the foundational level of explanatory factors of underlying data (attributes and distribution of data)
- Helps to generate brand new data that resemble the Data distribution

Generative Models

- A Generative Model is a powerful way of learning any kind of **data distribution** using **unsupervised learning**
- All types of generative models **aim at learning the true data distribution** of the training set so as to **generate new data points with some variations**

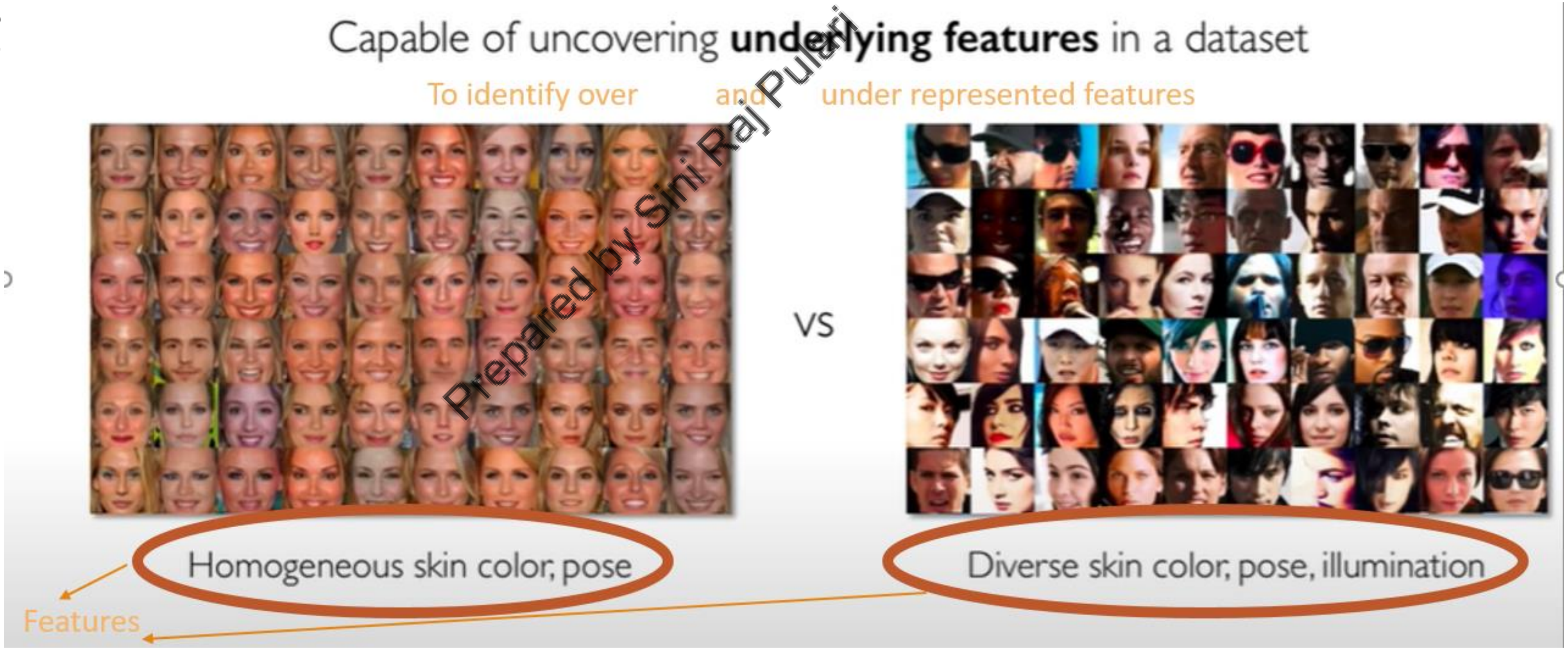
Why Use GANs?


Density Estimation
samples

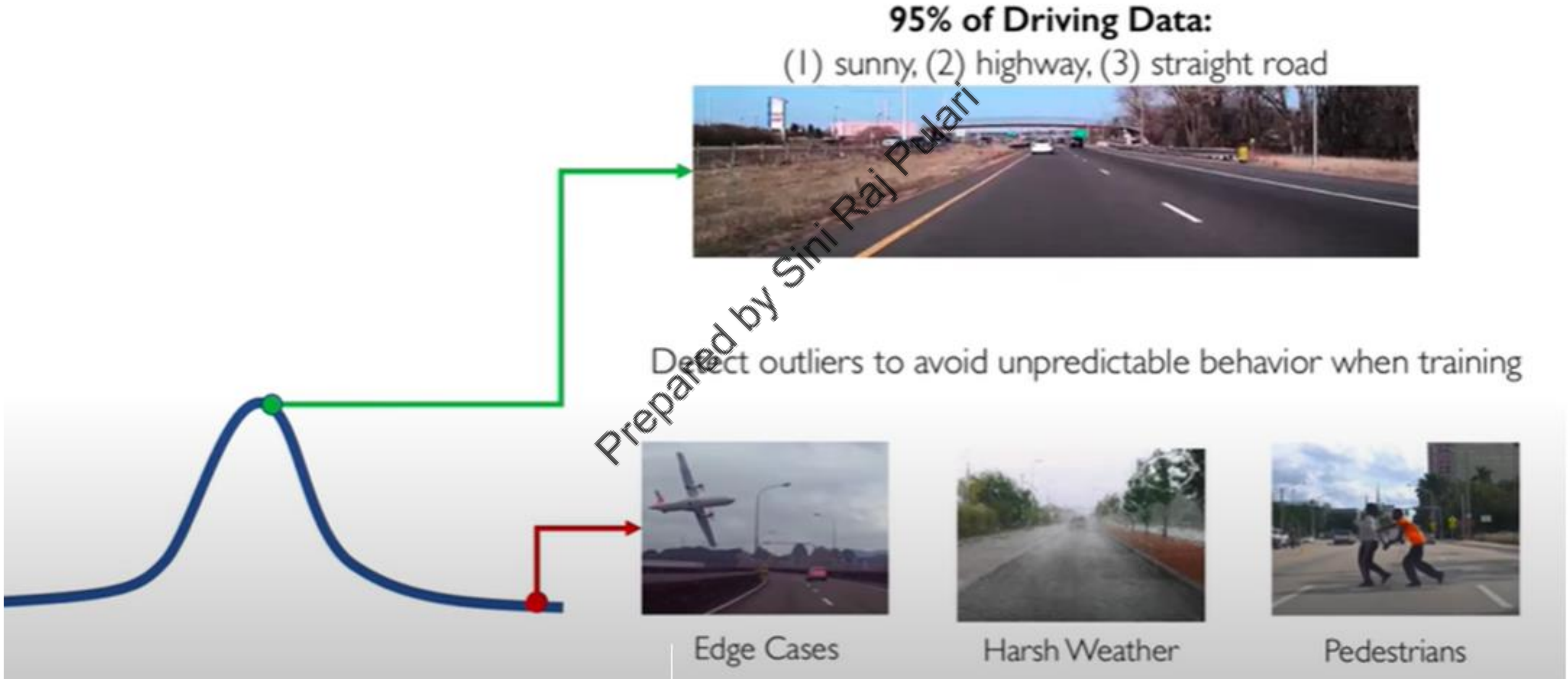

Sample Generation
Input samples
Generated samples

Why Use GANs?

Debasing



Why generative models? Outlier detection

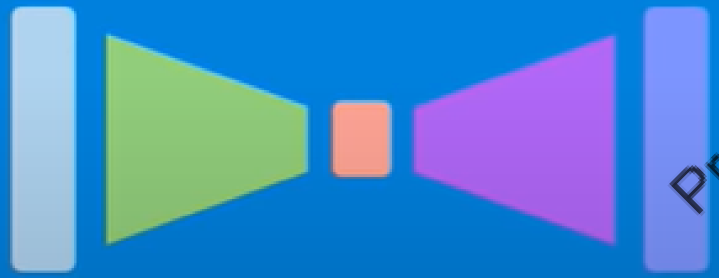


Generative Models – Why?

- **DENSITY ESTIMATION**
- **SAMPLE GENERATION**
- **DEBAISING**
- **OUTLIER DETECTION**

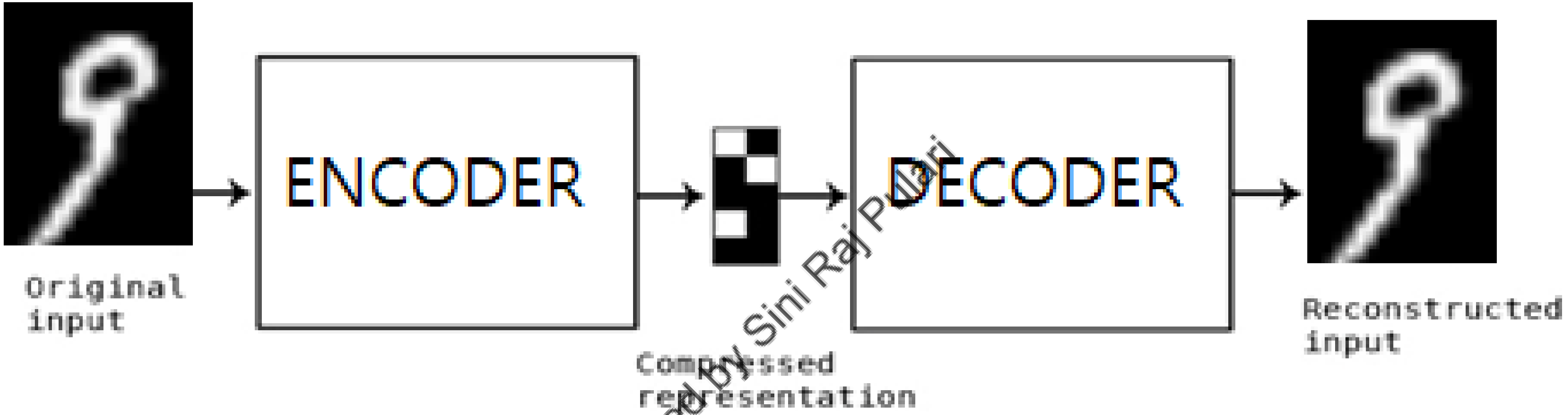
Latent variable models

Autoencoders and Variational Autoencoders (VAEs)



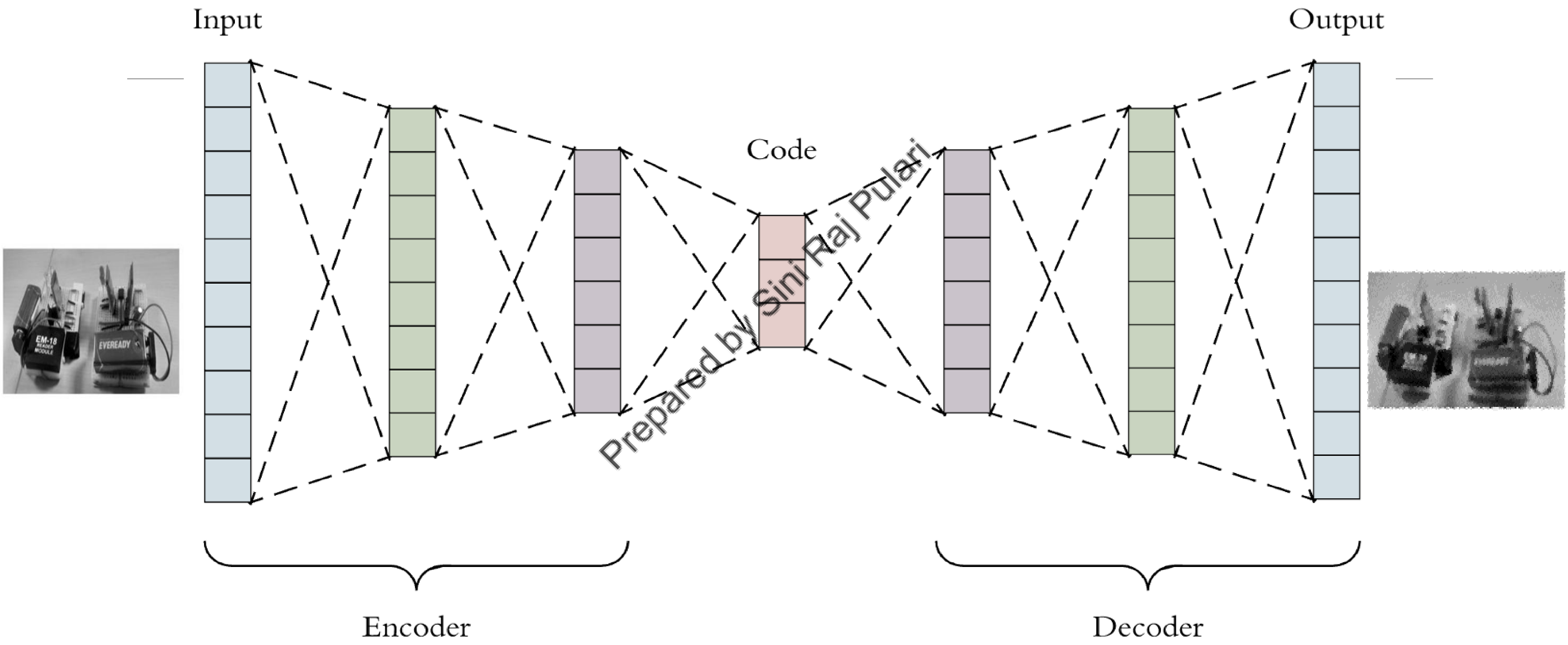
Generative Adversarial Networks (GANs)

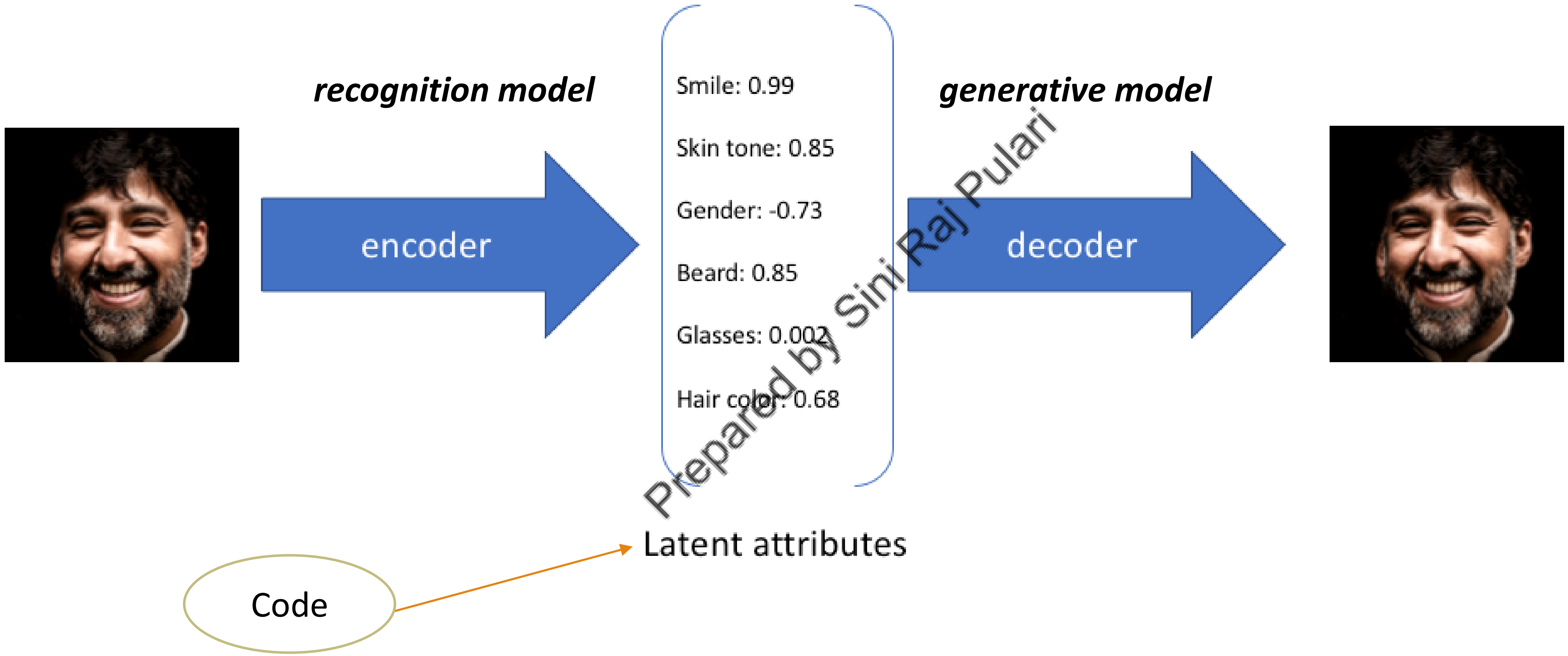
Auto Encoder…



- An auto encoder is a simple ML algorithm which acquires input image and it will reconstruct the same. I.e. the image is compressed - **dimensionality reduction**.
- Help to **reduce the unnecessary features** in datasets
- **Dimensionality reduction certainly helps in identifying the features which are useful** (i.e. of interest).
- Auto encoders are meant for this Dimensionality Reduction.

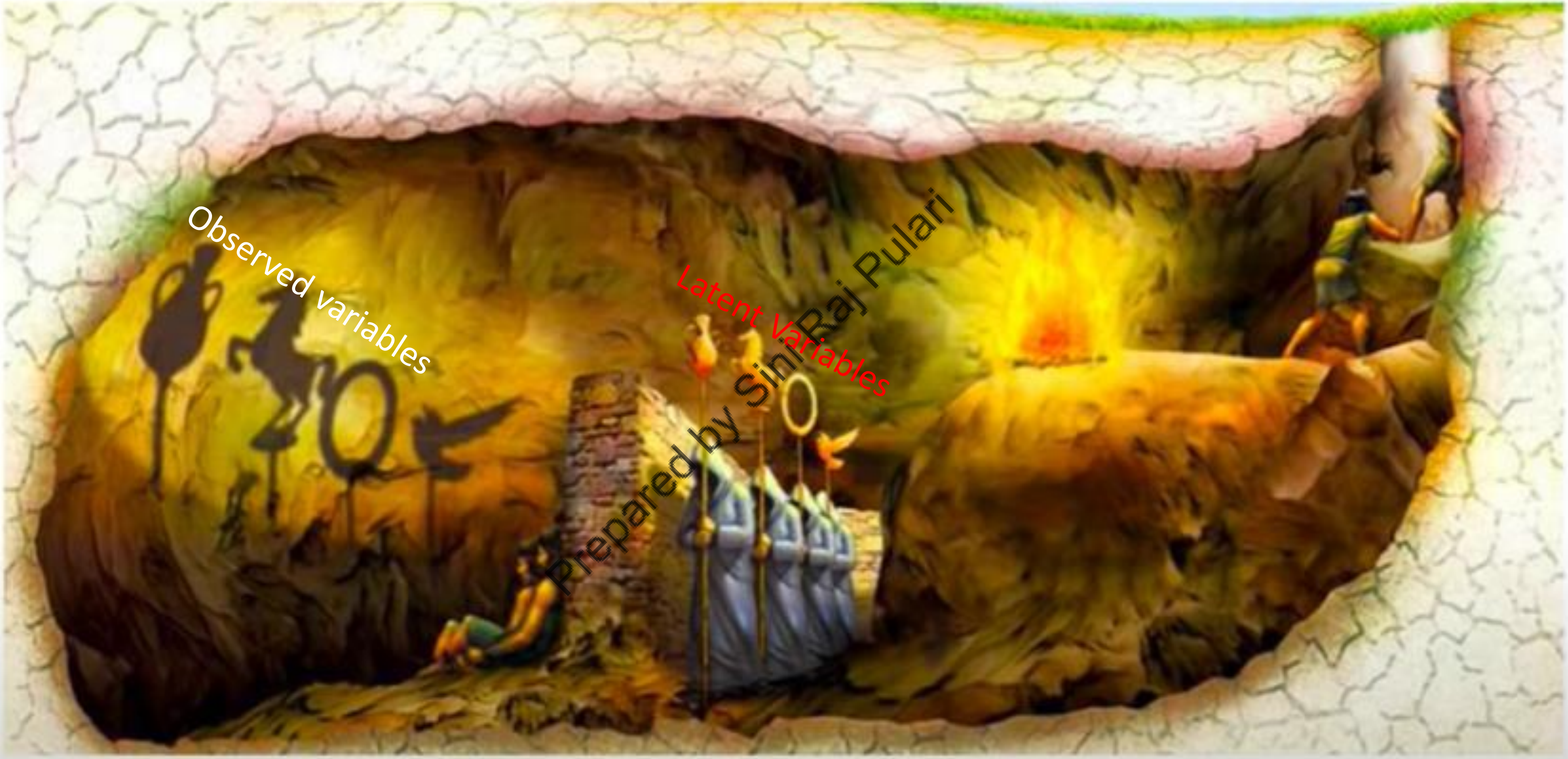
A better pic


Input
Output
Code
EM-18
READER MODULE
EVEREADY
Encoder
Decoder



recognition model
encoder
Smile: 0.99
Skin tone: 0.85
Gender: -0.73
Beard: 0.85
Glasses: 0.002
Hair color: 0.68
generative model
decoder
Latent attributes
Code

What is a latent variable?

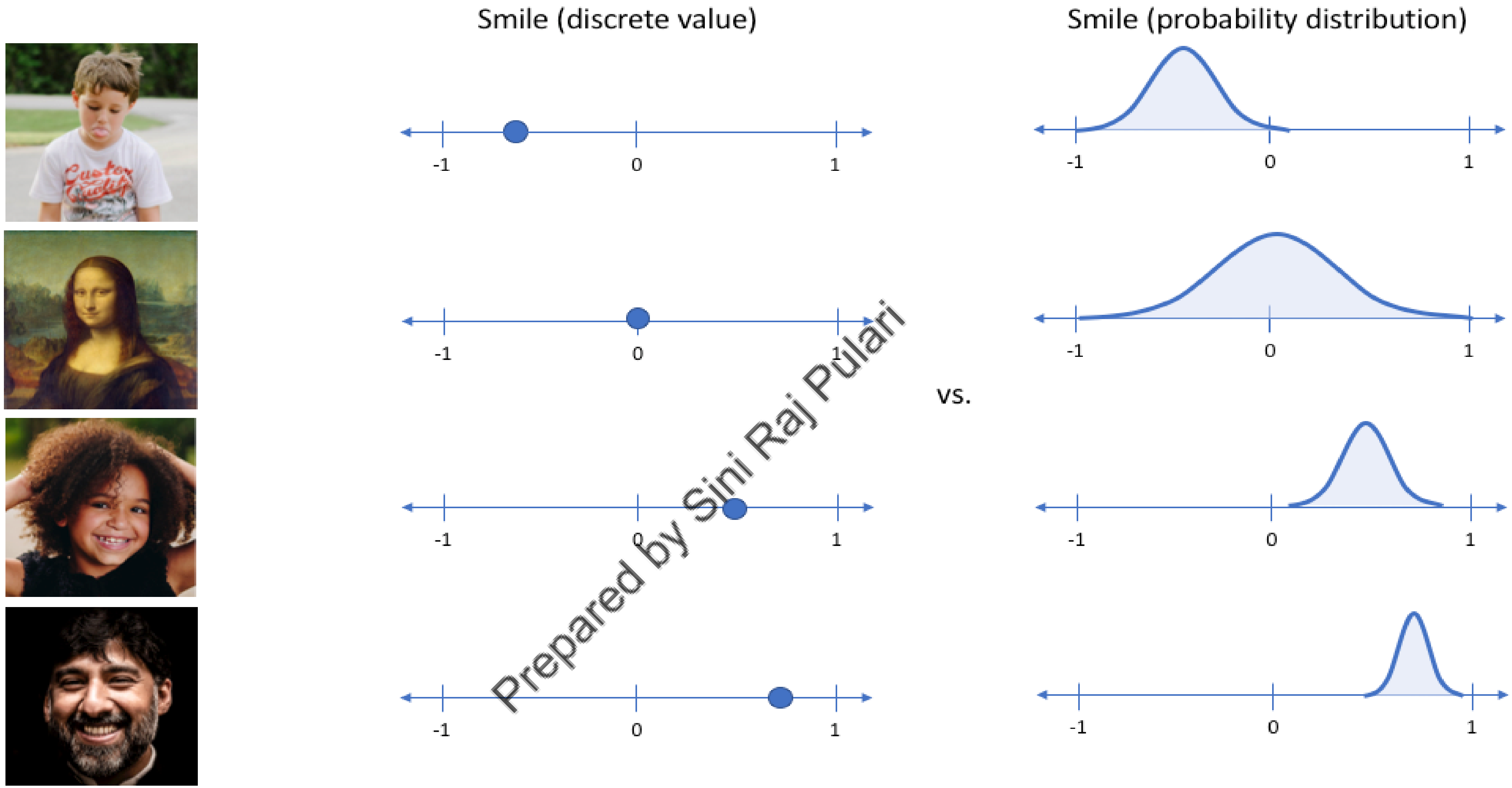


Myth of the Cave

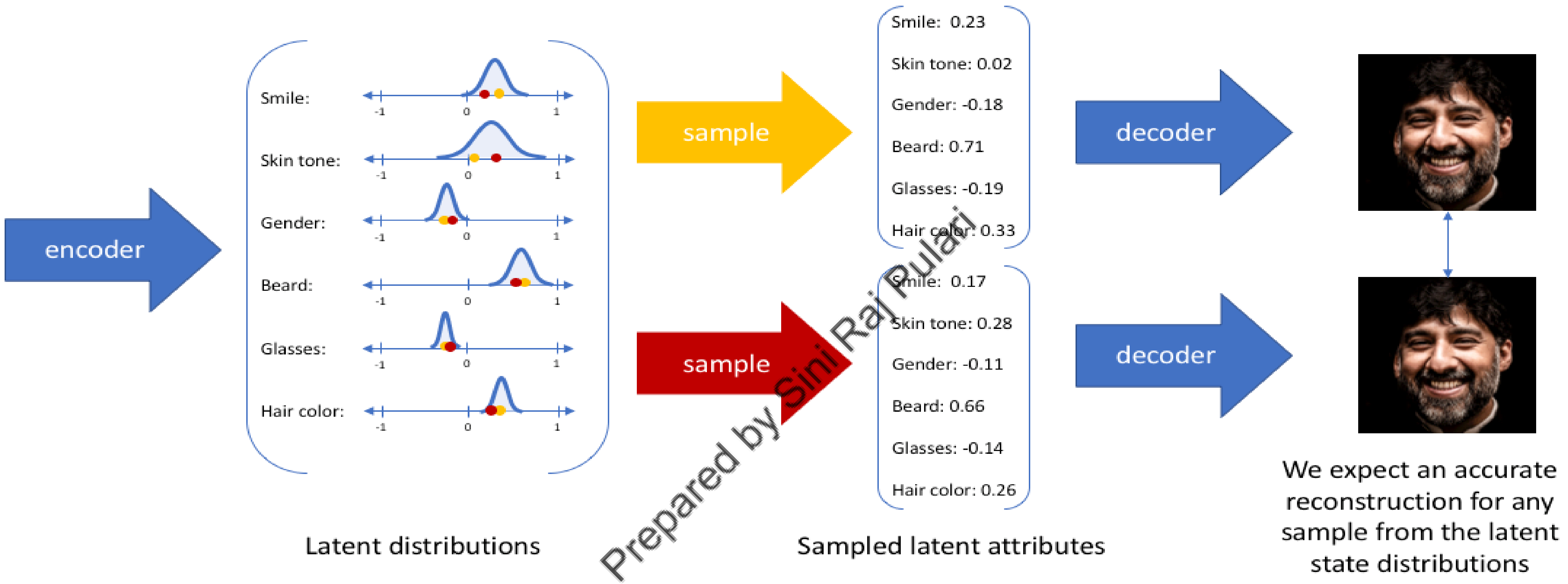
What is the difference between latent and observed variables?

Latent Variables

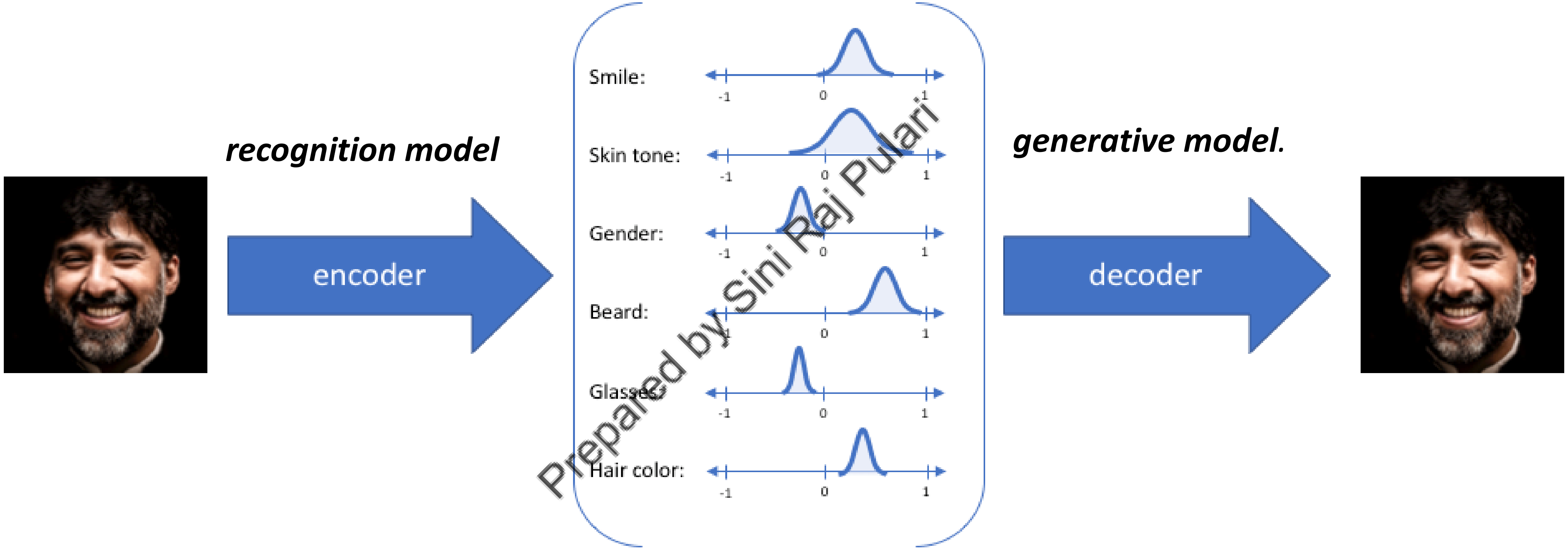
- Those are not directly observable , but are the true explanatory factors that are making the observed variables



- *Each latent attribute* for a given input as a probability distribution. When decoding from the latent state, randomly sample from each latent state distribution to generate a vector as input for our decoder model



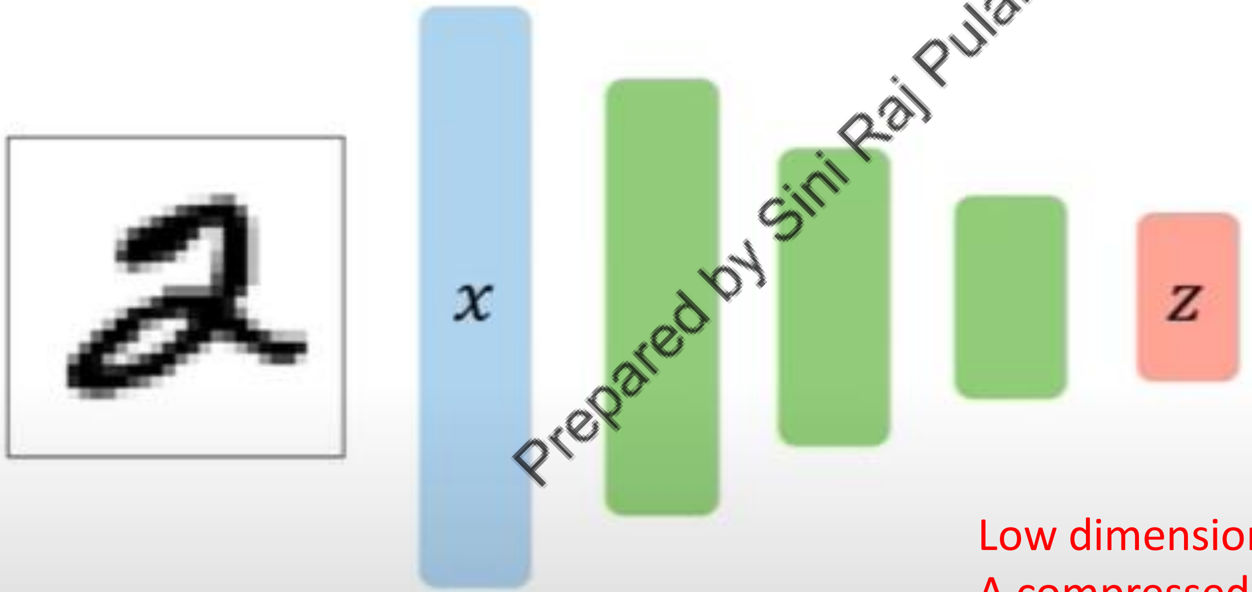
- For any sampling of the latent distributions, the decoder model should be able to accurately reconstruct the input



Latent attributes

Autoencoders: background

Unsupervised approach for learning a **lower-dimensional** feature representation from unlabeled training data



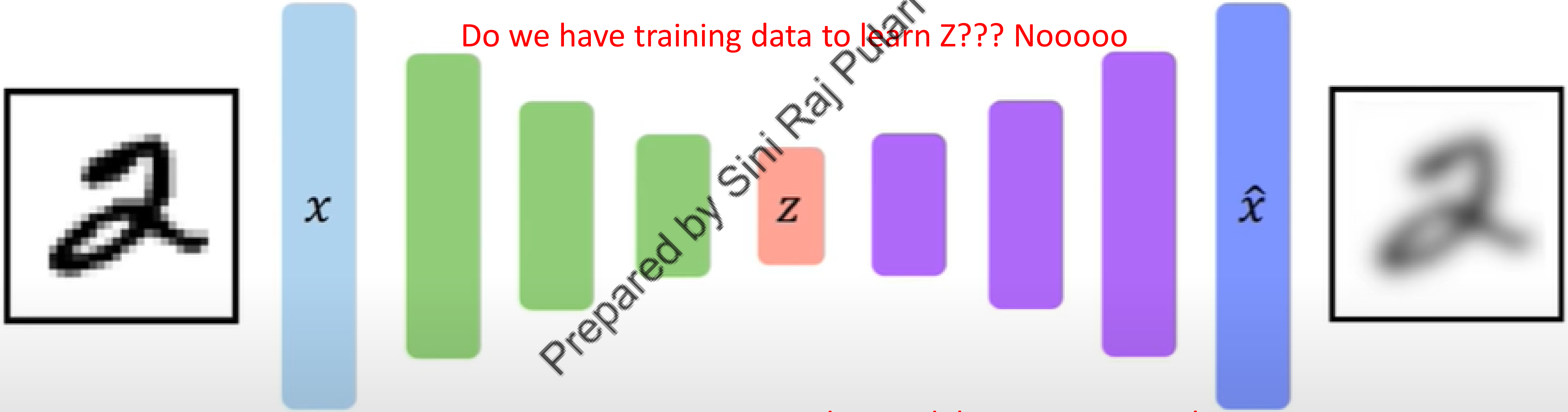
Low dimensional latent space …
A compressed representation

"Encoder" learns mapping from the data, x, to a low-dimensional latent space, z

Autoencoders: background

How can we learn this latent space?
Train the model to use these features to **reconstruct the original data**

Do we have training data to learn Z??? Nooooo



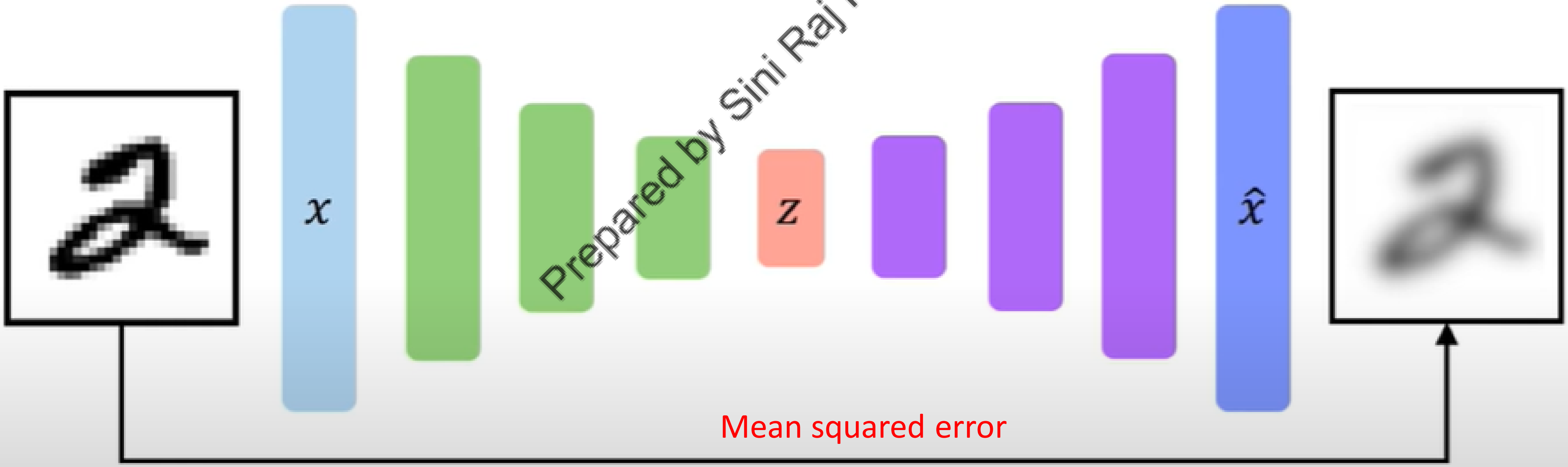
Use a Decoder Model to reconstruct the
image from learned latent space representation

"Decoder" learns mapping back from latent, z, to a reconstructed observation, $\hat{x}$

Autoencoders: background

How can we learn this latent space?

Train the model to use these features to **reconstruct the original data**



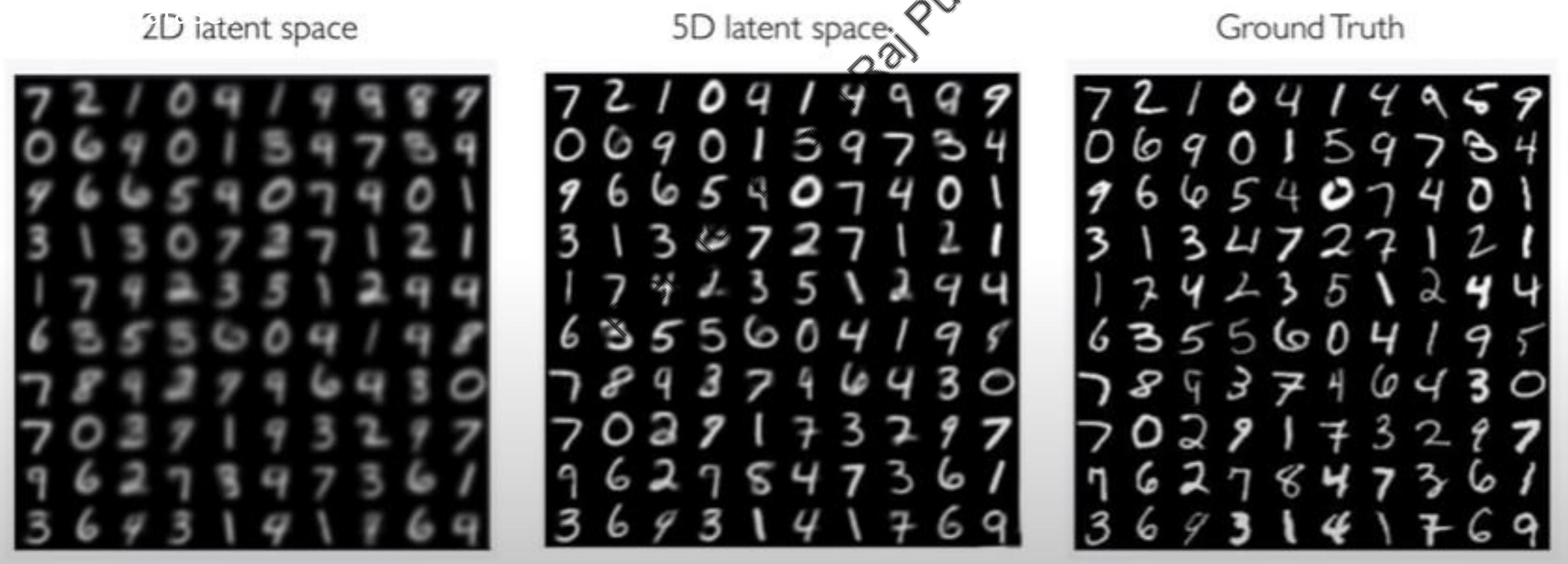
Loss function doesn't use any labels!!

$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

Just x and x^, no labels…
Hence unsupervised!!!

Dimensionality of latent space → reconstruction quality

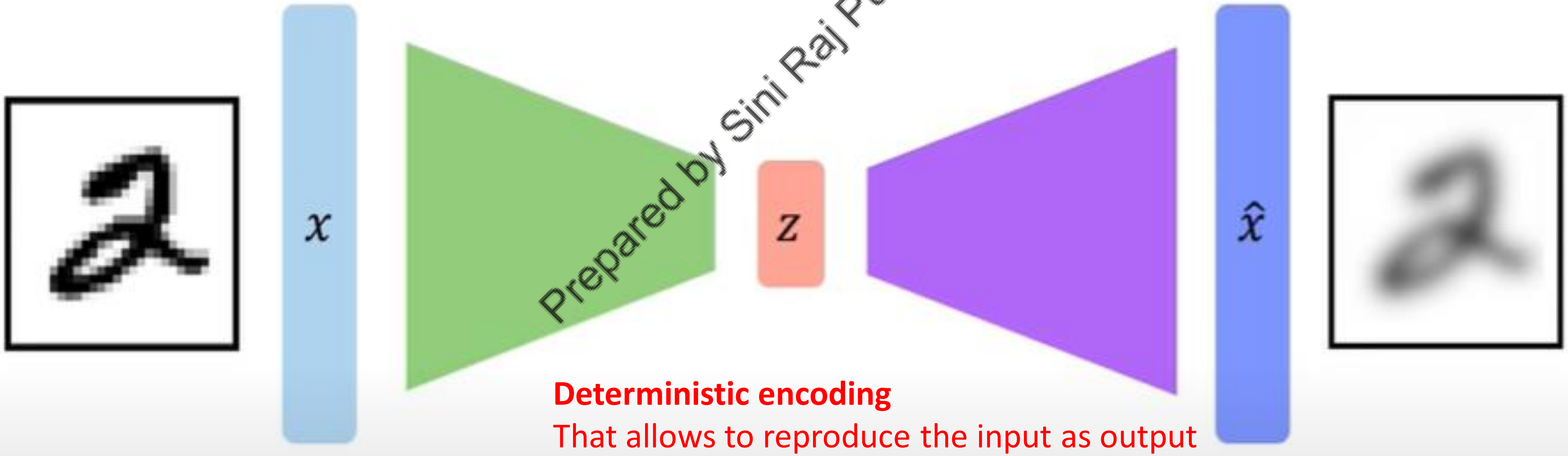
Autoencoding is a form of compression!
Smaller latent space will force a larger training bottleneck



Lower the dimensionality of latent space , poorer the quality of reconstructed image

Variational Autoencoders (VAEs)

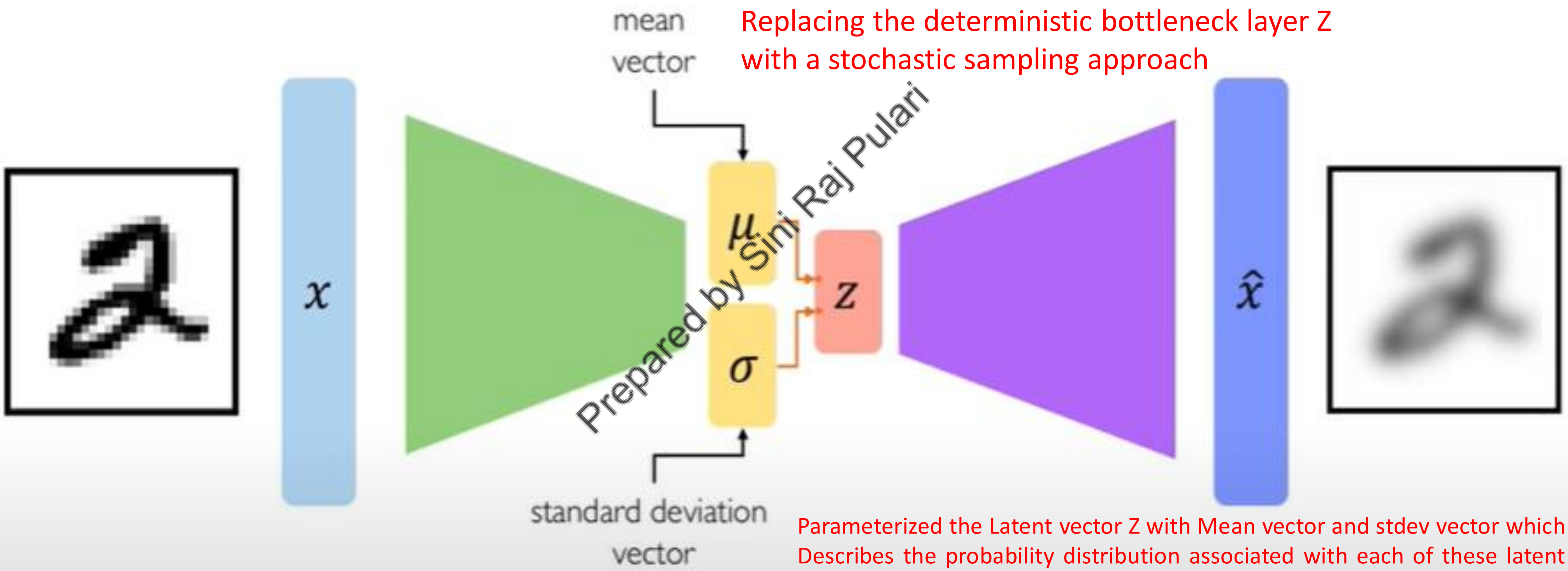
Traditional autoencoders



Deterministic encoding
That allows to reproduce the input as output
as far as the weights are unchanged

VAEs: key difference with traditional autoencoder

Replacing the deterministic bottleneck layer Z with a stochastic sampling approach

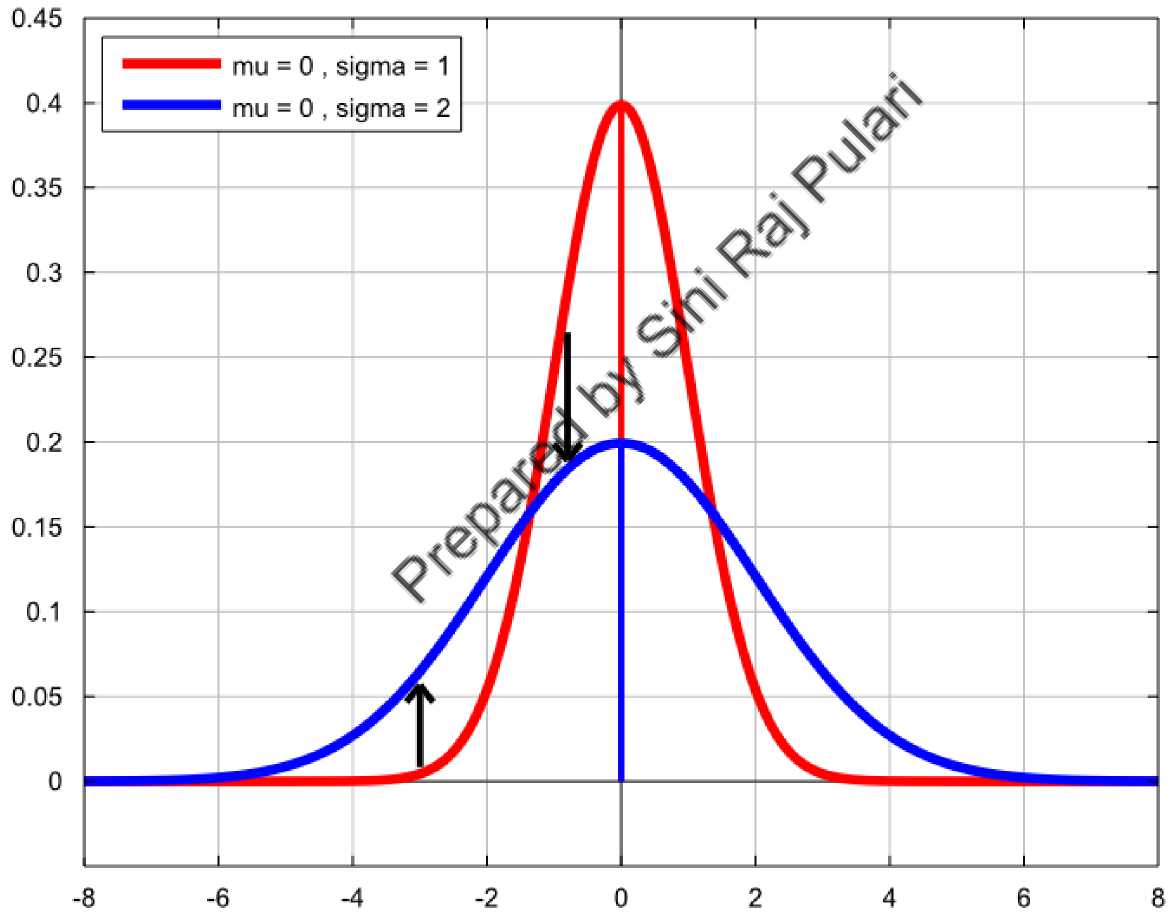


Parameterized the Latent vector Z with Mean vector and stdev vector which Describes the probability distribution associated with each of these latent variables Z

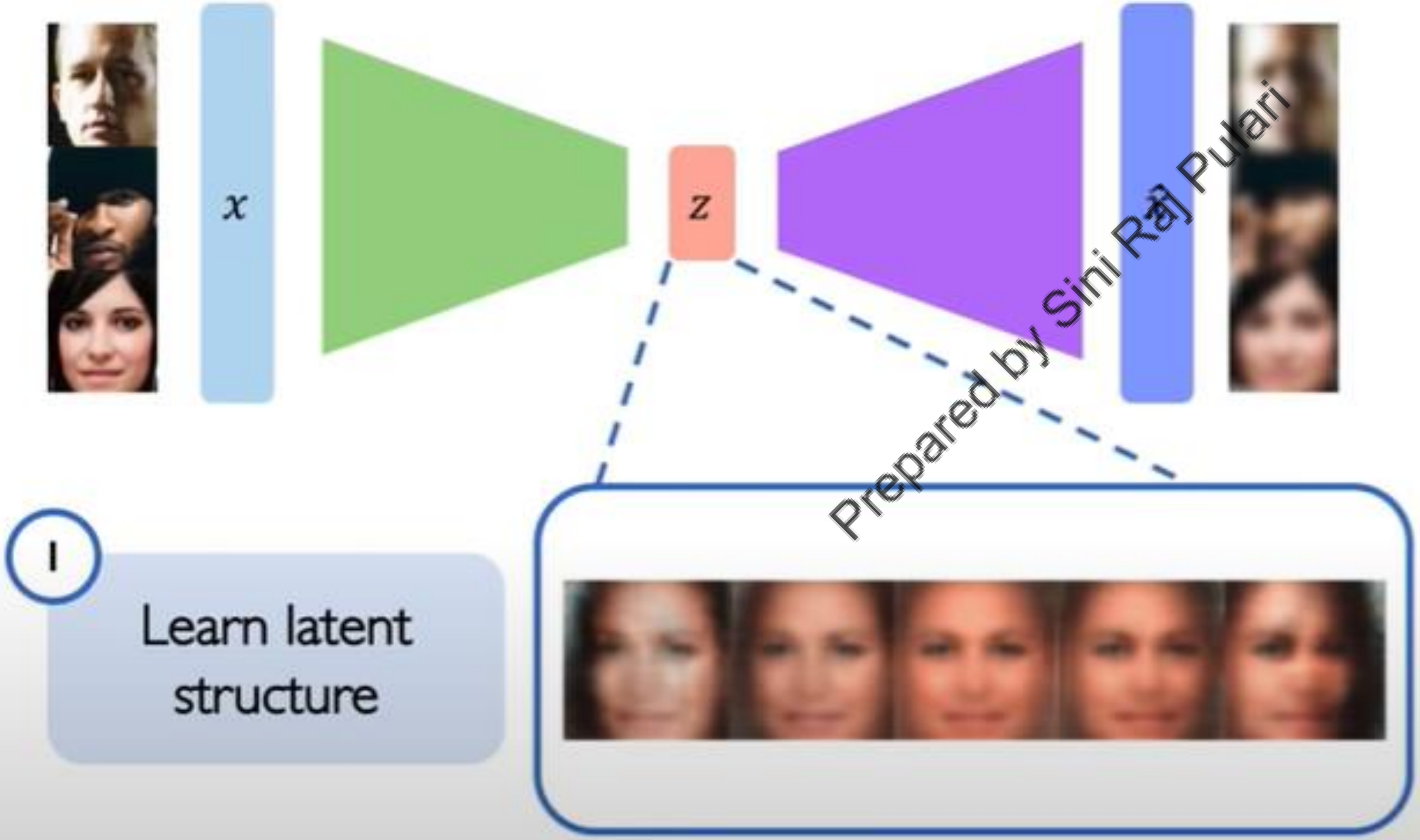
Variational autoencoders are a probabilistic twist on autoencoders!

Sample from the mean and standard dev. to compute latent sample

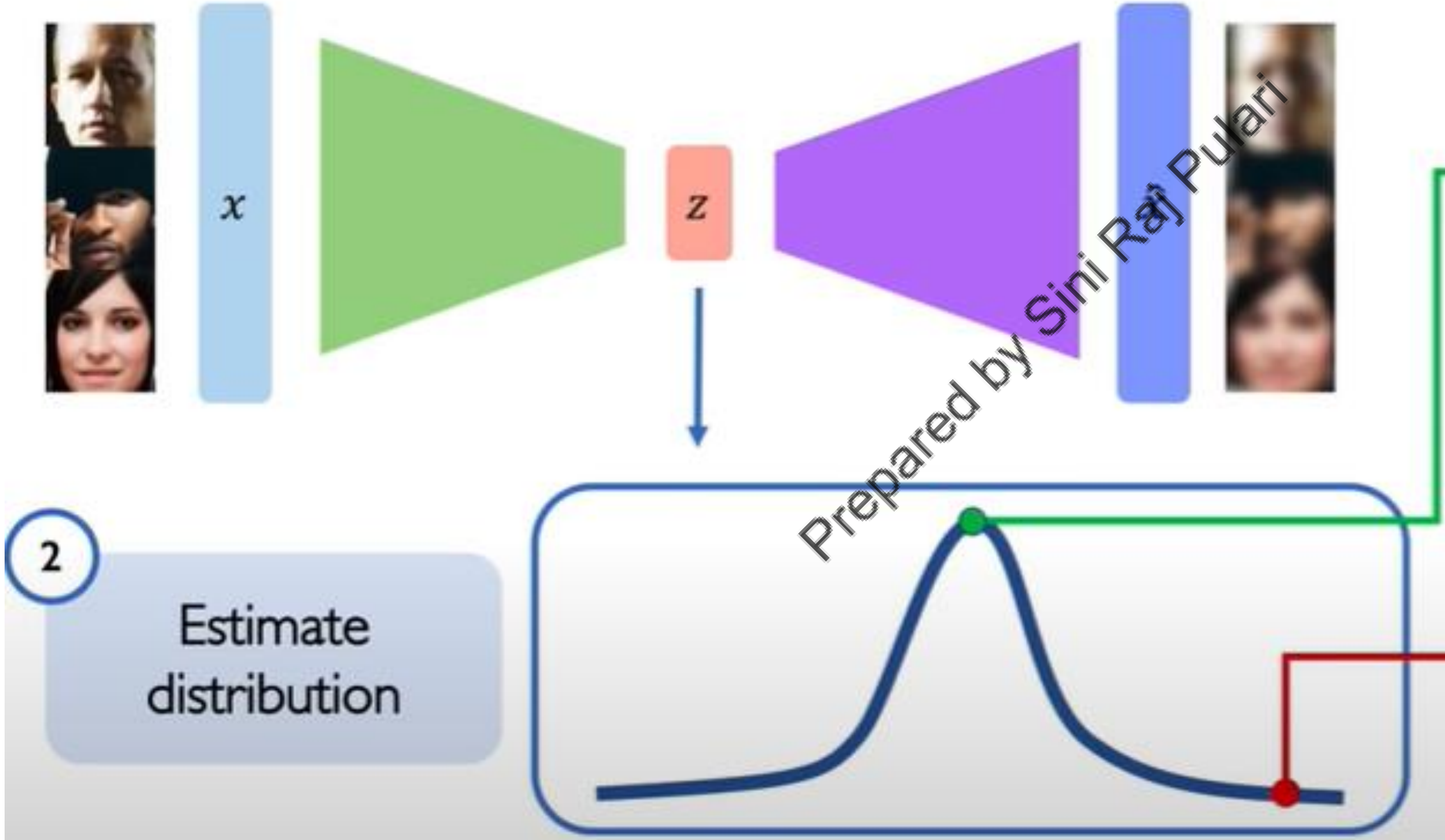
Mean and STD on ND



Mitigating bias through learned latent structure



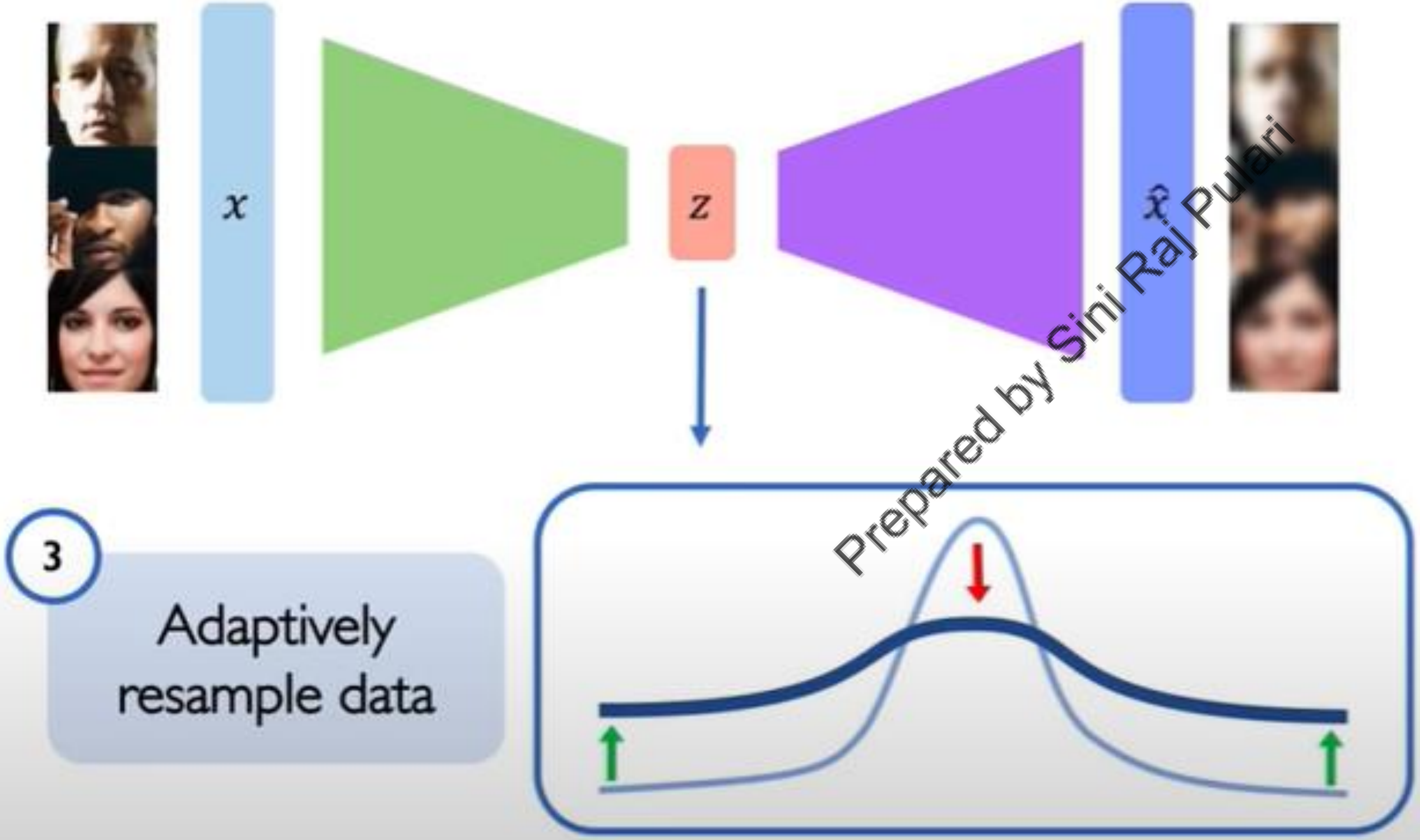
Mitigating bias through learned latent structure



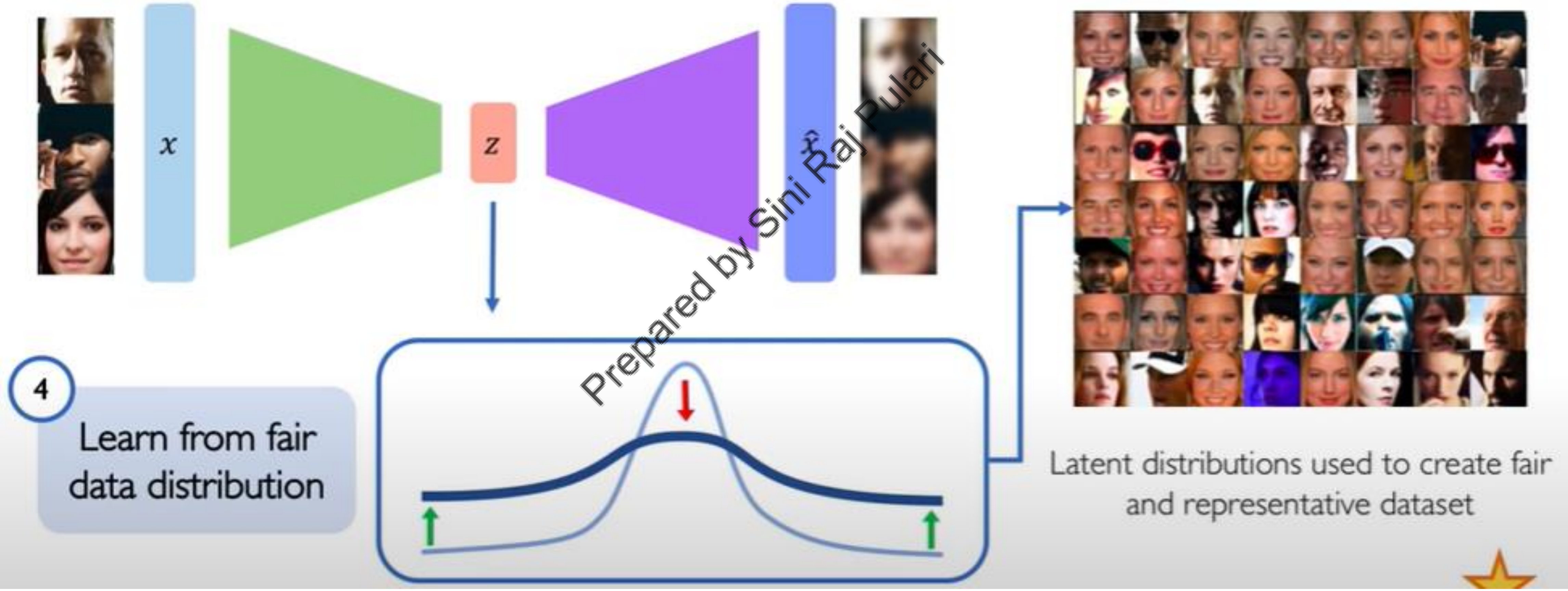

Homogeneous skin color, pose


Diverse skin color, pose, illumination

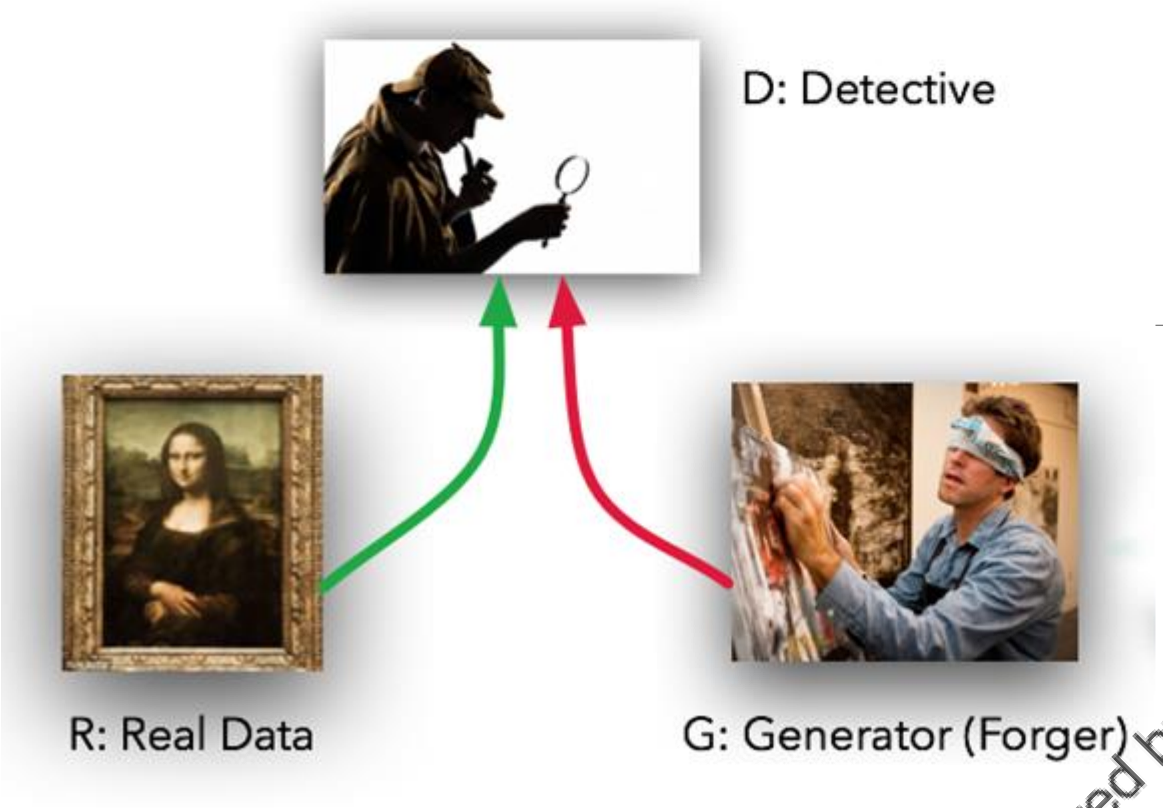
Mitigating bias through learned latent structure

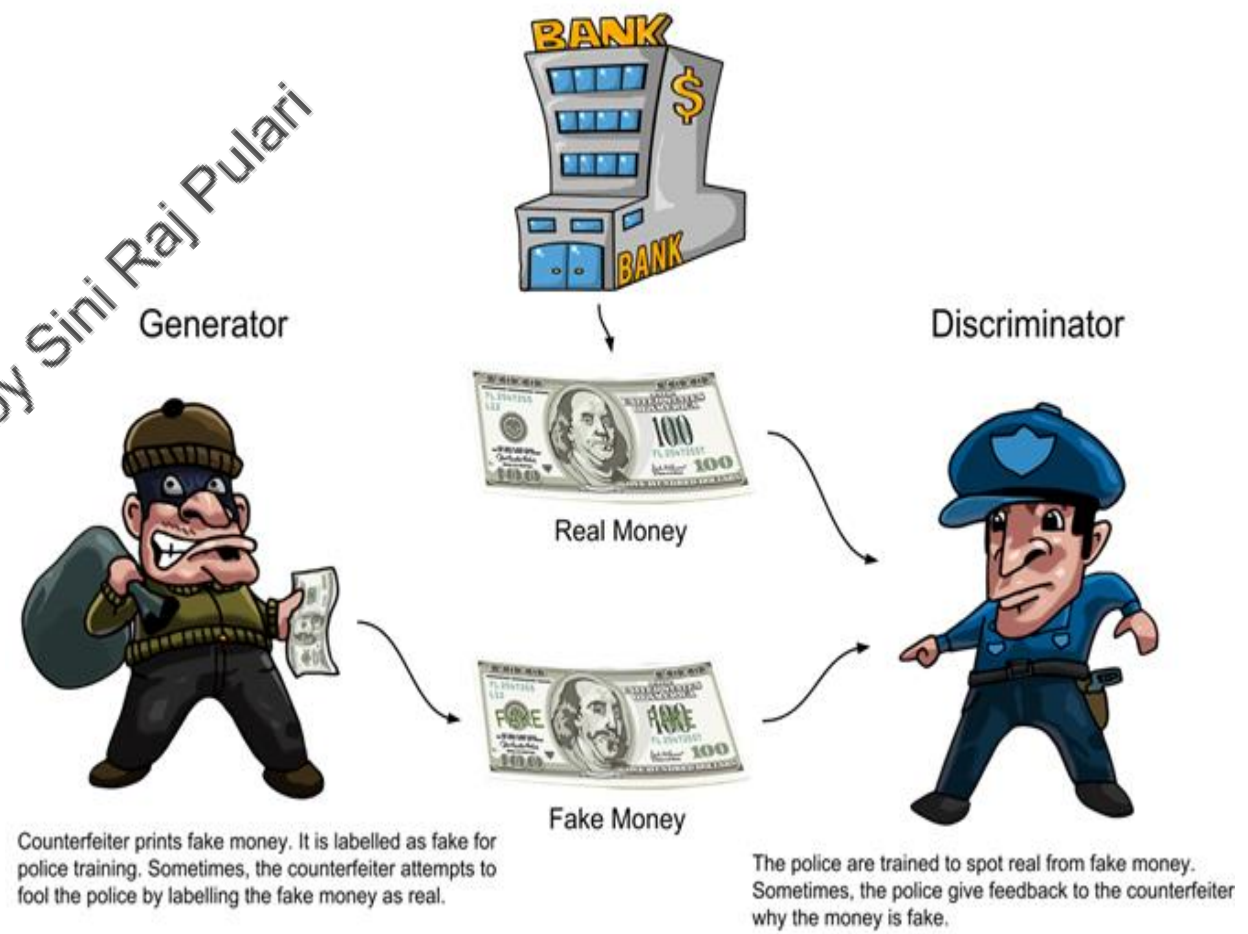


Mitigating bias through learned latent structure



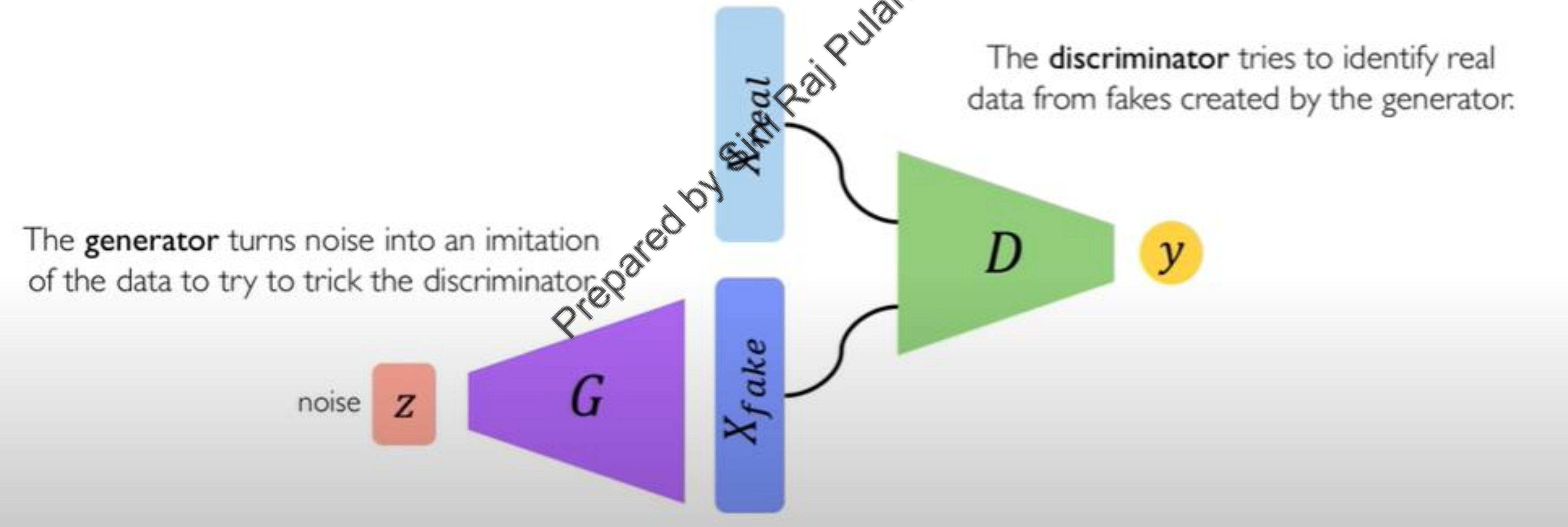
Deep Generative Adversial Networks


D: Detective
R: Real Data
G: Generator (Forger)

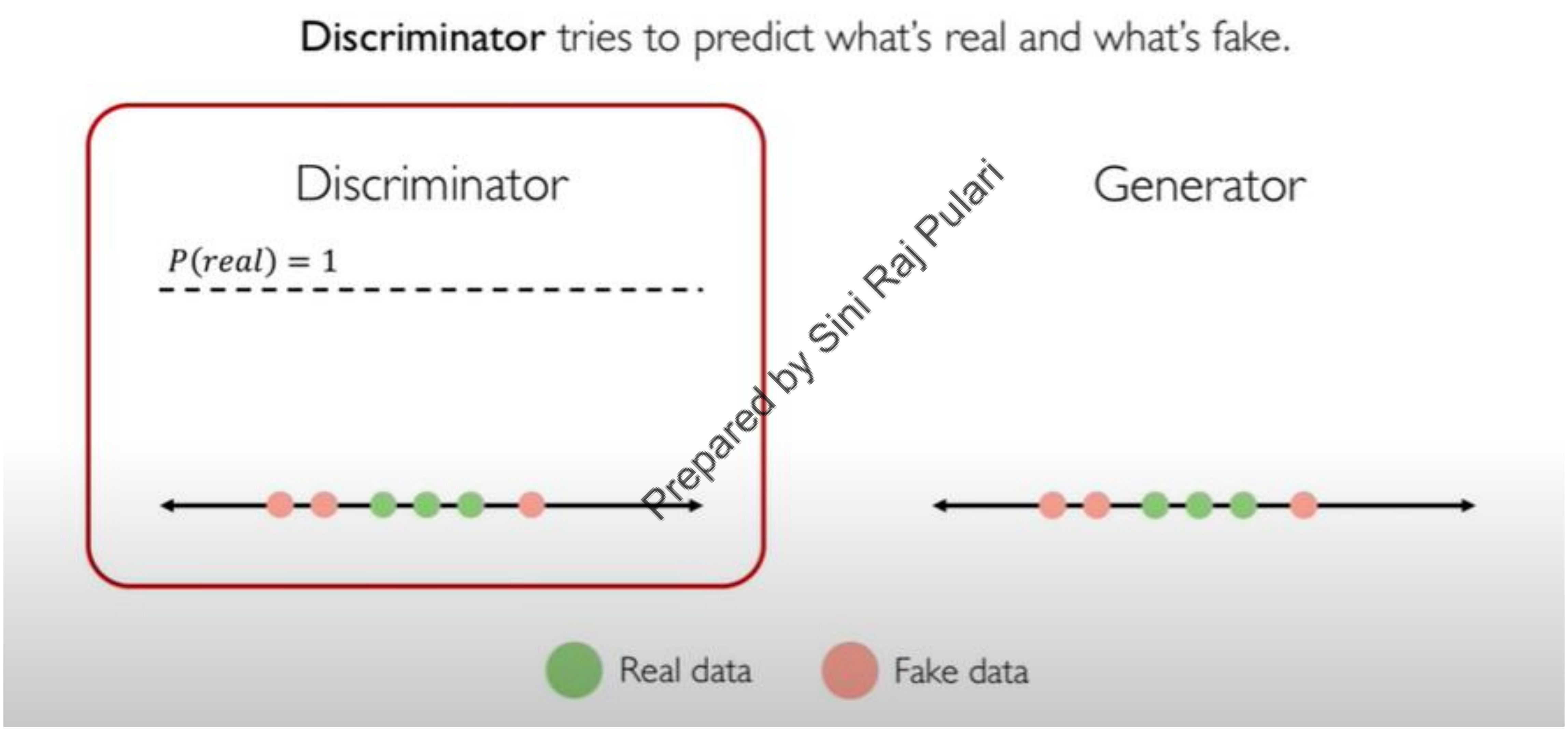

Generator
Discriminator
Real Money
Fake Money
Counterfeiter prints fake money. It is labelled as fake for police training. Sometimes, the counterfeiter attempts to fool the police by labelling the fake money as real.
The police are trained to spot real from fake money. Sometimes, the police give feedback to the counterfeiter why the money is fake.

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



Intuition behind GANs



Discriminator and Generator acts as adversaries making each other better in course of time

Intuition behind GANs

Generator tries to improve its imitation of the data.

Discriminator works well and train in such a way it can identify the fake ones clearly

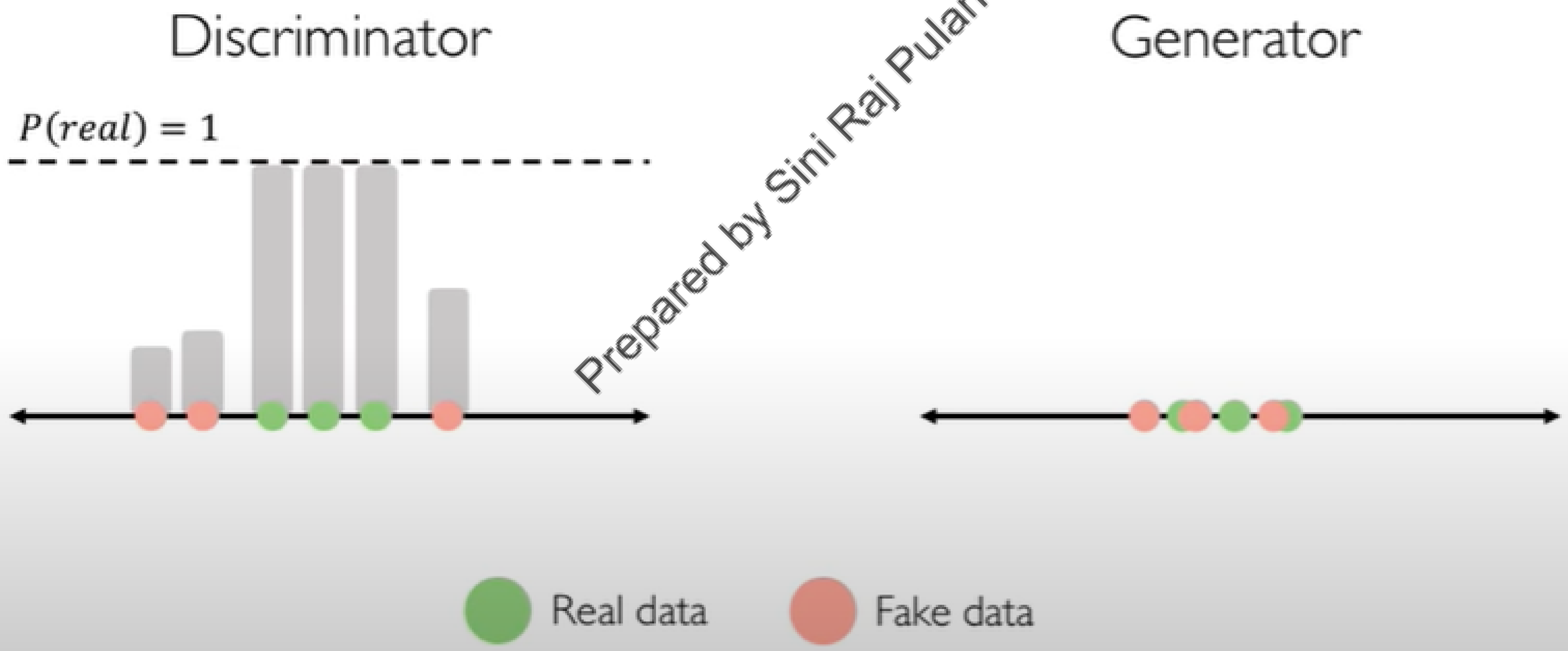
Intuition behind GANs

Generator tries to improve its imitation of the data.

Now generator tries to move the fake points move as close to real points that discriminator would be unable to identify

Intuition behind GANs

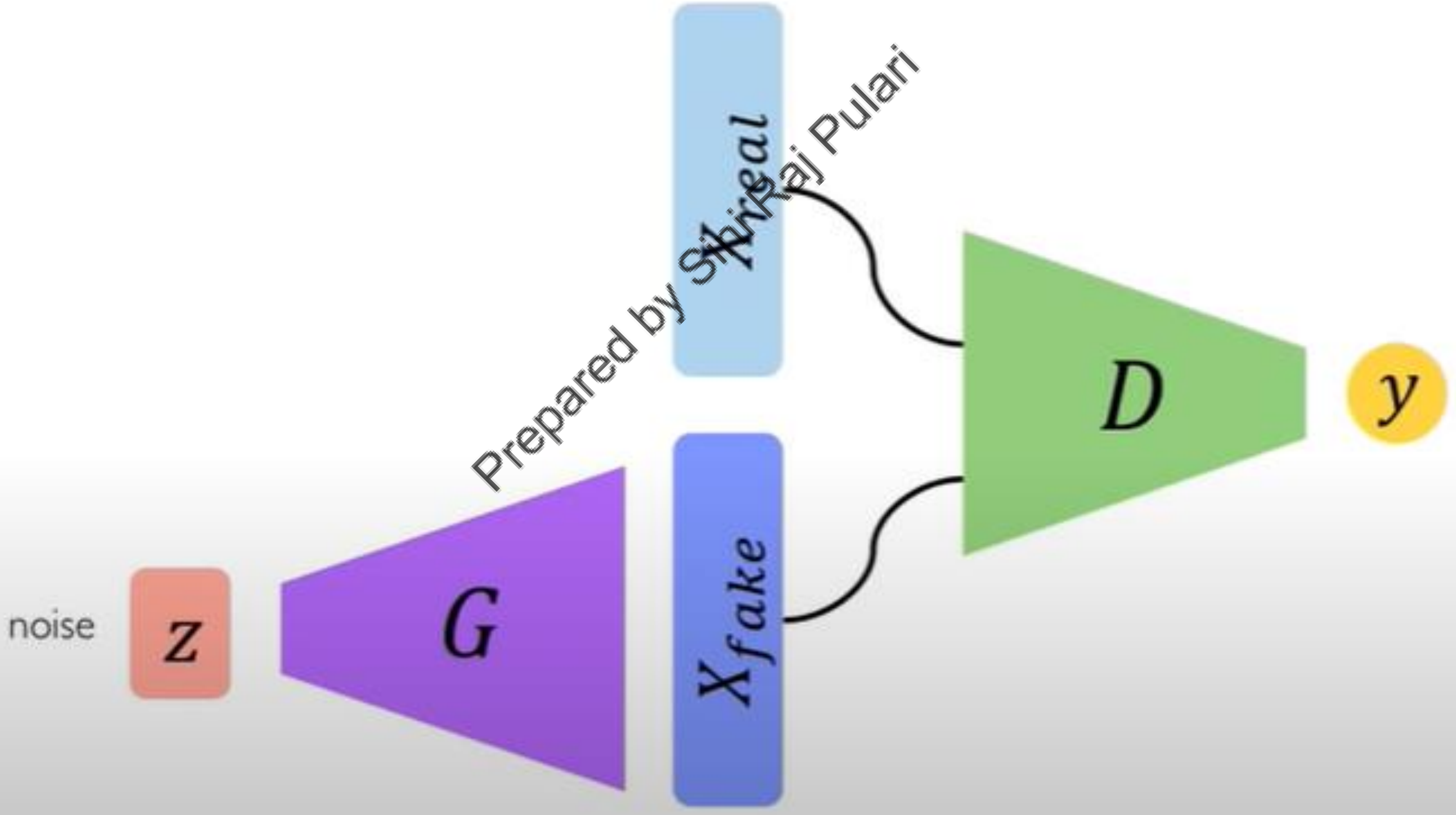
Discriminator tries to identify real data from fakes created by the generator.
Generator tries to create imitations of data to trick the discriminator.



Eventually discriminator will find it difficult to distinguish between fake and real

Generating new data with GANs

After training, use generator network to create **new data** that's never been seen before.



Some Applications of GAN

Applications of GAN

Generating Images

Image Modification

Super Resolution

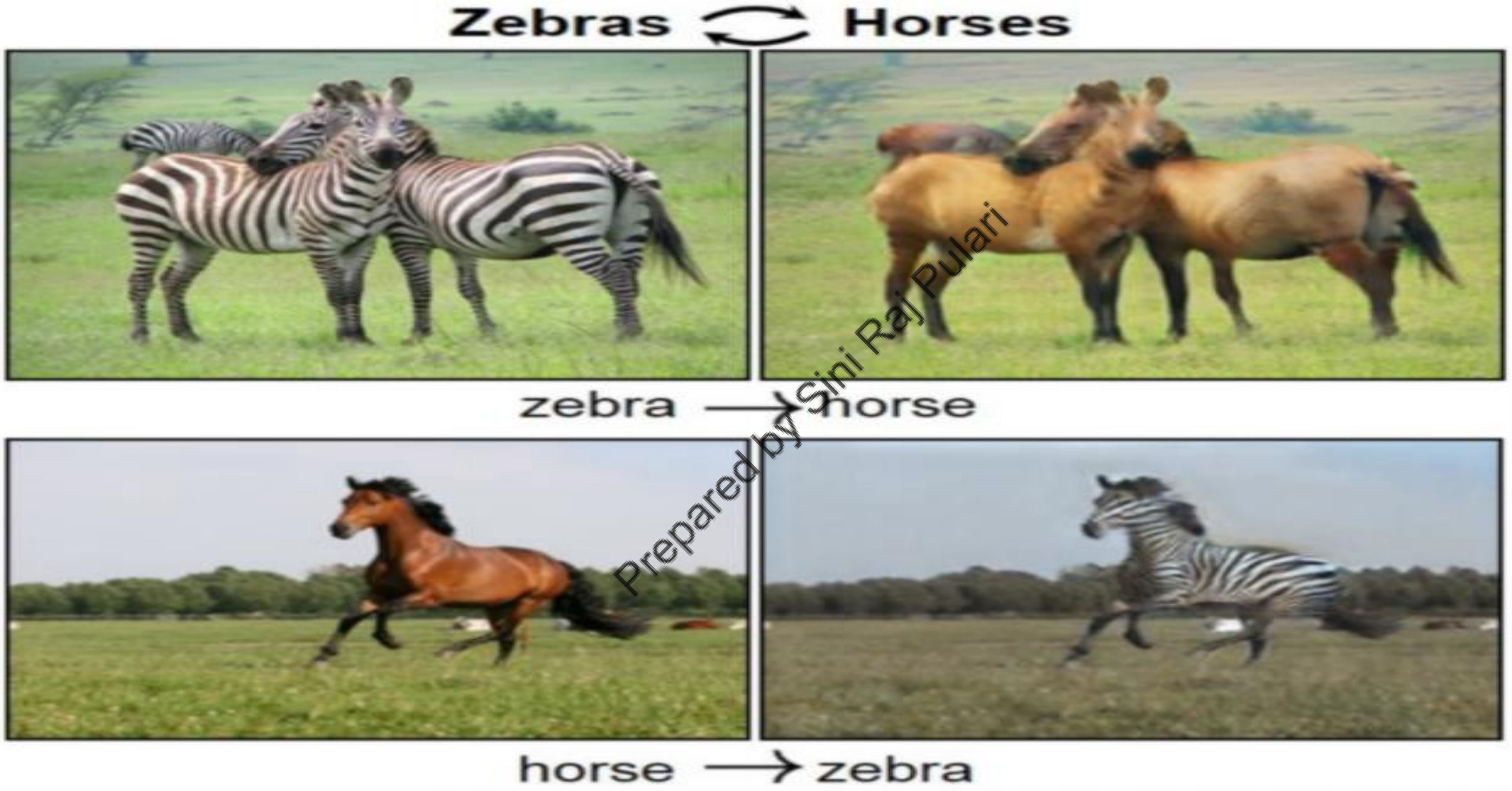
Assisting Artists

Photo-Realistic Images

Speech Generation

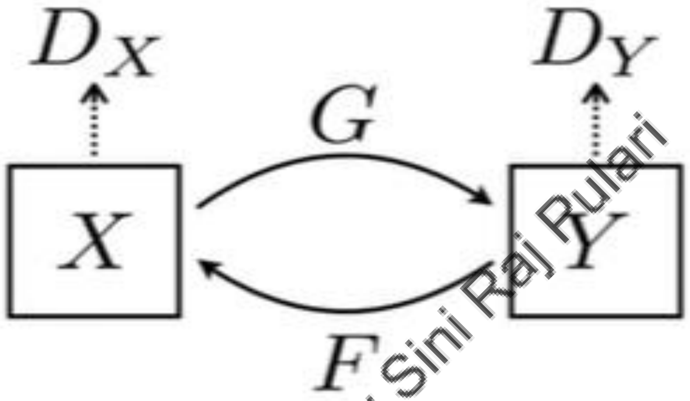
Face Ageing

Image Modification and Generation



CycleGAN: domain transformation

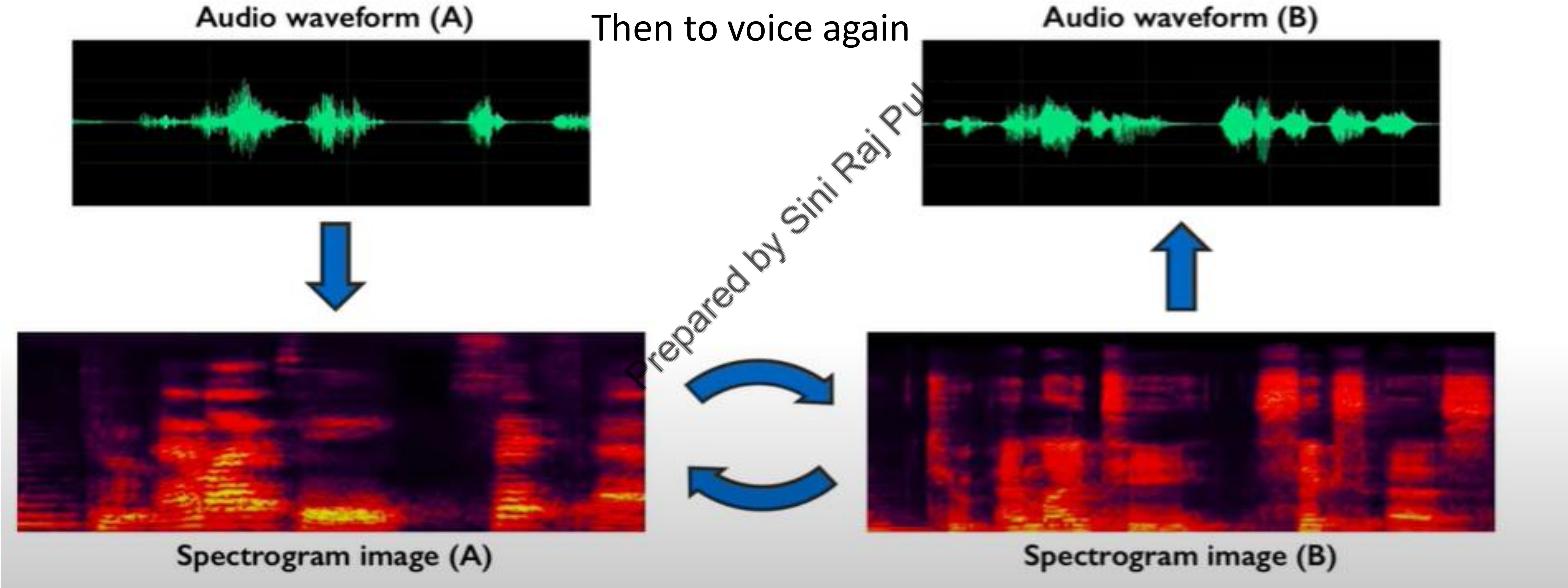
CycleGAN learns transformations across domains with unpaired data.



Cycle GAN

CycleGAN: Transforming speech

Voice converted to image
Then to voice again

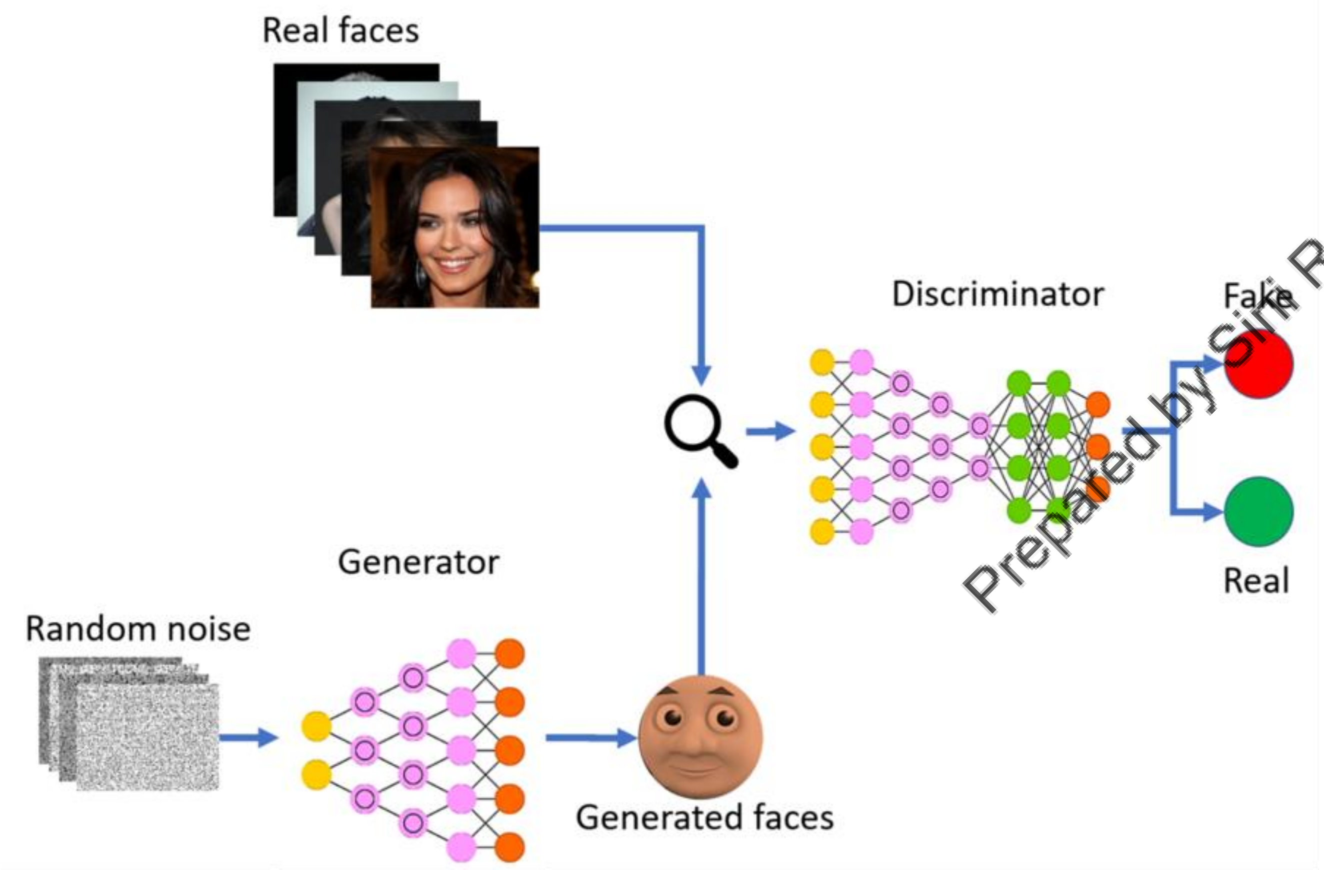


Let’s watch a video now

https://youtu.be/l82PxsKHxYc

https://youtu.be/l82PxsKHxYc

StyleGANs



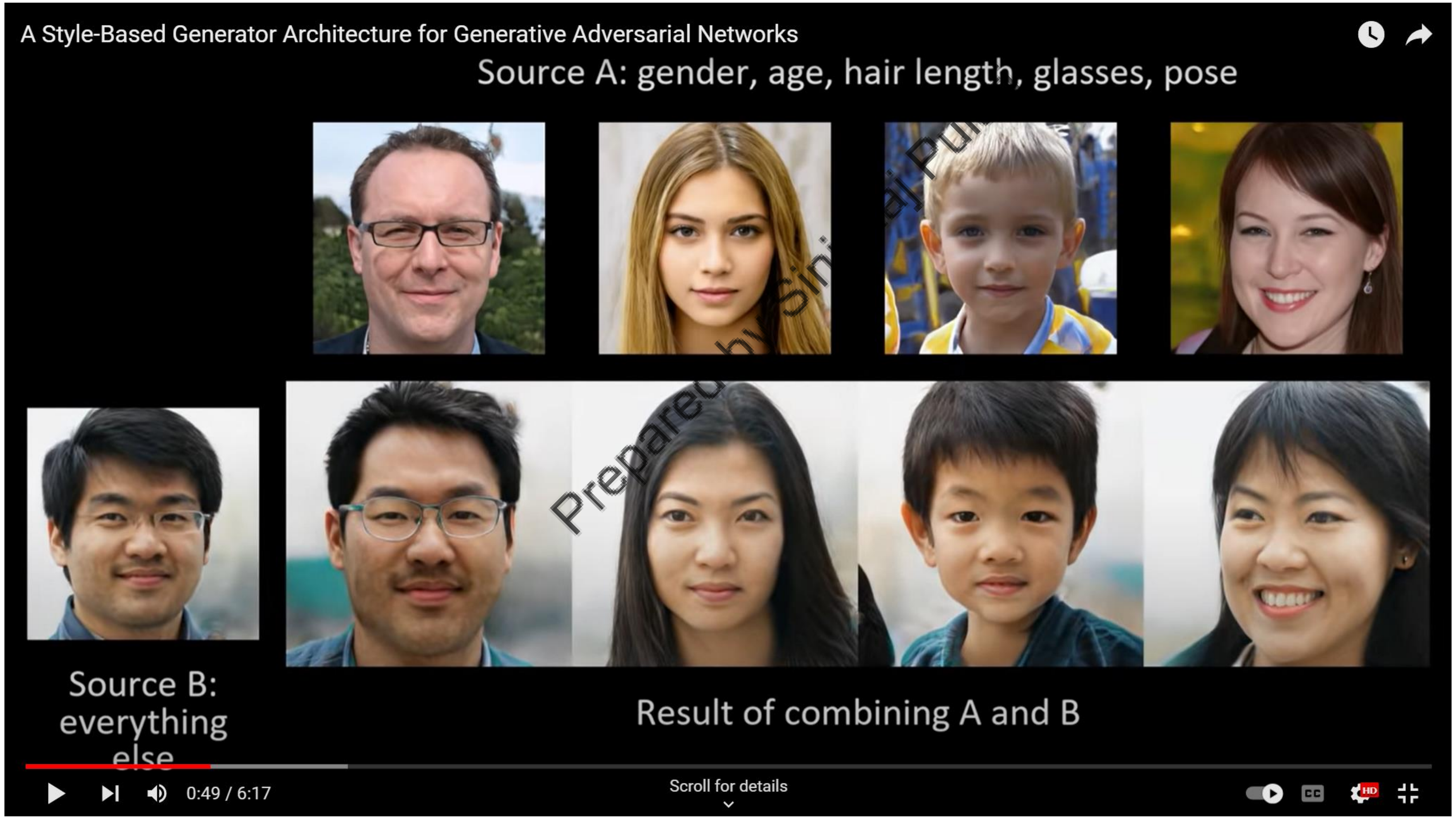
Changes in age

Changes in gender

Changes in gender

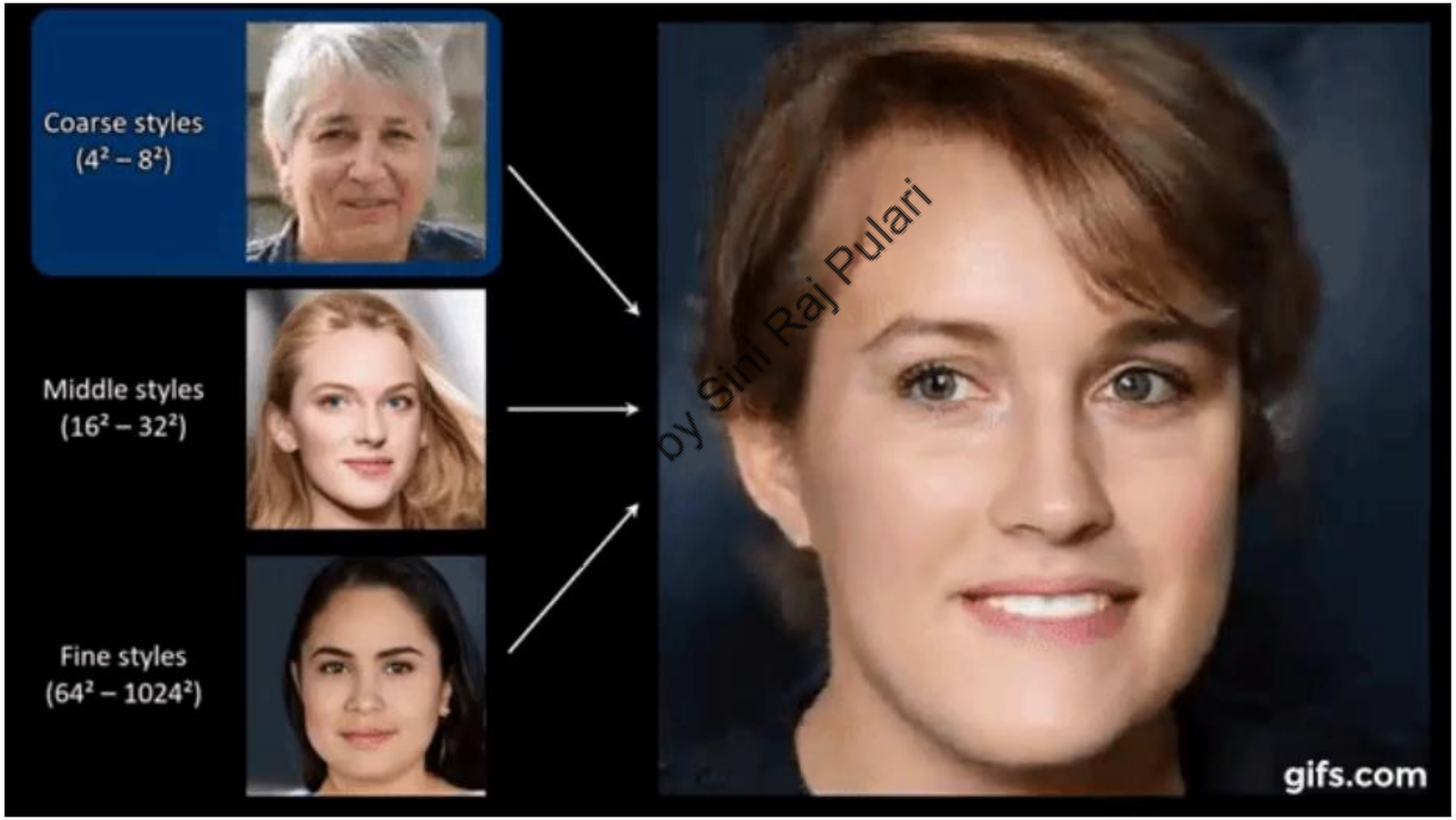
Changes in smiling

StyleGANs



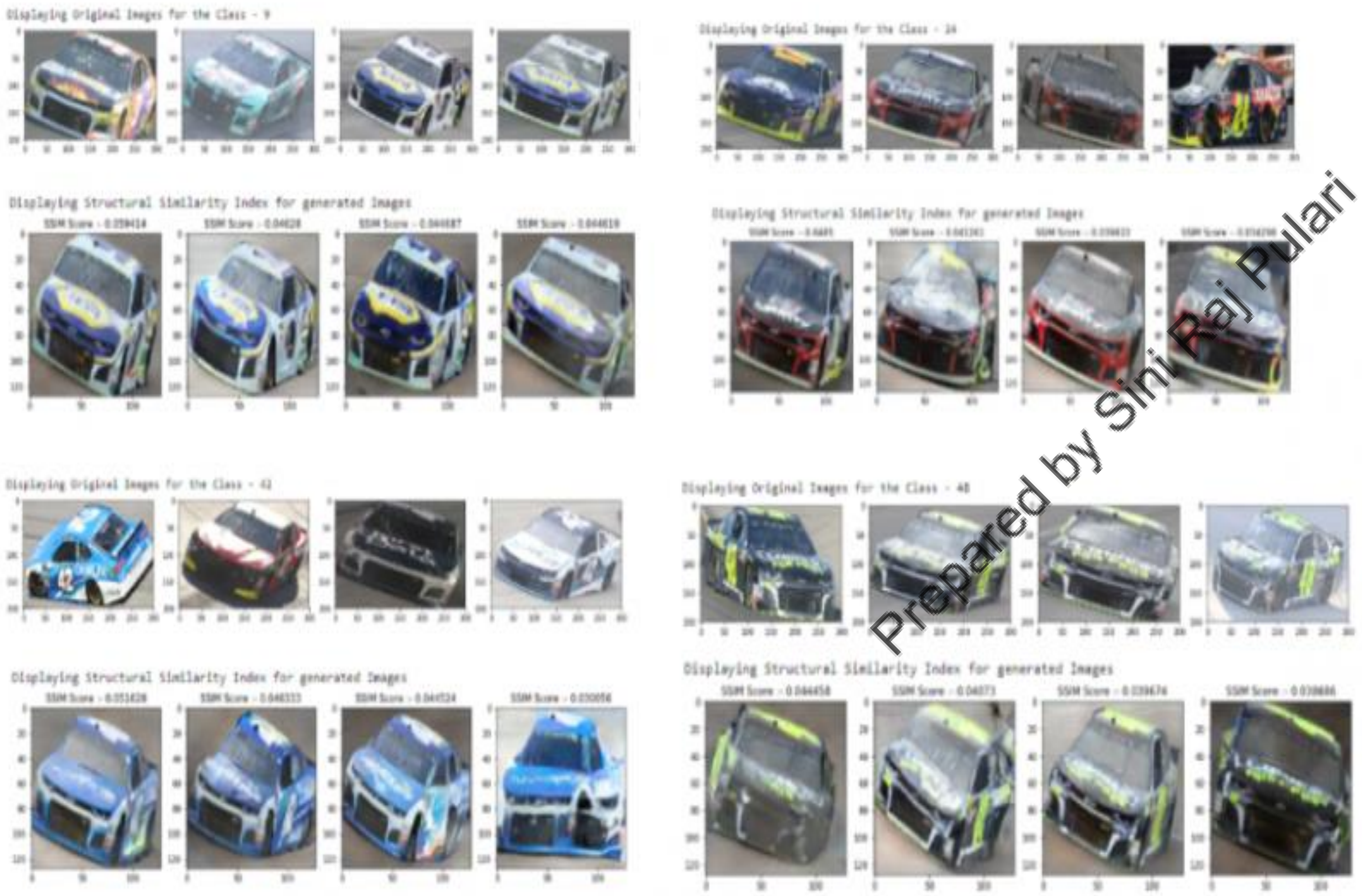
https://youtu.be/kSLJriaOumA

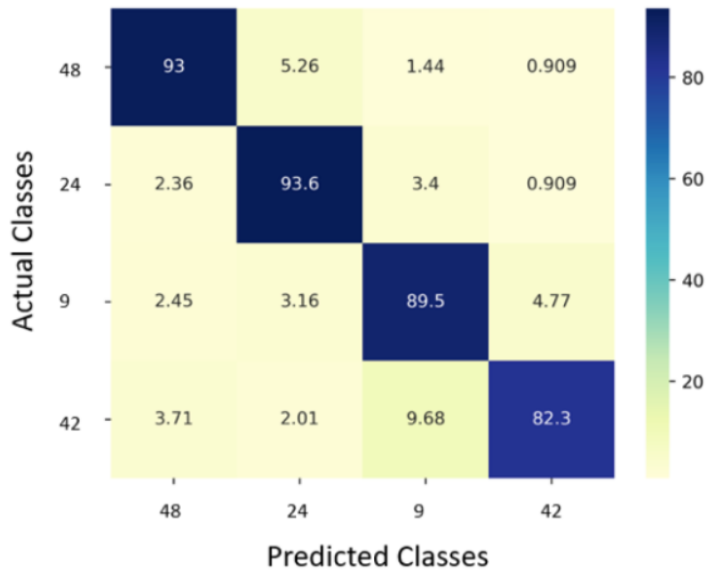
Style GANS



https://www.theaidream.com/post/generate-realistic-human-face-using-gan-1

Conditional GANS in Automotive Industry

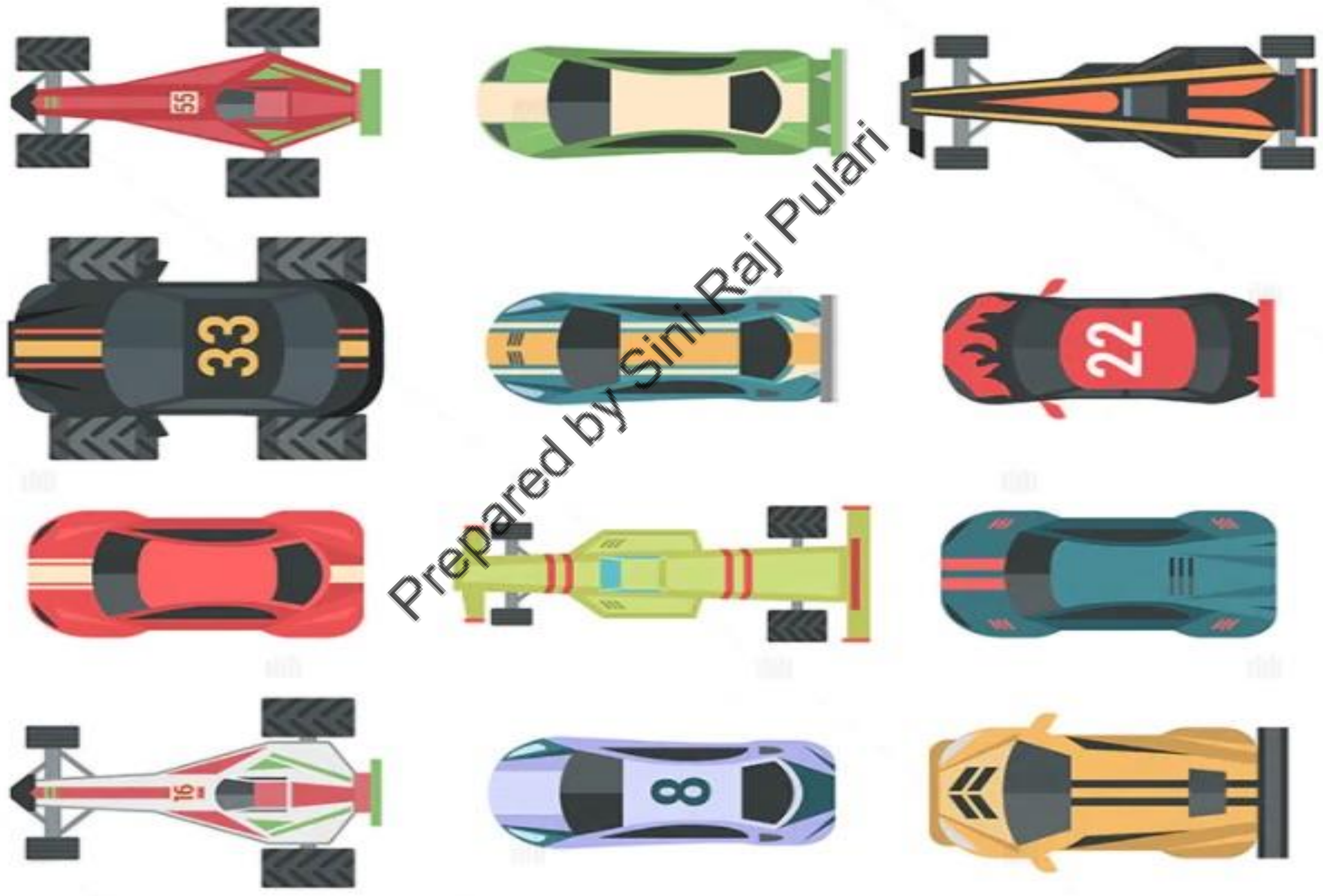




Overall the model had an average of 89.6% accuracy.

https://opendatascience.com/using-gans-to-generate-images-of-race-cars/

Generating Race Cars Designs -CGAN



https://analyticsindiamag.com/hands-on-guide-to-generate-car-models-using-deep-convolutional-gan/

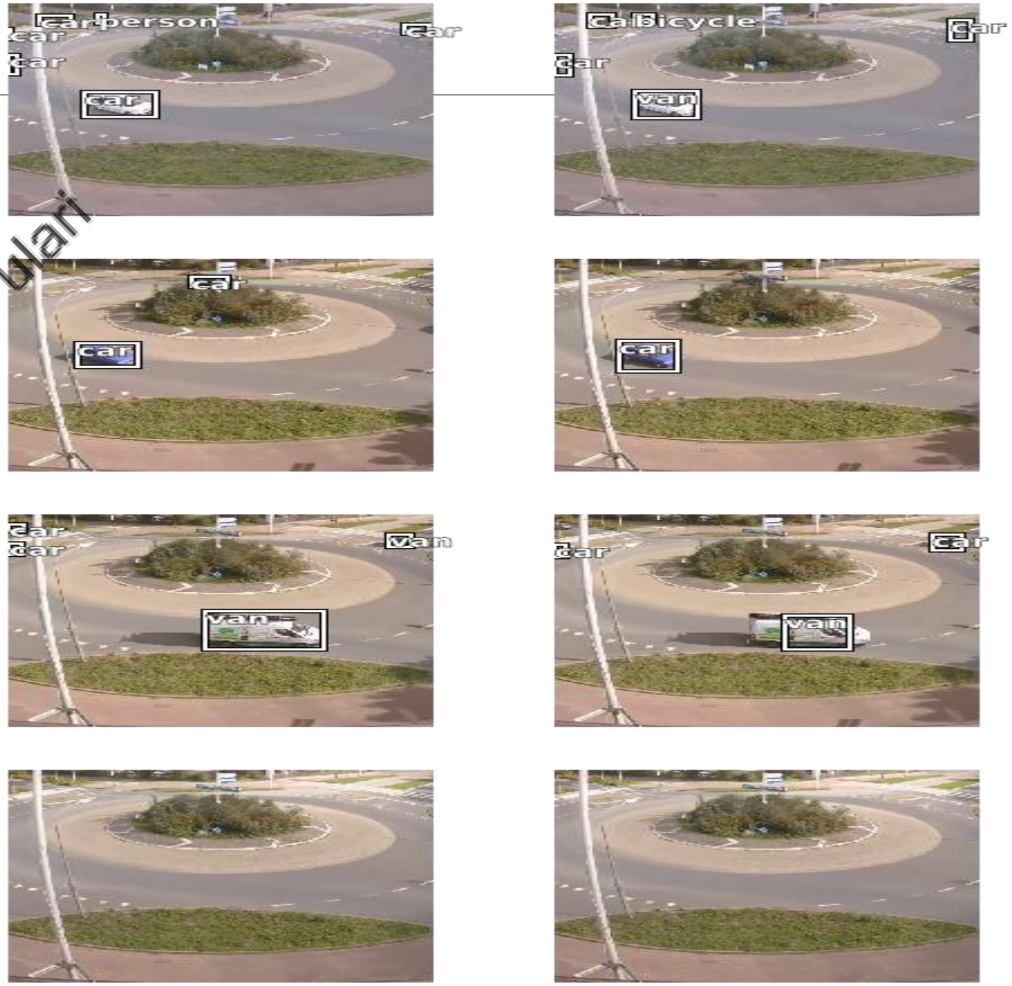
Deep Learning GAN

```python
all_classes = []
for i, bb in enumerate(data.train_ds.y):
    all_classes += bb.data[1].tolist()

df = pd.value_counts(all_classes, sort=False)
df.index = [data.classes[i] for i in df.index]
df
```

```
bicycle       266
bus            19
car           756
motorcycle     33
person         24
scooter         6
tempo           1
tractor         4
truck          30
van            69
dtype: int64
```


Ground truth/Predictions



Design Cars - Generating Car Images, Car Body Classifier , Re-construction Using

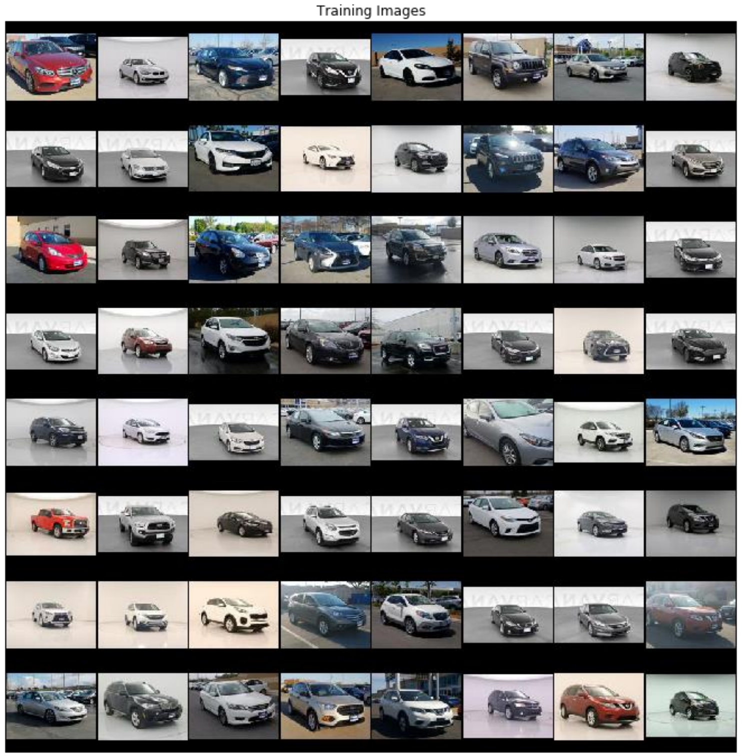
Evolution of 128×128 generated images from DCGAN using SELU (From left to right: Evolution for epochs 1–70, Epochs 71–120, Epochs 121–170)

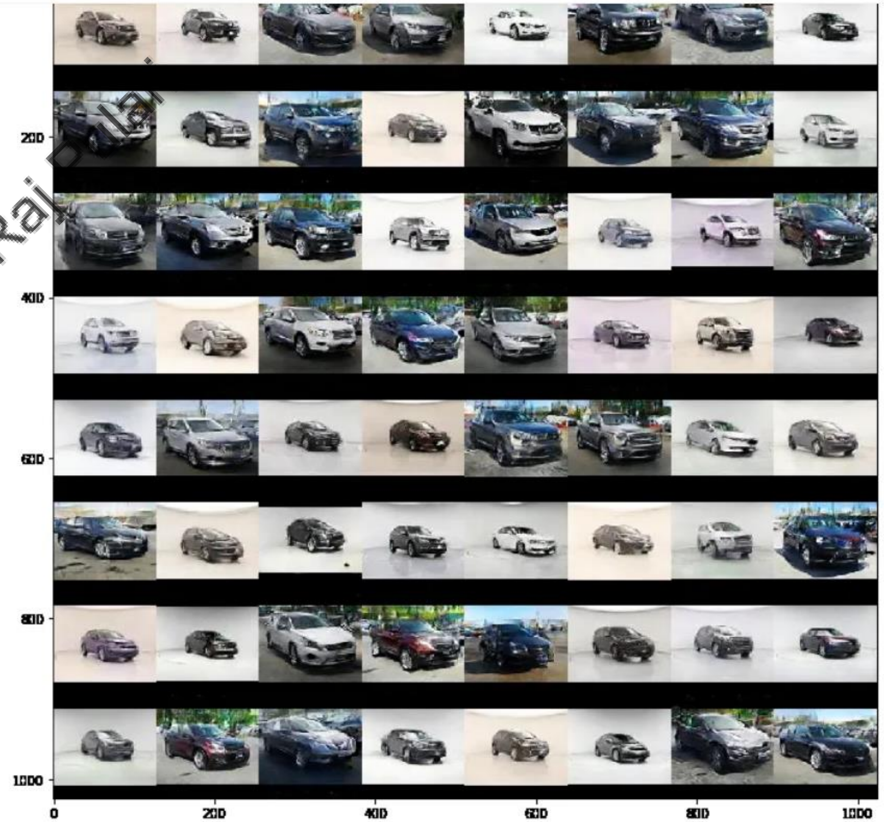
Github Repository: https://github.com/asoomar/car-design-generation

DCGANS – Generation of Car Images

Training Images

Generated Images





128×128 Images Generated with the DCGAN using SELU after 170 epochs

DCGANS – Car Body Classifier

Classified GAN Images

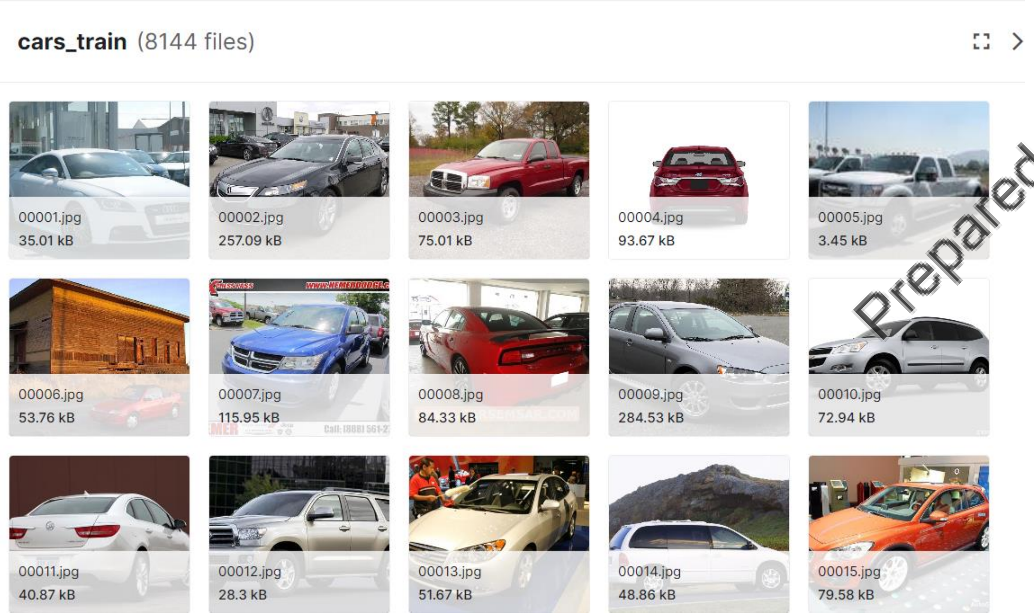
DCGANS- Image Reconstruction

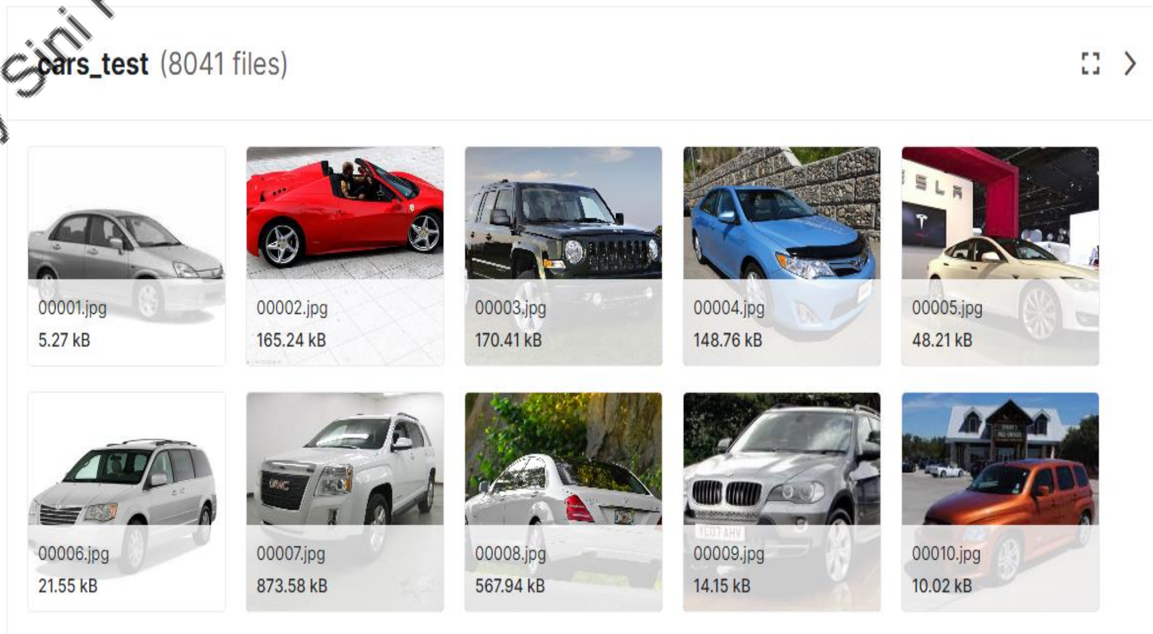
Original

Reconstruction

Stanford Cars Dataset

- The Cars dataset contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of Make, Model, Year, e.g. 2012 Tesla Model S or 2012 BMW M3 coupe.





https://www.kaggle.com/datasets/eduardo4jesus/stanford-cars-dataset

GAN for generating Car imag... Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

Cancel Run Code

Draft Session (34m)

I just made a GAN for cars from this dataset, cause I was bored. The dataset isn't meant for it, But I did it anyways :D

Code uses pytorch for making a GAN

+ Code + Markdown

[1]:

```python
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 5GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

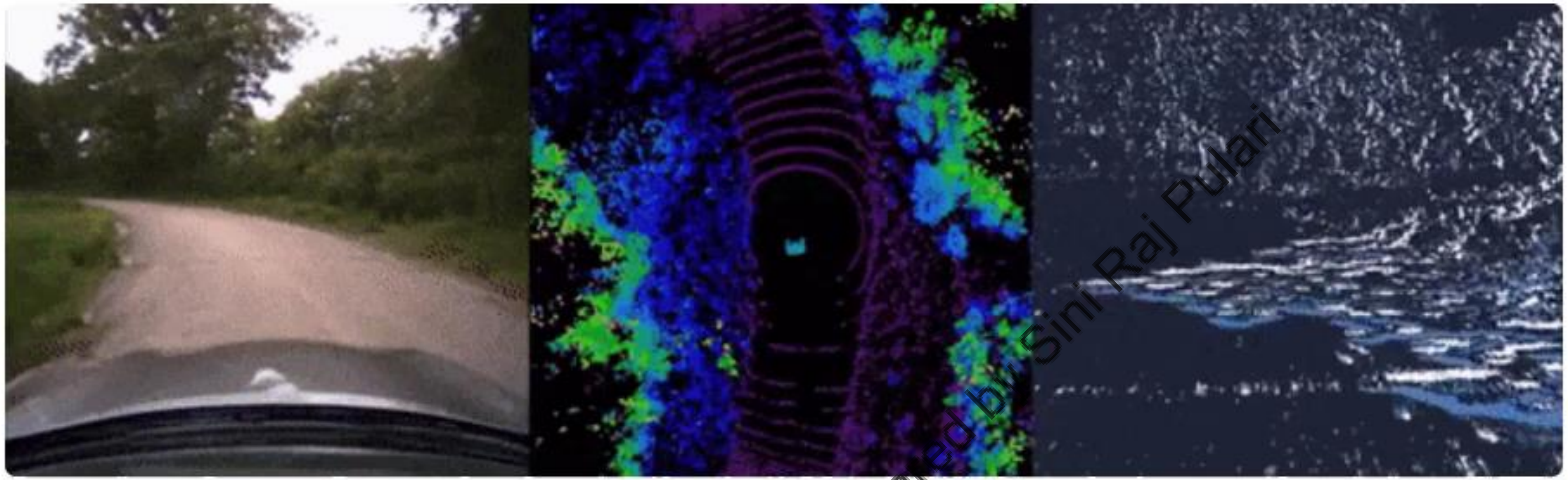
```
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/05938.jpg
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/06122.jpg
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/04168.jpg
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/02371.jpg
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/04377.jpg
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/00767.jpg
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/07457.jpg
/kaggle/input/stanford-cars-dataset/cars_test/cars_test/02360.jpg
```

Just to get Motivated …

https://youtu.be/pUhckFVXN2A

LIDAR + GAN + VAE Research @MIT

Data-driven Synthesis of Learning Environments



This project develops a platform for data-driven simulation of embodied agent perception and closed-loop control learning directly from real-world data. Our research produced the first successful instance of training an end-to-end autonomous driving control policy using reinforcement learning entirely in simulation, and then deploying this policy directly in the real world on new roads and environments.

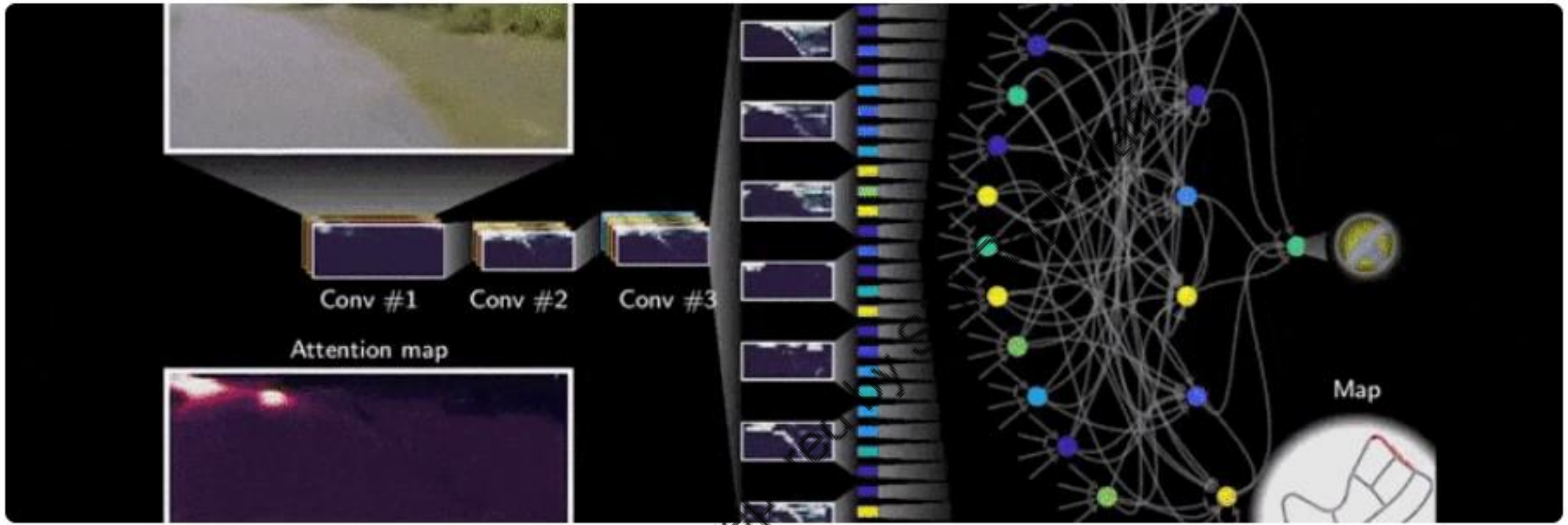
Light Detection and Ranging (LiDAR) sensors are used to detect objects and estimate distances, uses Laser

RADAR uses Radio Waves

- https://www.csail.mit.edu/news/more-efficient-lidar-sensing-self-driving-cars

Newest Research from MIT

Interpretable and Robust Neural Models



In this project we develop a novel continuous-time neuron model with a dynamically changing (liquid) time-constant to enable improved representation learning. Our models take inspiration from biology and enable 100x smaller and more auditable networks for end-to-end autonomous driving and aerial drone navigation.

END TO END LEARNING OF AUTONOMOUS SYSTEMS

End-to-End Learning of Robotic Actuation

Autonomous systems need the ability to learn from and adapt to their environment. In this project, we are developing algorithms that learn directly from raw sensory data (e.g. from a RGB video camera) to navigate a full-scale autonomous vehicle through complex scenarios.

Resources

https://www.youtube.com/watch?v=BtpkFqDwfa4&pp=ygUNdmVoaWNsZSArIEdBTg%3D%3D

https://machinelearningmastery.com/cyclegan-tutorial-with-keras/

https://analyticsindiamag.com/hands-on-guide-to-generate-car-models-using-deep-convolutional-gan/

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

https://github.com/danielSmagly/GAN-Vehicles

https://www.aipicasso.app/

https://www.tensorflow.org/tutorials/generative/deepdream

Research Articles:

VISTA 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles, Amini, A.*, Wang, T.*, Gilitschenski, I., Schwarting, W., Liu, Z., Han, S., Karaman, S., Rus, D.

Efficient and Robust LiDAR-Based End-to-End Navigation
Liu, Z.*, Amini, A.*, Zhu, S., Karaman, S., Han, S., Rus, D.

Queries....

Once we've accumulated knowledge, we can't go back to ignorance
: Plato’s Allegory of the cave!

Thank You…

SINI RAJ PULARI

REACH ME @ SINIKISHAN@GMAIL.COM

YOUTUBE CHANNEL : @SINZSMA

LINKEDIN : HTTPS://IN.LINKEDIN.COM/IN/SINIRAJPULARI