Game of Deception & Truth

Mentor: Jerry Liao
Member: Shih-Hsuan Lin, Aaron Kao, Johnson Chang, Irene Chen

Outline

Background

- Adversarial Attack
- Adversarial Defense

Methodologies and Results

- Fast Gradient Sign Method (FGSM)
- DeepFool
- Adversarial Perturbation Elimination with GAN (APE-GAN)
- Diffusion model

Summary & Future Works

What Is Adversarial Attack?

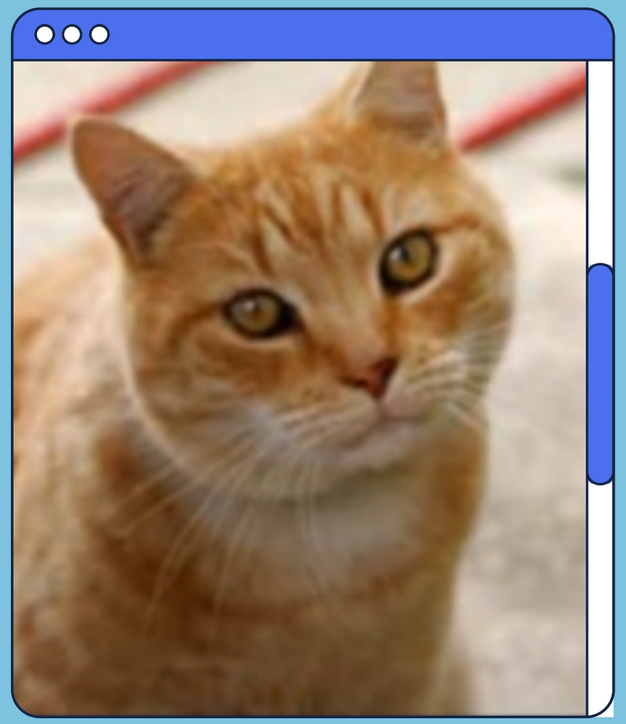
Original Image

Tiger Cat. 0.64

Reference: Hung-Yi Lee, Machine Learning Slides

What Is Adversarial Attack?

Original Image



Tiger Cat

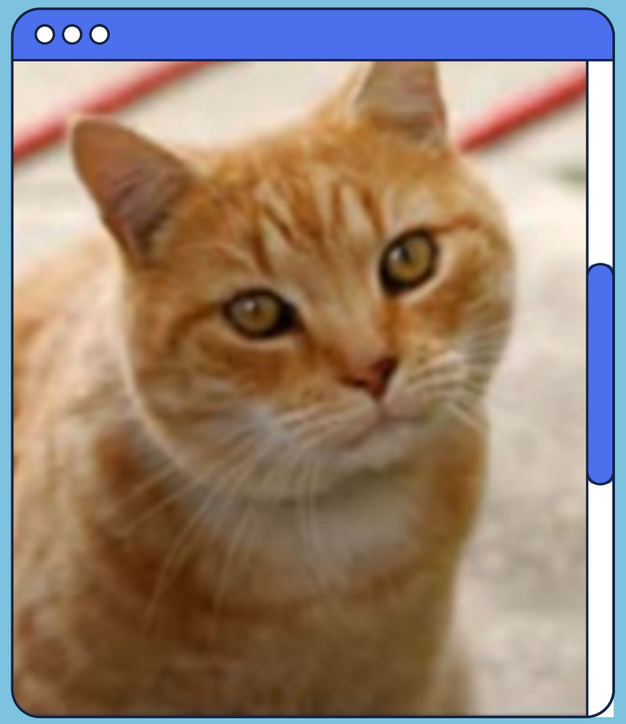
0.64

+

Noise

=

Adversarial Image



Star Fish

1.00

Reference: Hung-Yi Lee, Machine Learning Slides

How to Defend?

Generative Model

- Adversarial Perturbation Elimination with GAN (APE-GAN)
- Diffusion Model

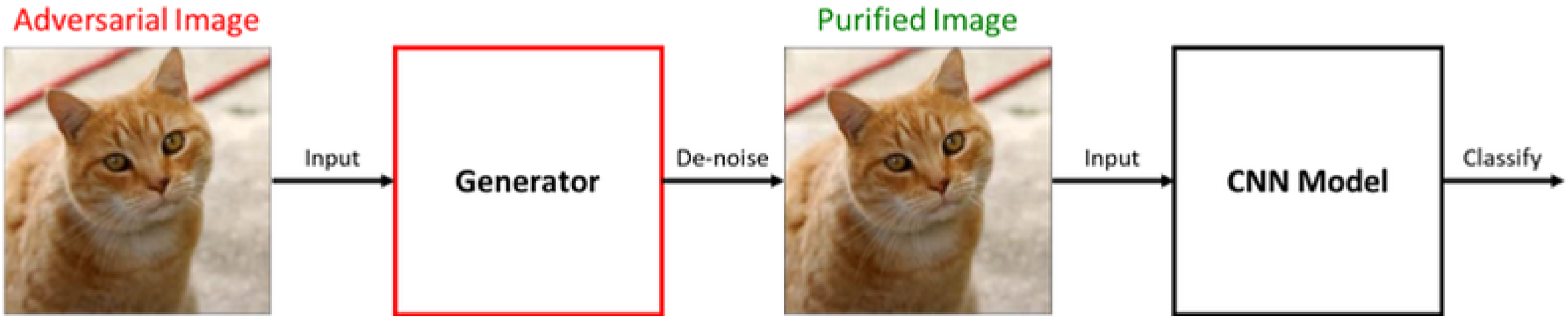
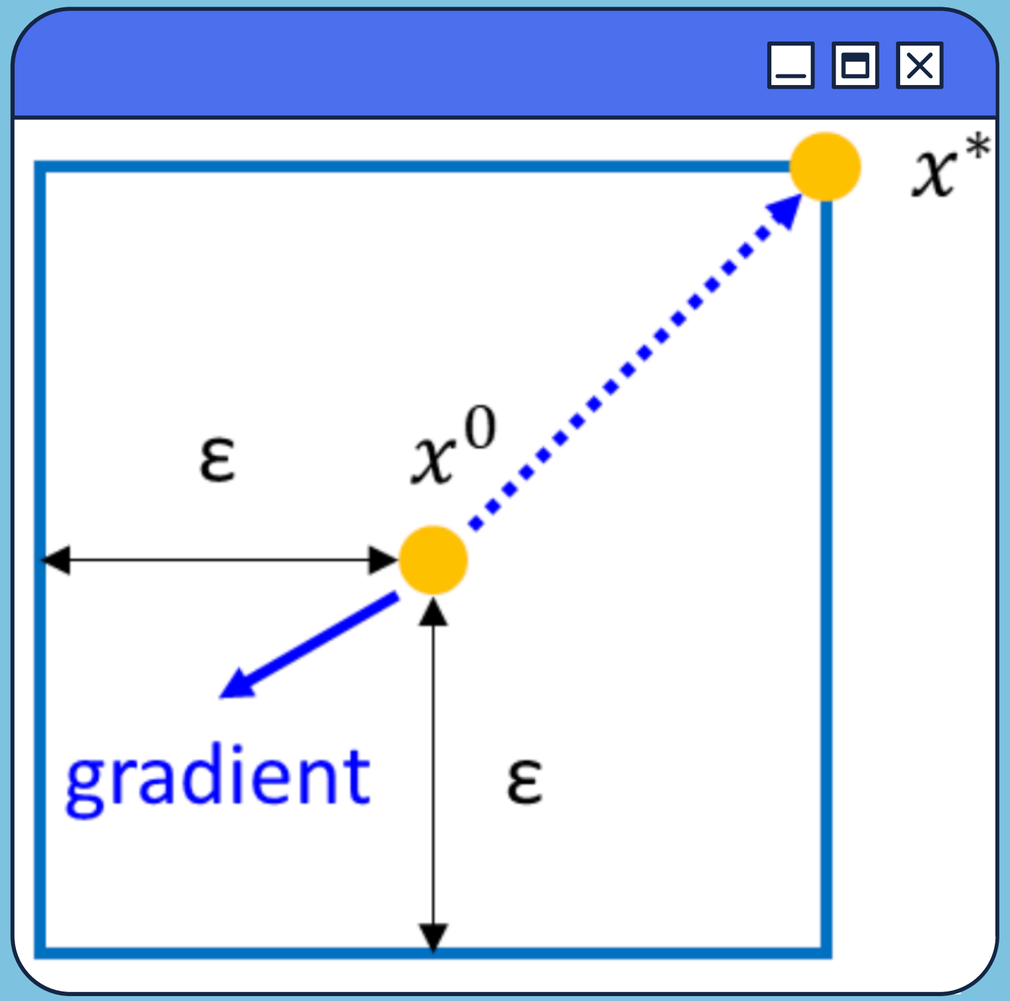


Image Reference: Hung-Yi Lee, Machine Learning Slides

Fast Gradient Sign Method (FGSM)

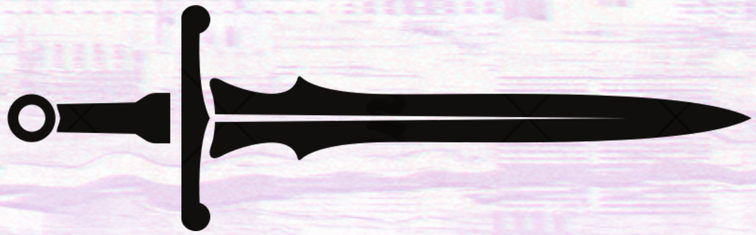
Fast Gradient Sign Method (FGSM)



Reference: Hung-Yi Lee, Machine Learning Slides

PLAY

DeepFool

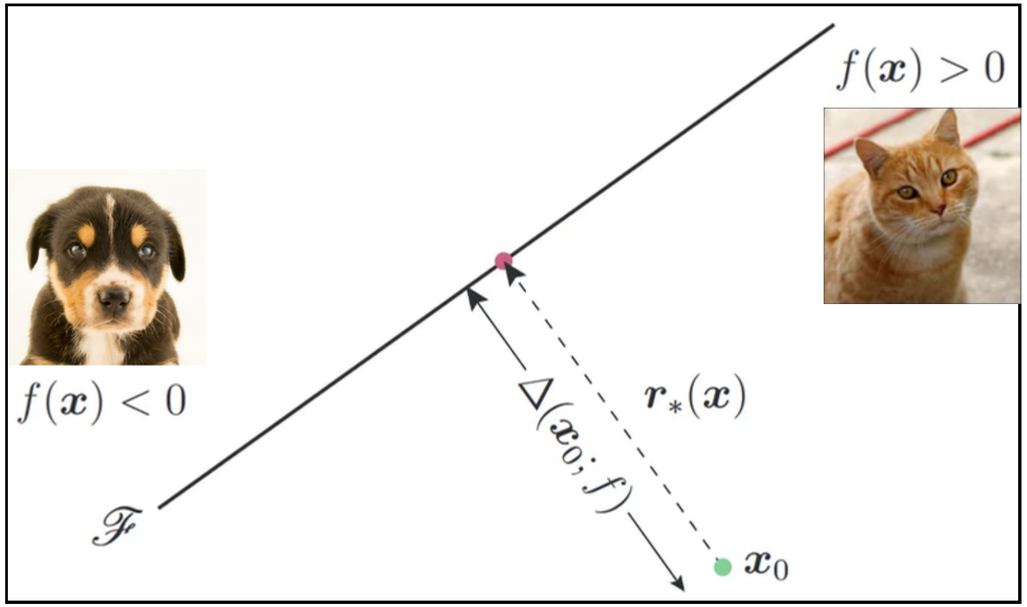


PLAY

4-24-22
12:36:10 THURS 4

DeepFool

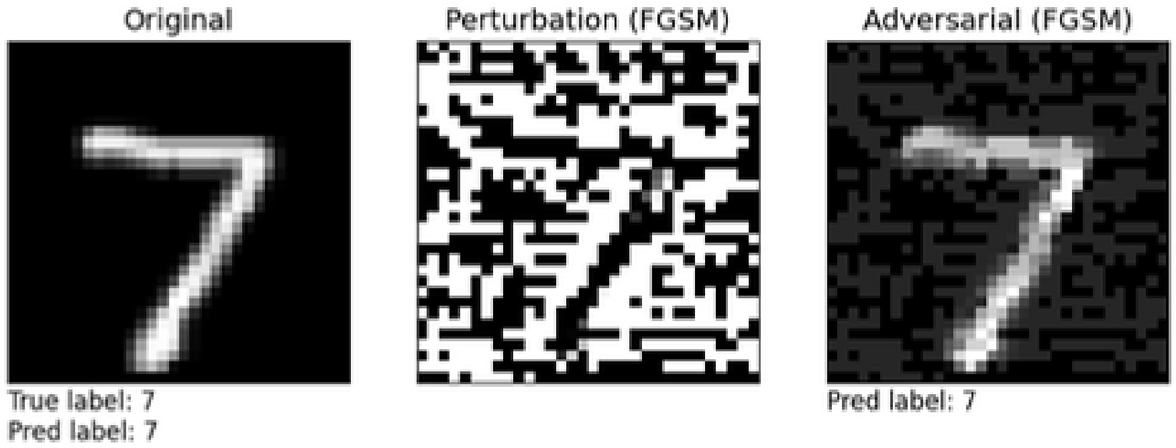
- Use Binary Classifier as an example
- Obtain minimal perturbation (distance) → Iterate until label change

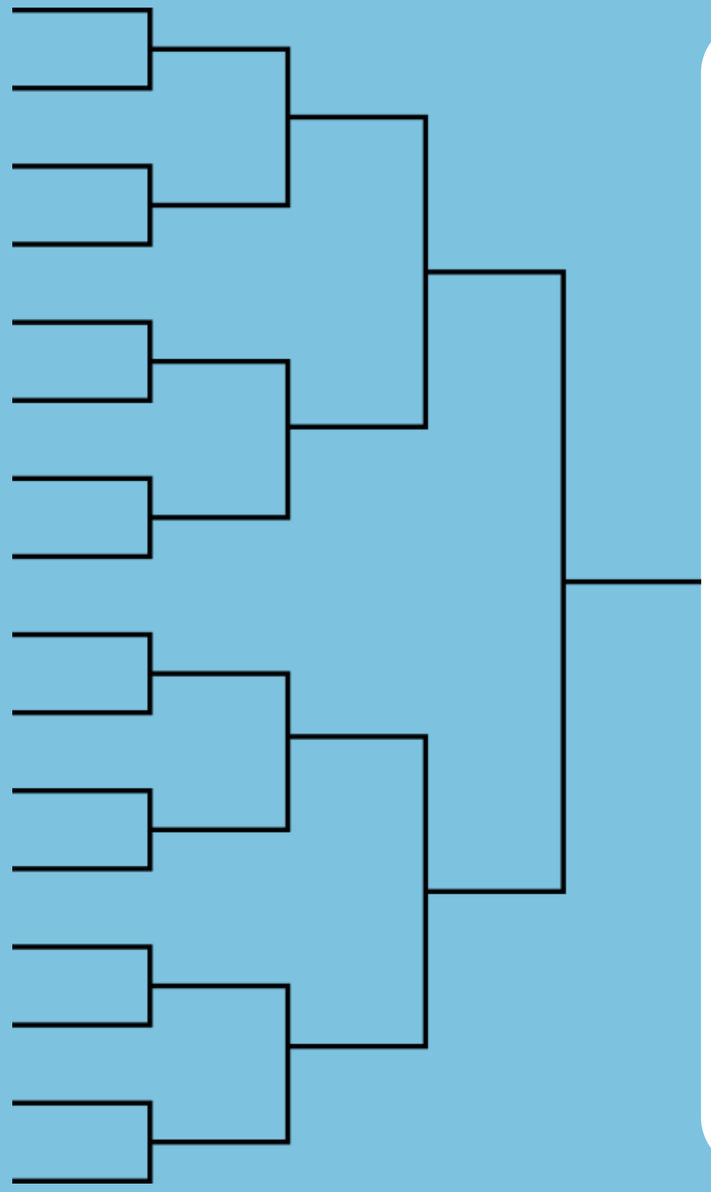


Reference: [Paper] DeepFool: a simple and accurate method to fool deep neural networks

FGSM vs DeepFool

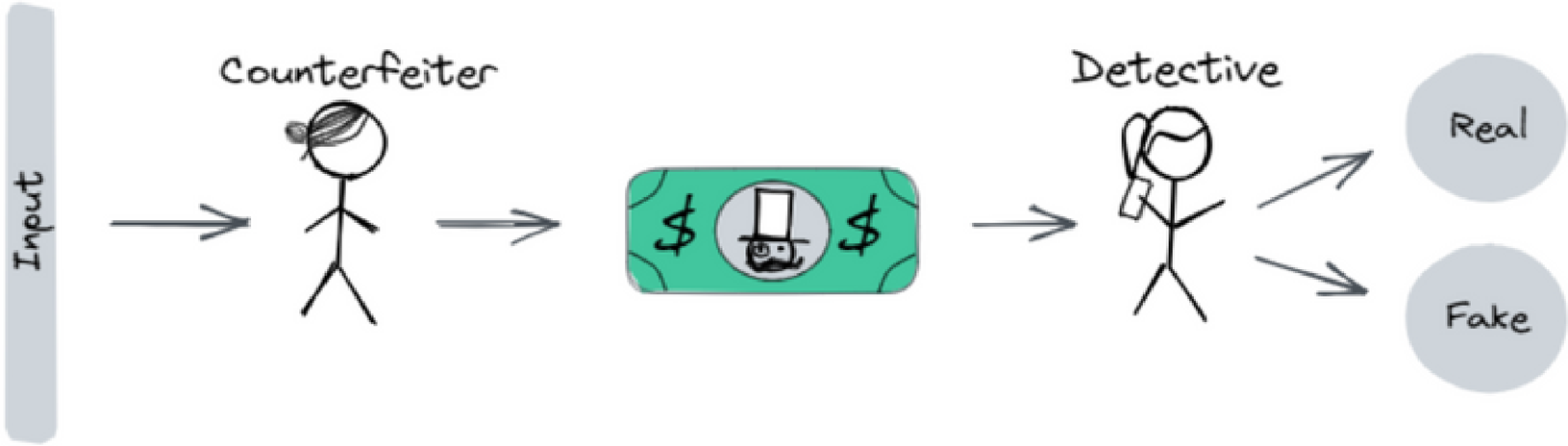
- **DeepFool add less perturbation and more misclassification**





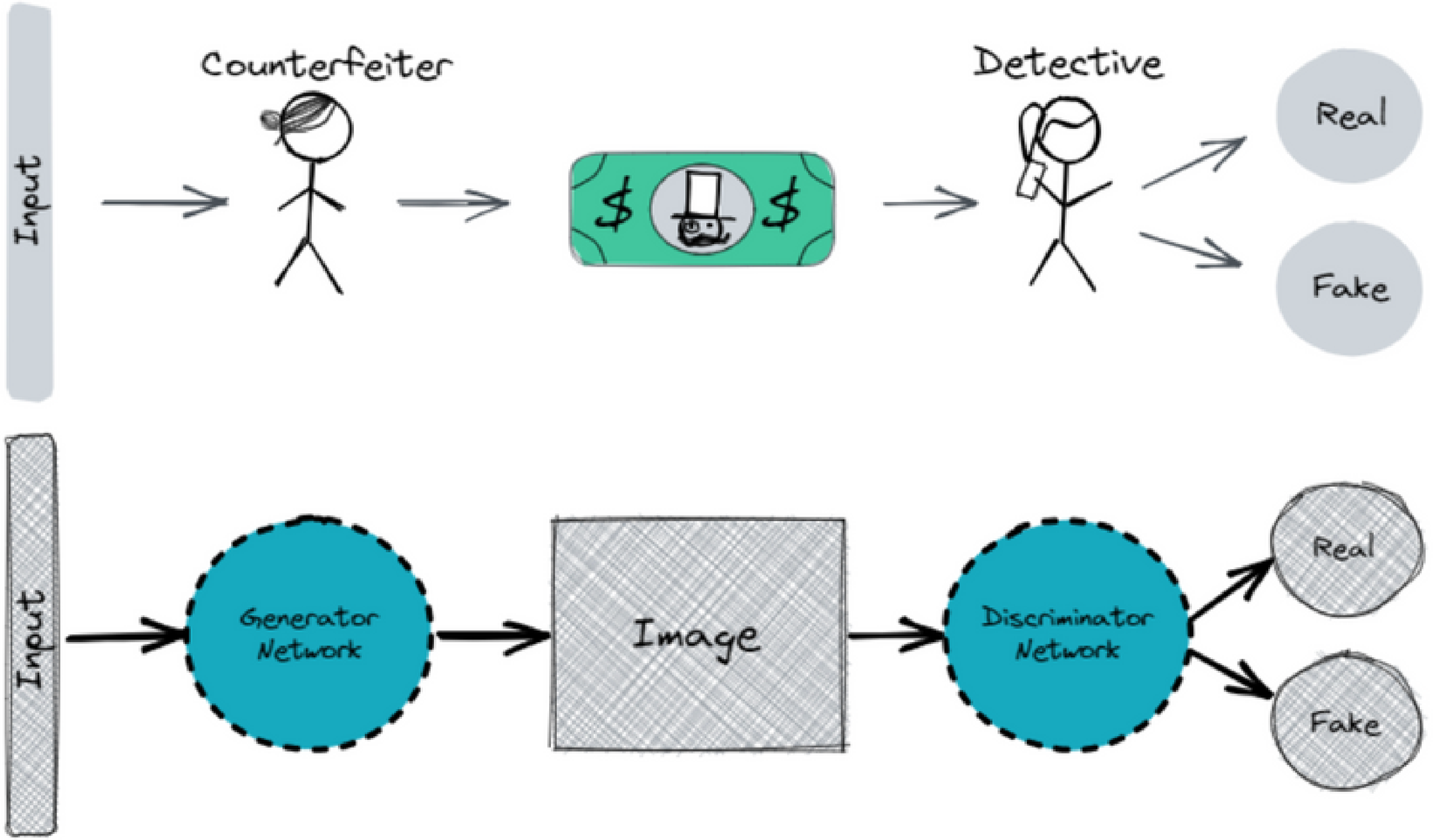
APE-GAN

Generative Adversarial Network (GAN)



Reference: [DeepLizard] Generative Adversarial Networks - GANs Intro

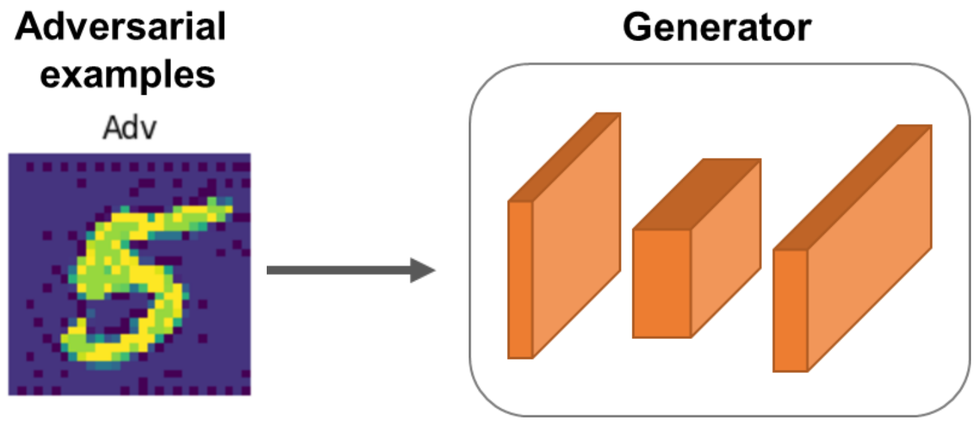
Generative Adversarial Network (GAN)



Reference: [DeepLizard] Generative Adversarial Networks - GANs Intro

APE-GAN

- **Build Generator**



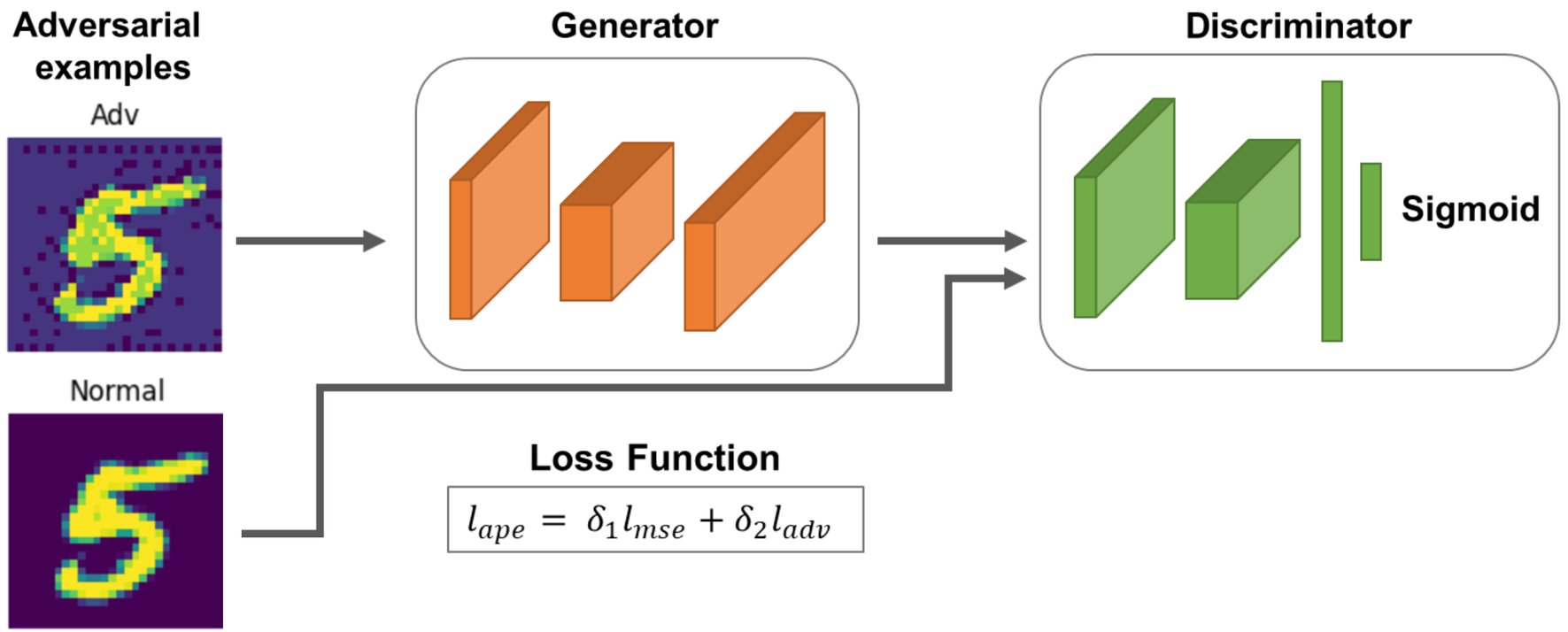
Loss Function

$$l_{ape} = \delta_1 l_{mse} + \delta_2 l_{adv}$$

Reference: [Paper] Ape-gan: Adversarial perturbation elimination with gan

APE-GAN

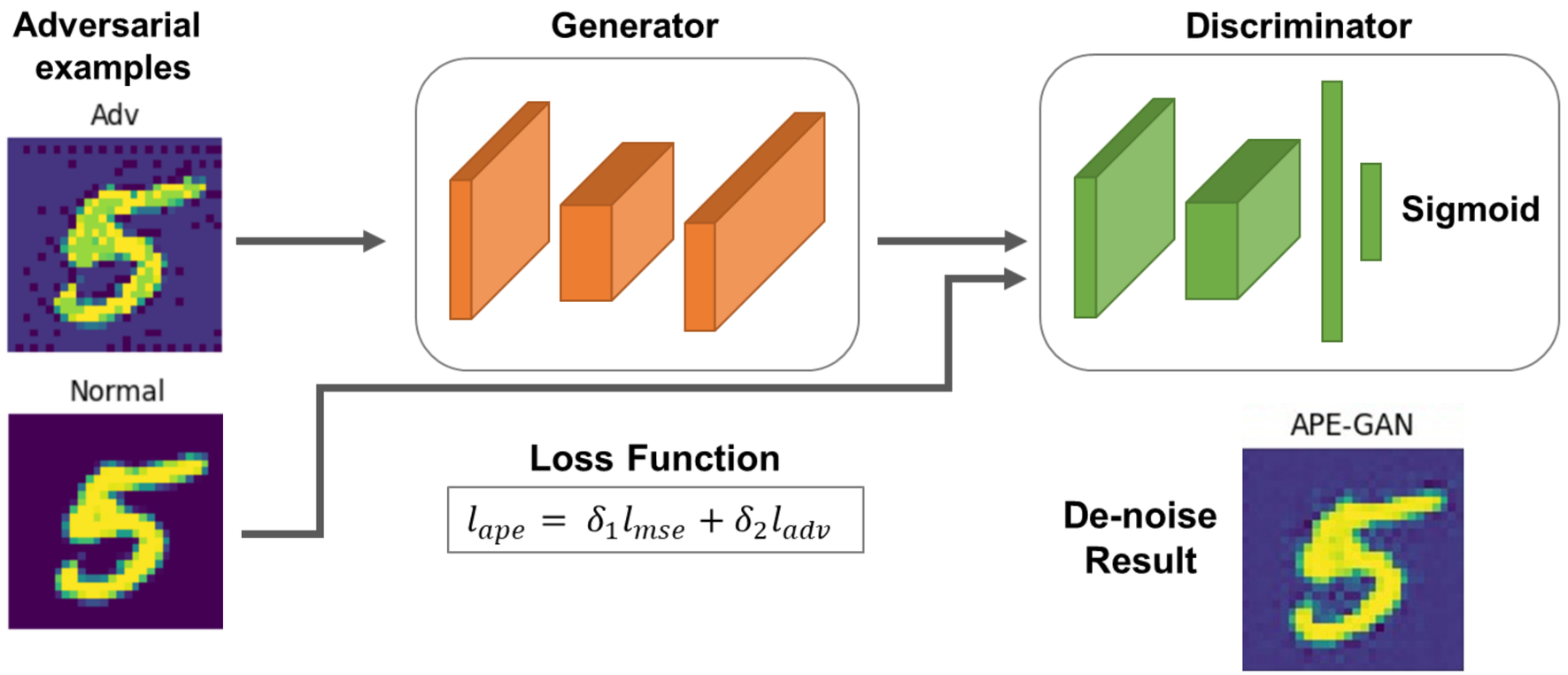
- **Build Discriminator**



Reference: [Paper] Ape-gan: Adversarial perturbation elimination with gan

APE-GAN

- Train generator twice + discriminator one time in each epoch



Reference: [Paper] Ape-gan: Adversarial perturbation elimination with gan

Diffusion Model
&
U-Net

Diffusion Model

Forward Diffusion Process

Data **Noise**

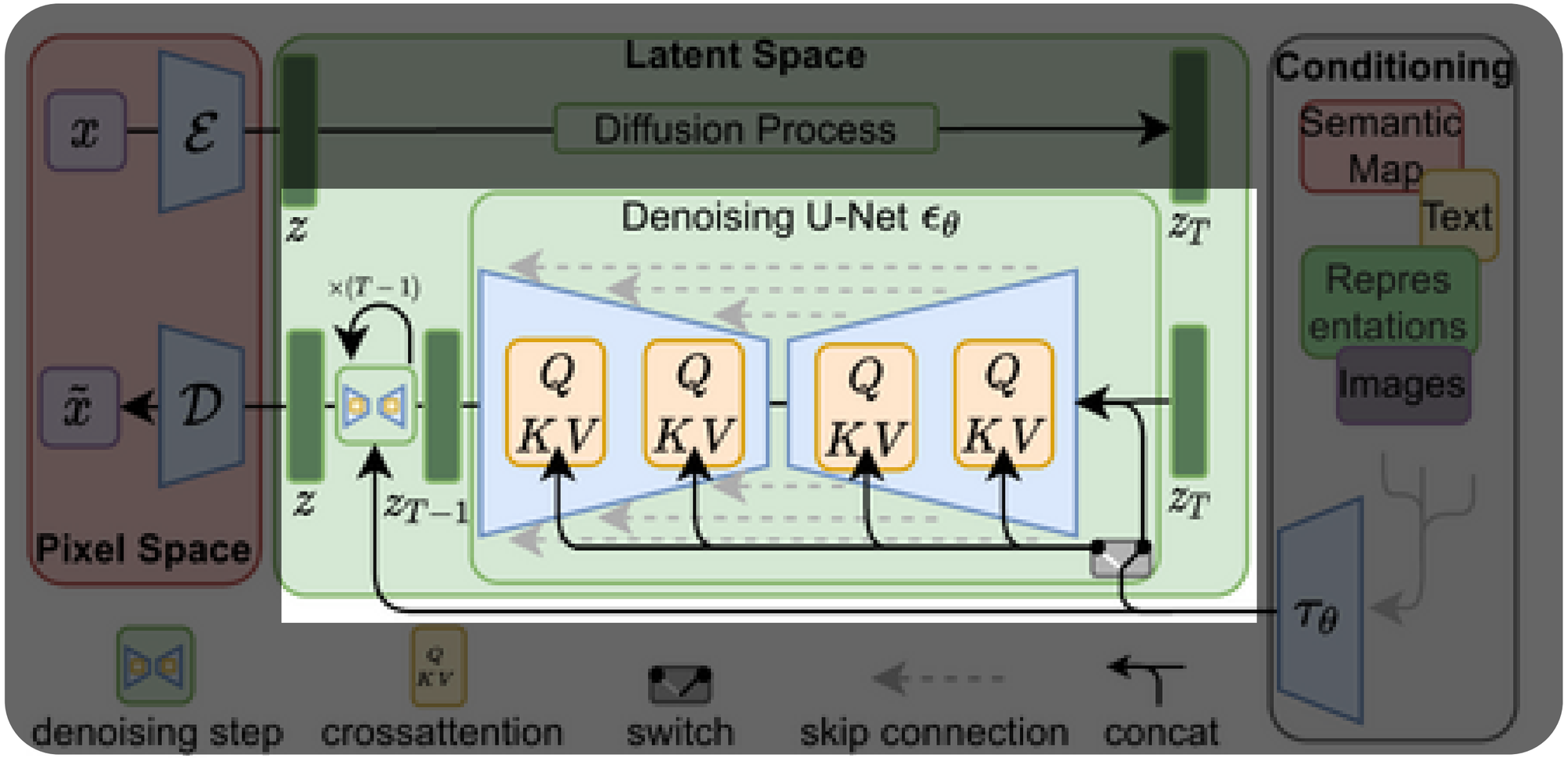
Diffusion Model

Reverse Denoise Process(Generative)

Data

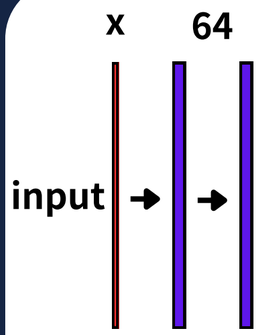
Noise

Diffusion Architecture



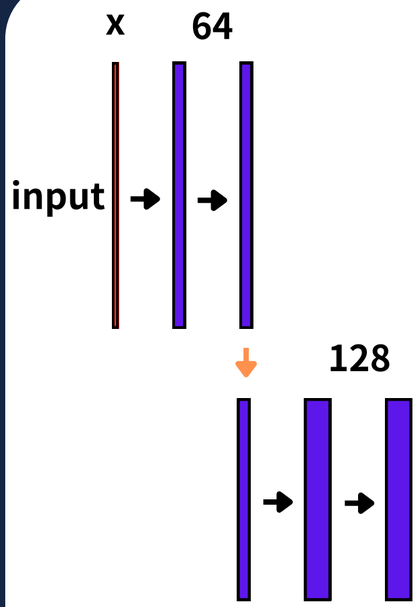
Reference: https://arxiv.org/abs/2112.10752

U-Net Architecture



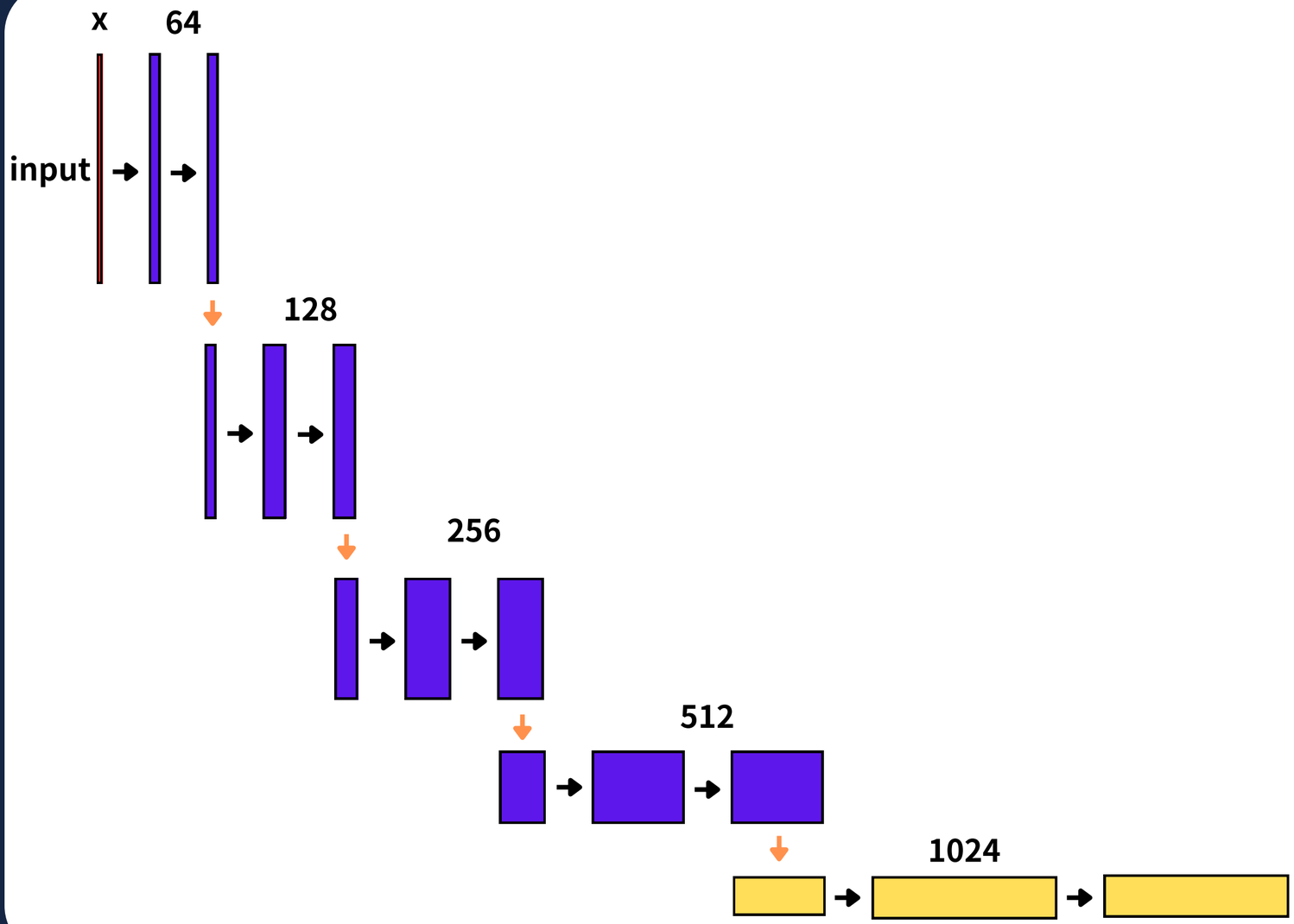
→ Conv, ReLU
↓ MaxPooL
↑ UpSample
→ Concatenate
→ FinalConv

U-Net Architecture



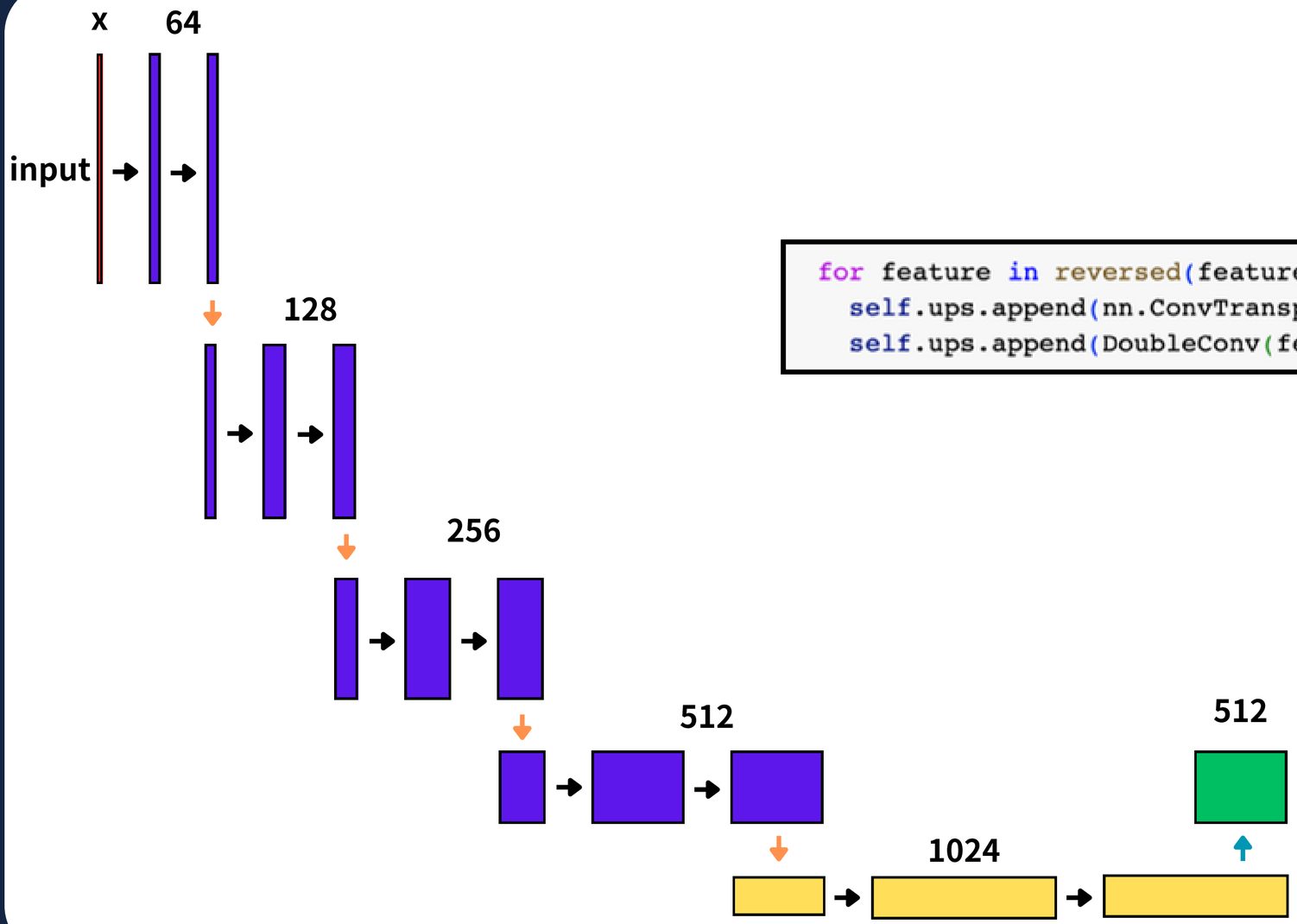
Conv, ReLU
MaxPooL
UpSample
Concatenate
FinalConv

U-Net Architecture



Conv, ReLU
MaxPooL
UpSample
Concatenate
FinalConv

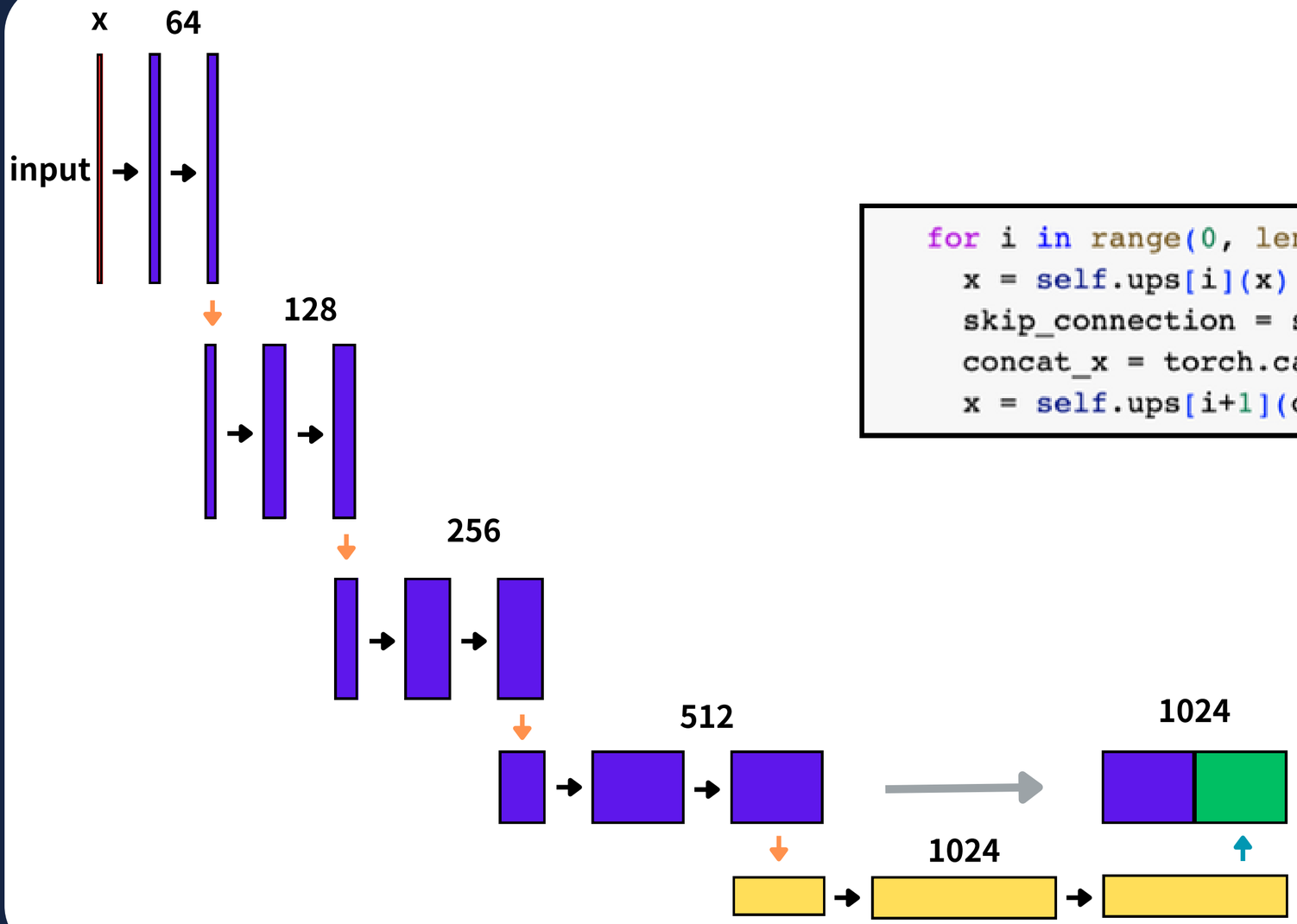
U-Net Architecture




```
for feature in reversed(features):
  self.ups.append(nn.ConvTranspose2d(feature*2, feature, 2, 2))
  self.ups.append(DoubleConv(feature*2, feature))
```

→ Conv, ReLU
↓ MaxPooL
↑ UpSample
→ Concatenate
→ FinalConv

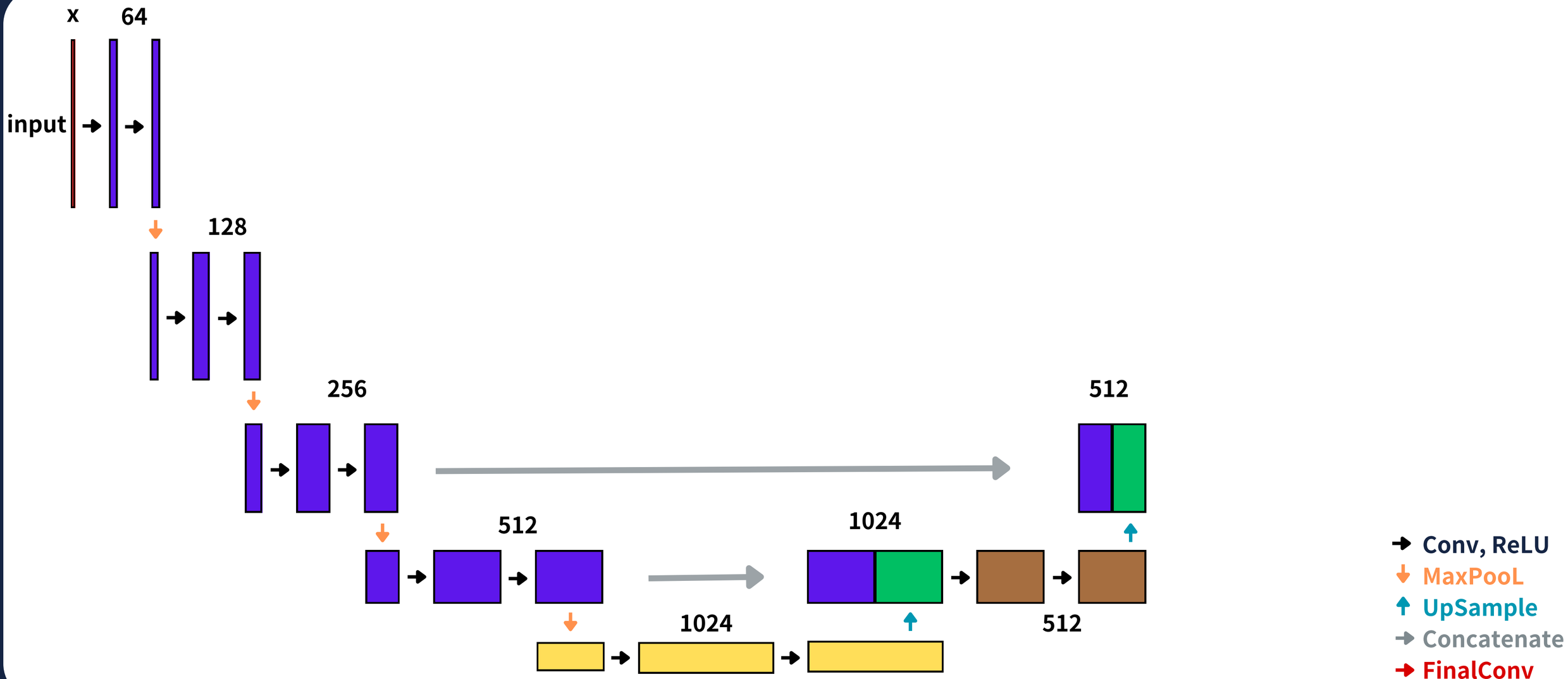
U-Net Architecture



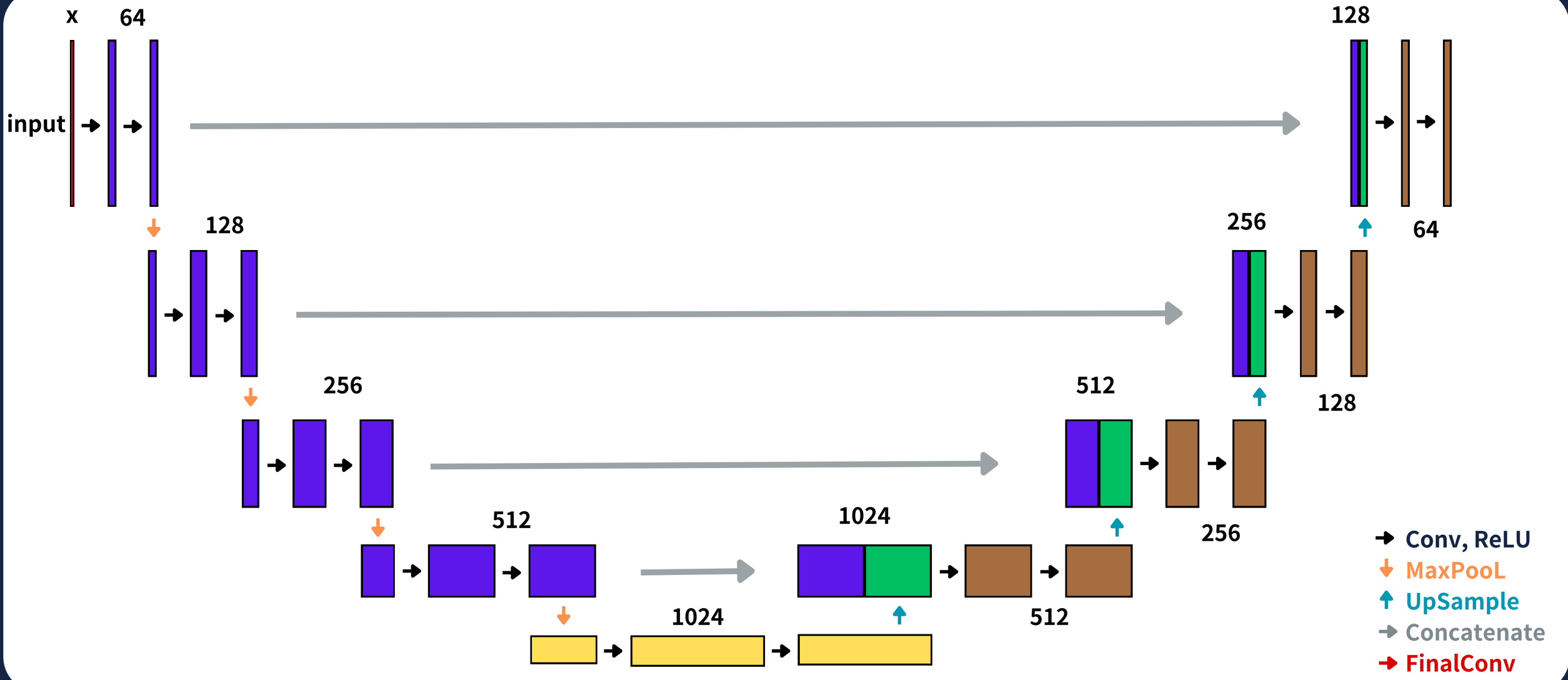

```
for i in range(0, len(self.ups), 2):
  x = self.ups[i](x)
  skip_connection = skip_connections[i//2]
  concat_x = torch.cat((skip_connection, x), dim=1)
  x = self.ups[i+1](concat_x)
```

- Conv, ReLU
- MaxPooL
- UpSample
- Concatenate
- FinalConv

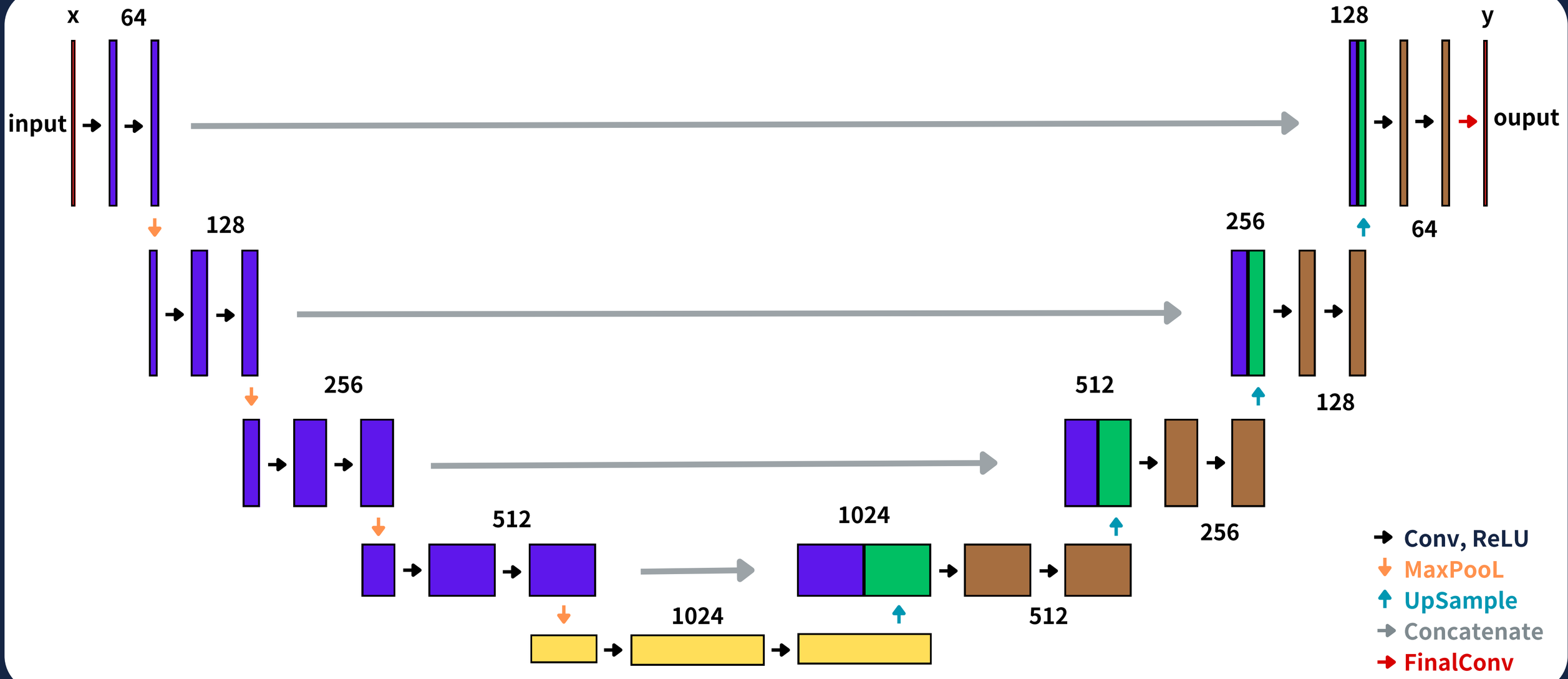
U-Net Architecture



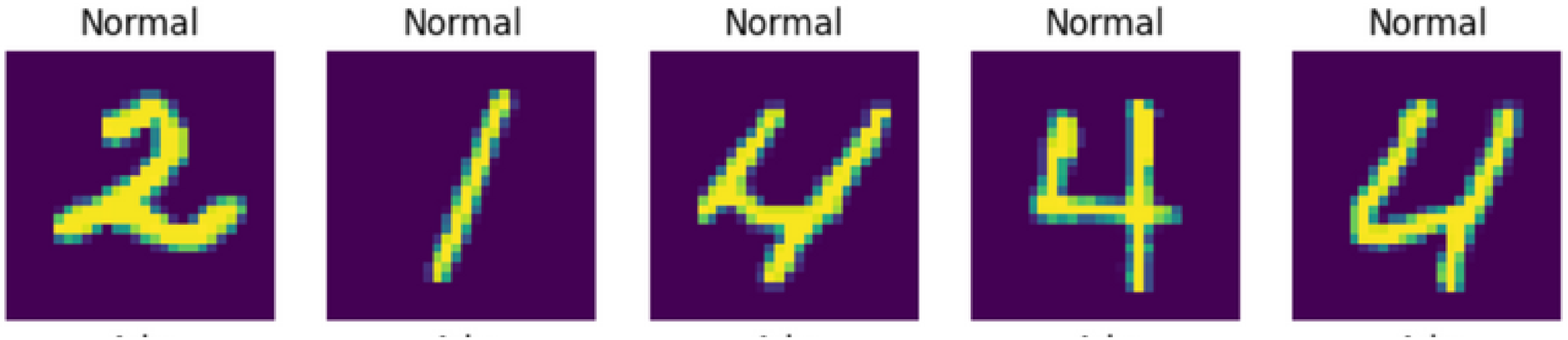
U-Net Architecture

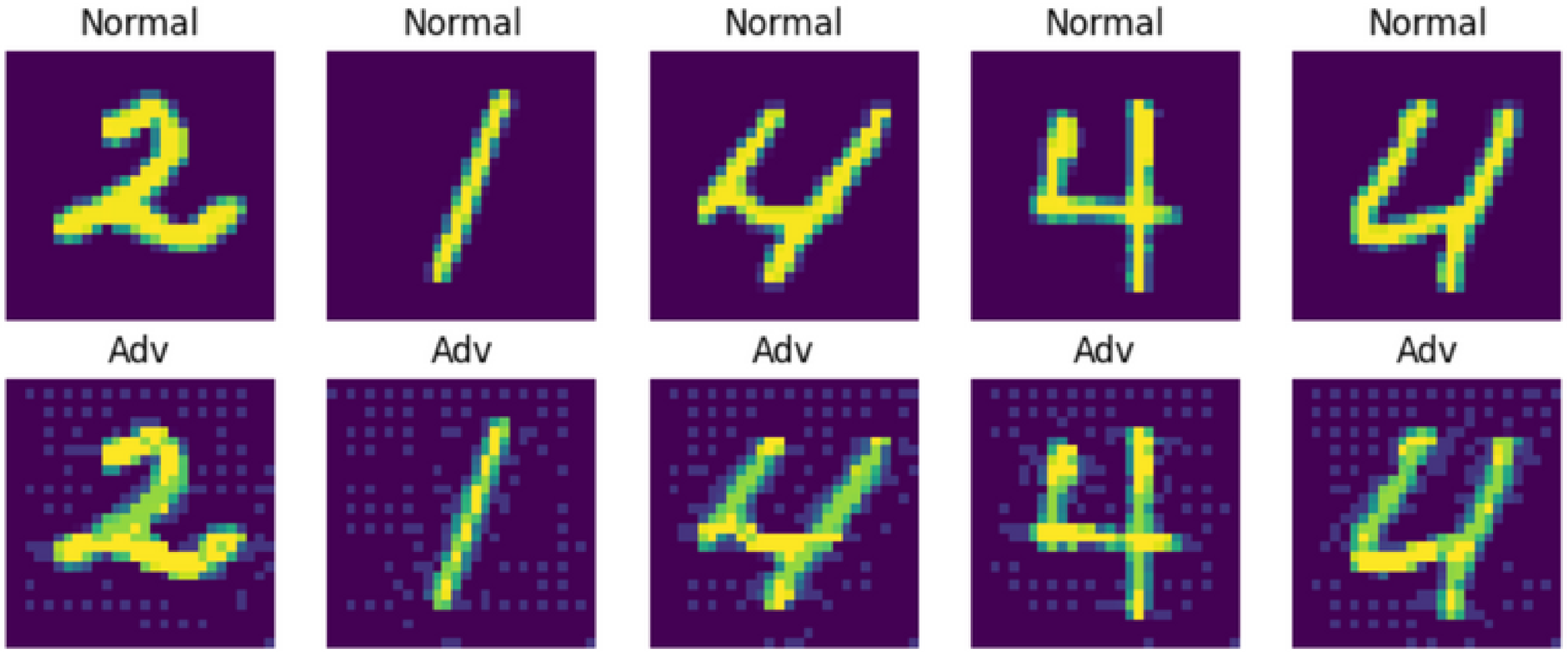


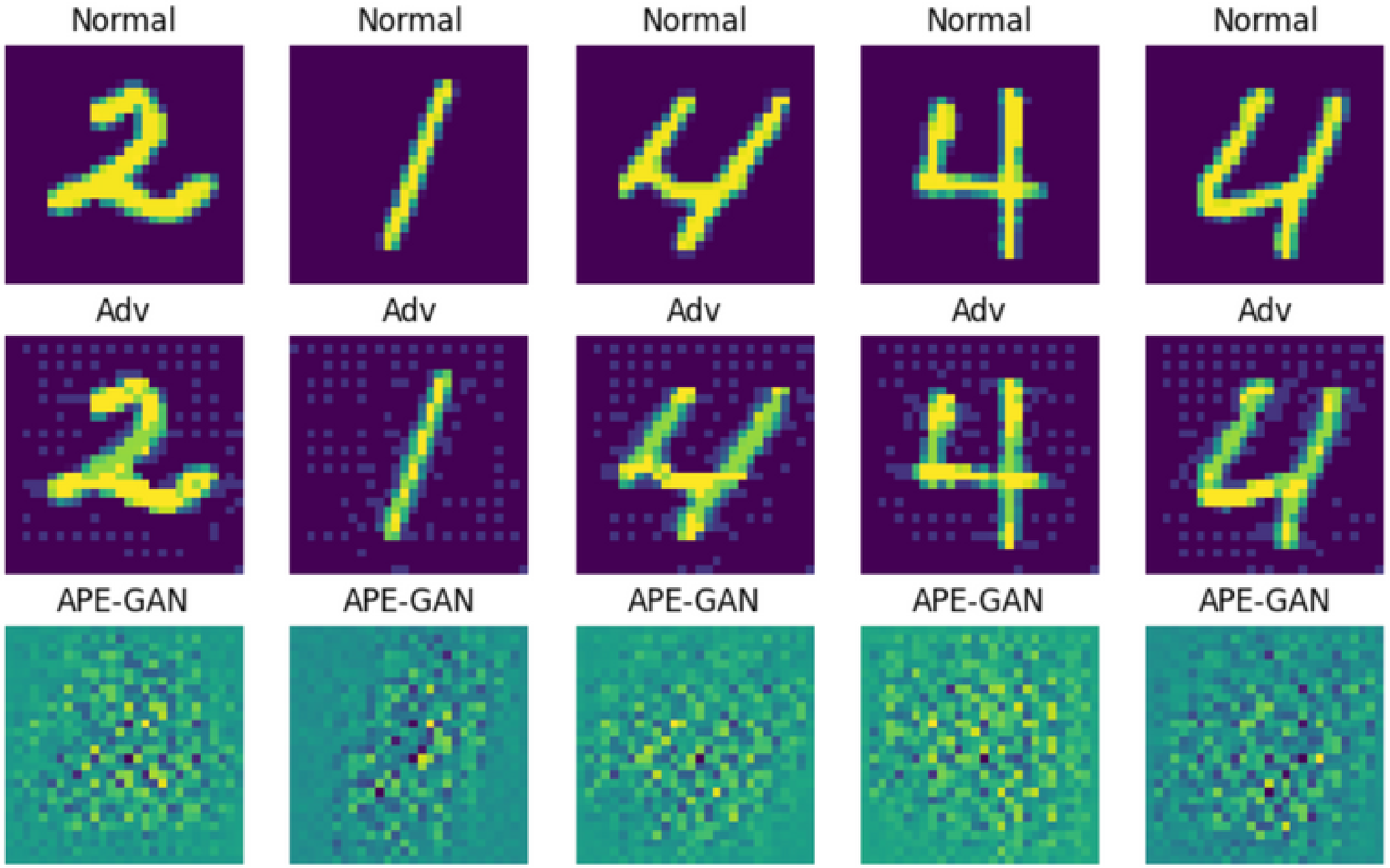
U-Net Architecture

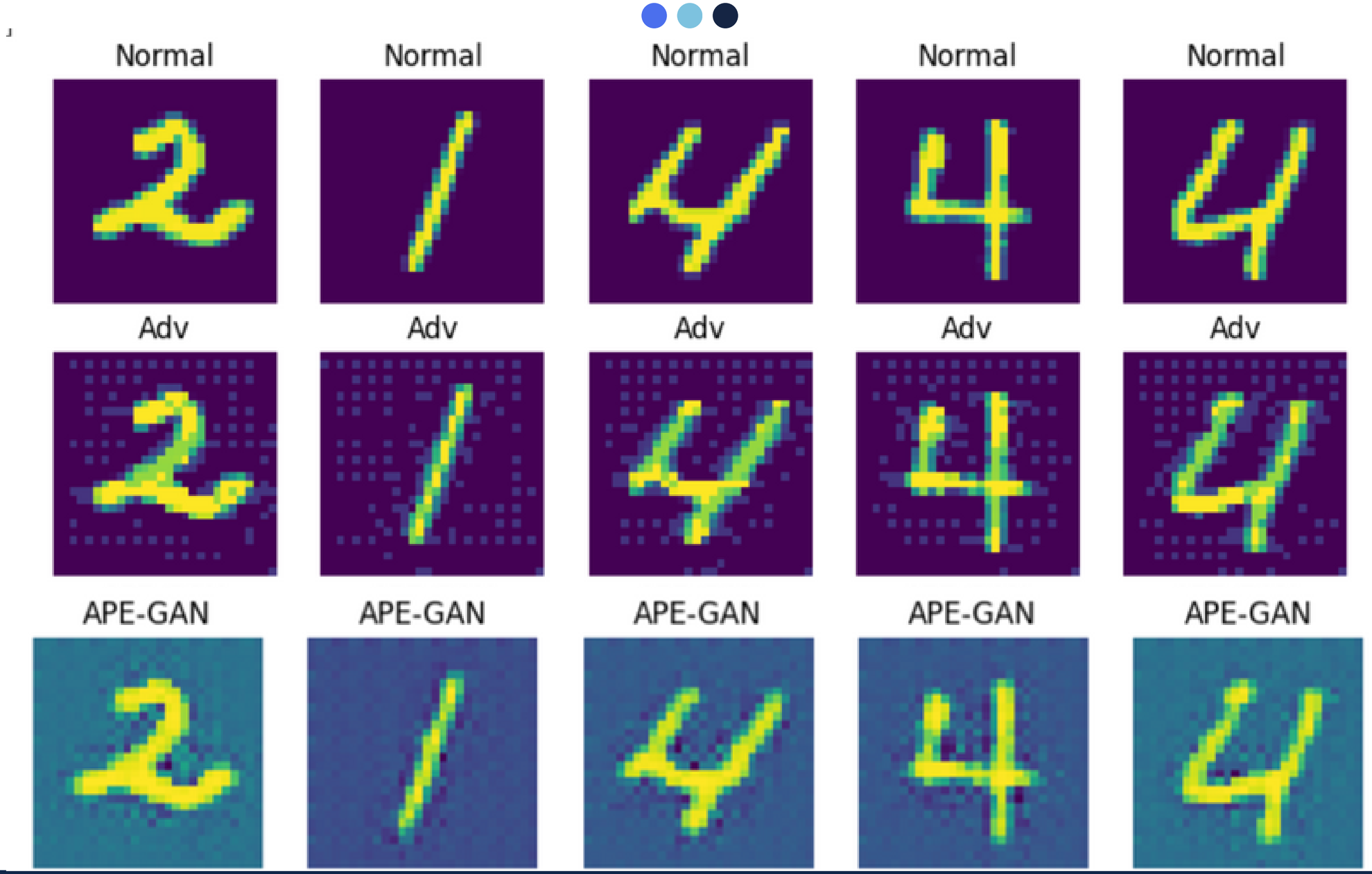


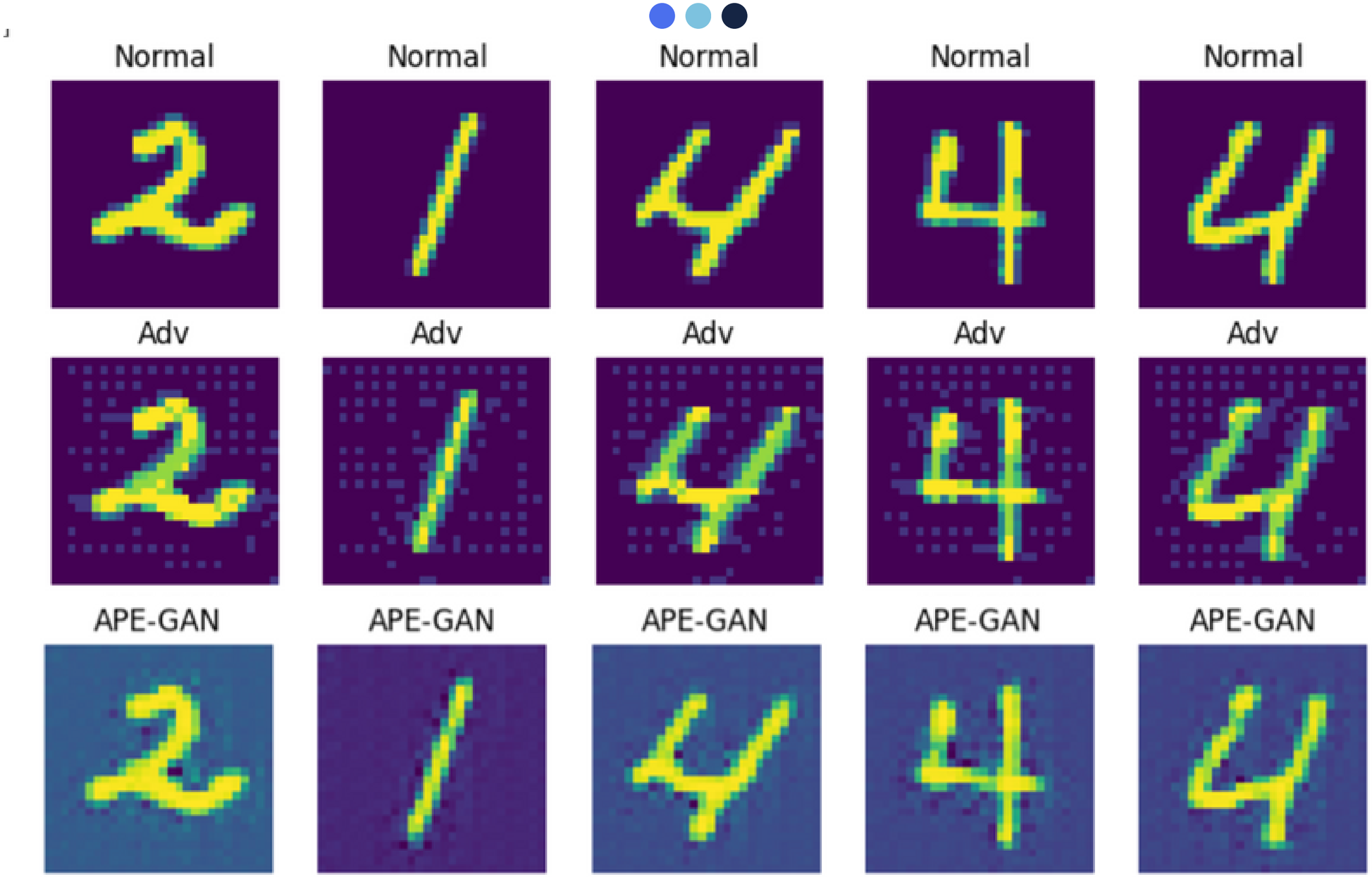
Results of APE-GAN Defense

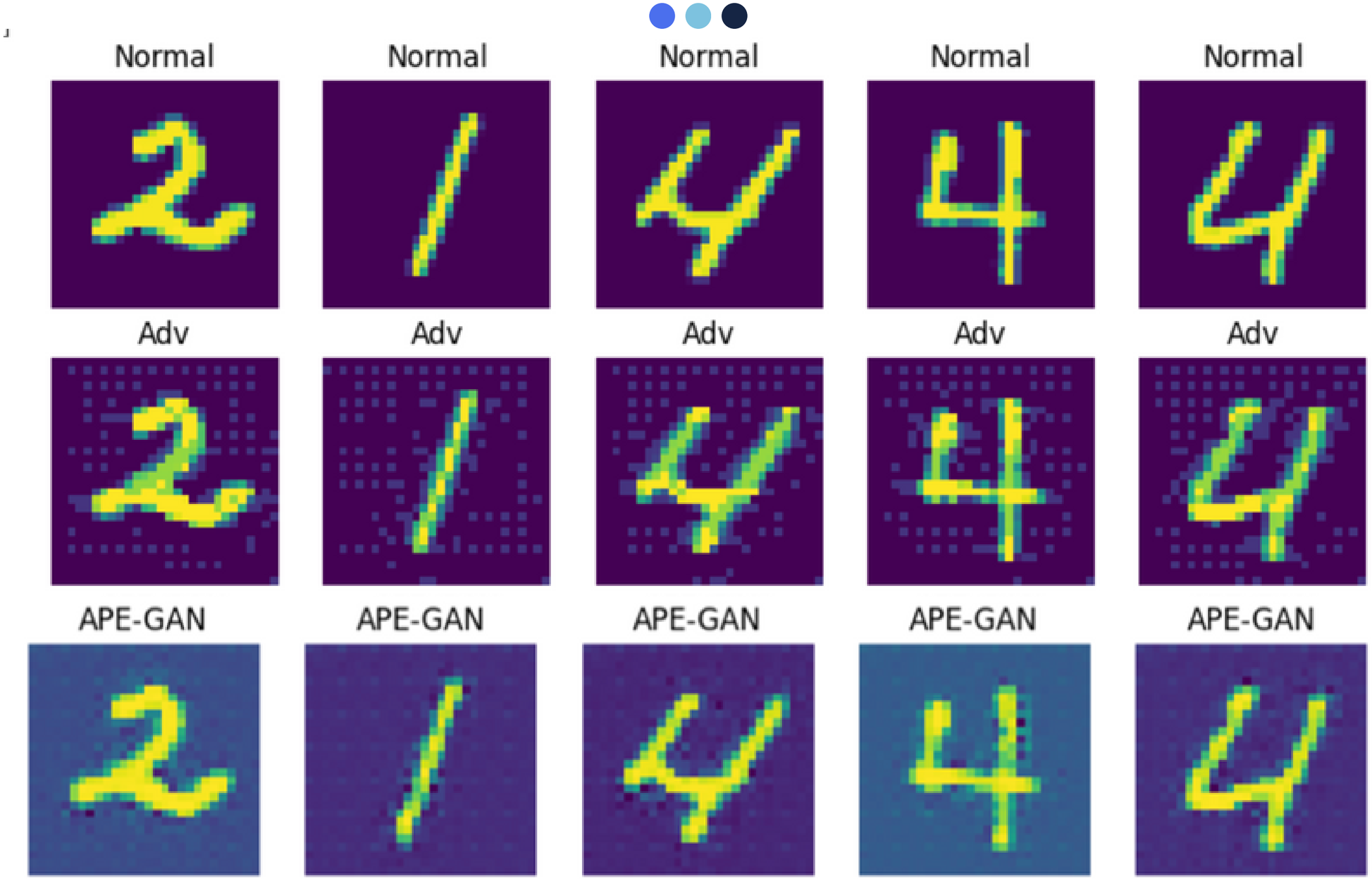

Normal
Normal
Normal
Normal
Normal

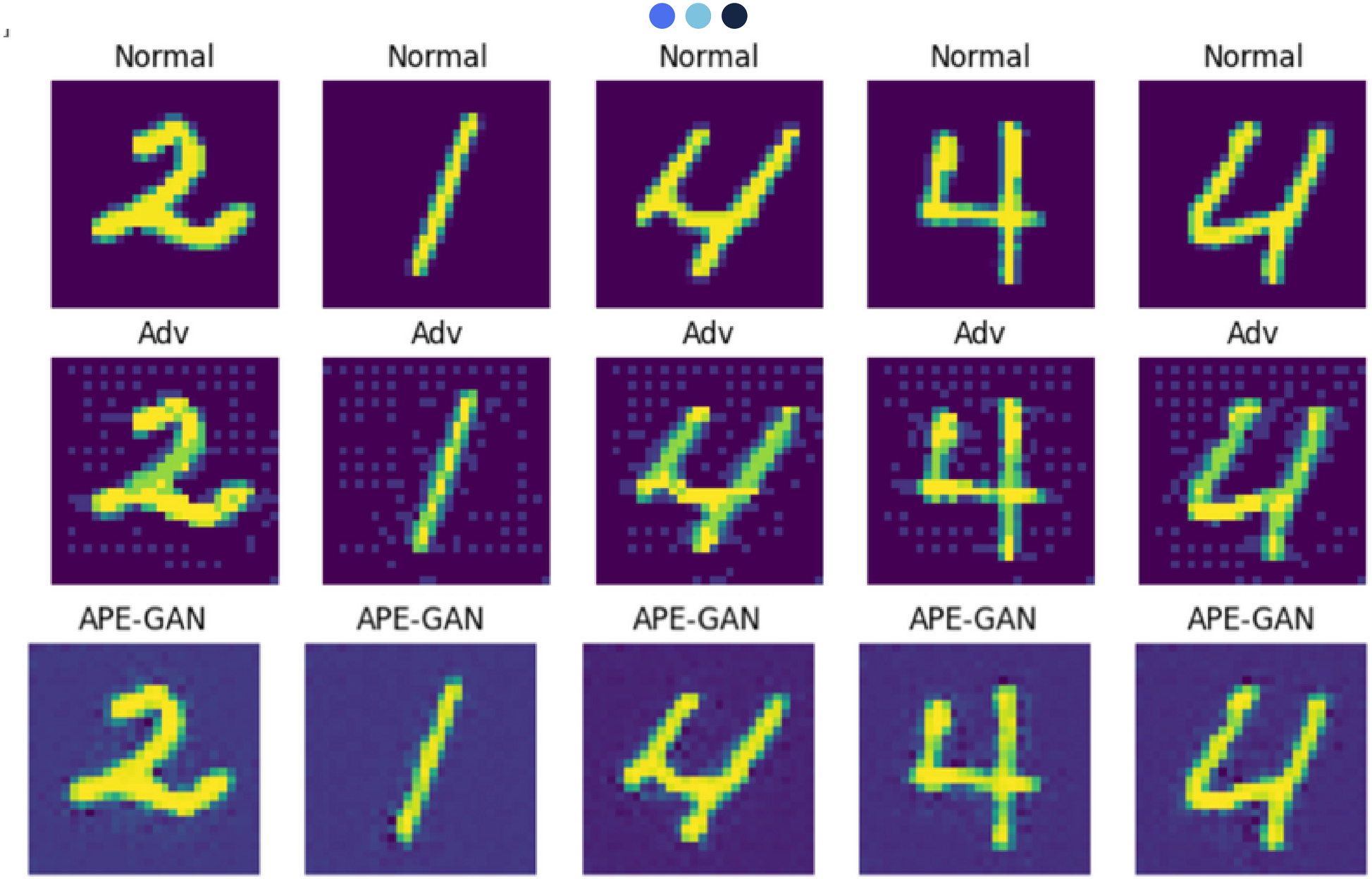

Normal
Normal
Normal
Normal
Normal
Adv
Adv
Adv
Adv
Adv

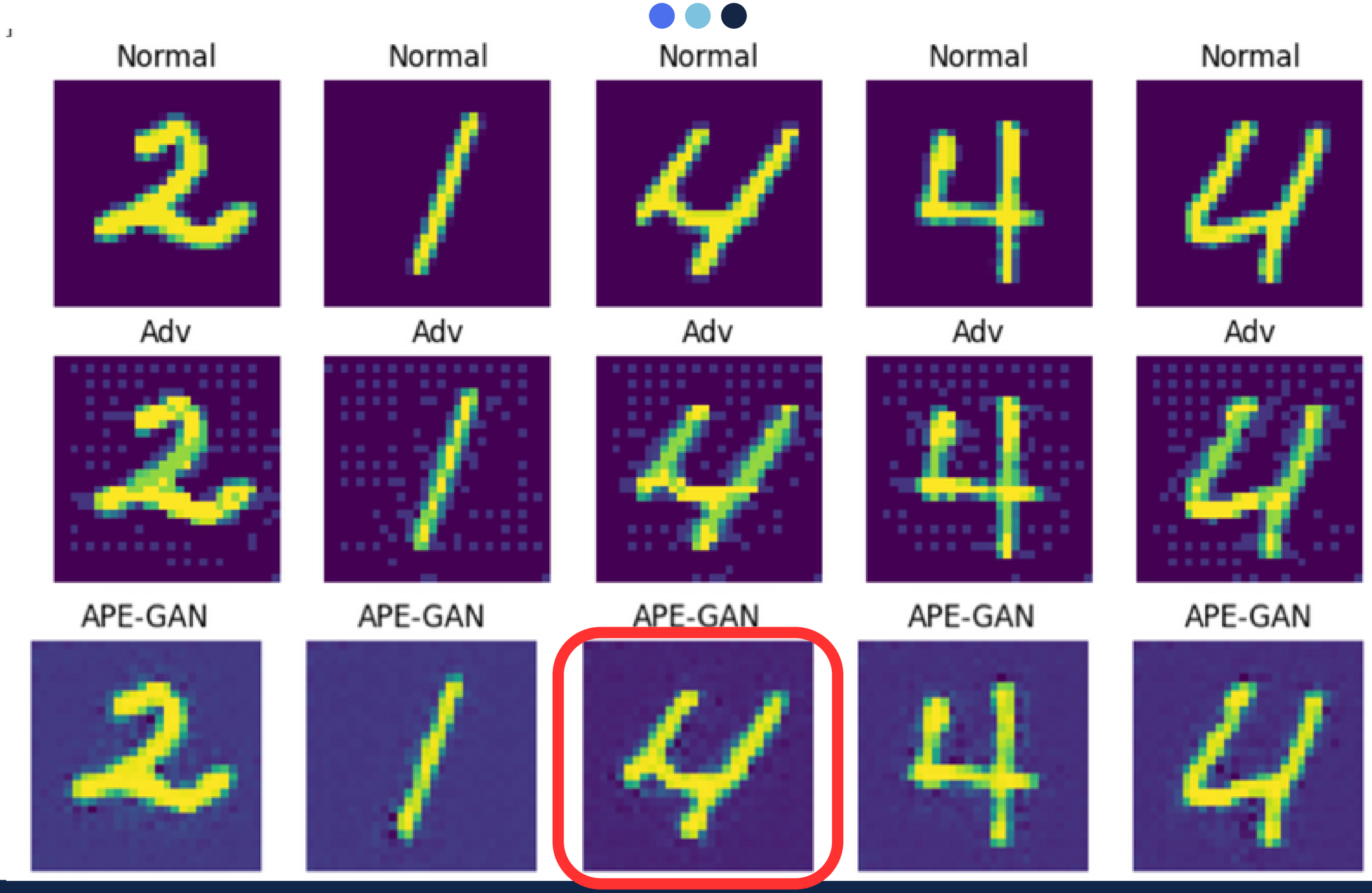

Normal
Normal
Normal
Normal
Normal
Adv
Adv
Adv
Adv
Adv
APE-GAN
APE-GAN
APE-GAN
APE-GAN
APE-GAN


Normal
Normal
Normal
Normal
Normal
Adv
Adv
Adv
Adv
Adv
APE-GAN
APE-GAN
APE-GAN
APE-GAN
APE-GAN


Normal
Normal
Normal
Normal
Normal
Adv
Adv
Adv
Adv
Adv
APE-GAN
APE-GAN
APE-GAN
APE-GAN
APE-GAN


Normal
Normal
Normal
Normal
Normal
Adv
Adv
Adv
Adv
Adv
APE-GAN
APE-GAN
APE-GAN
APE-GAN
APE-GAN

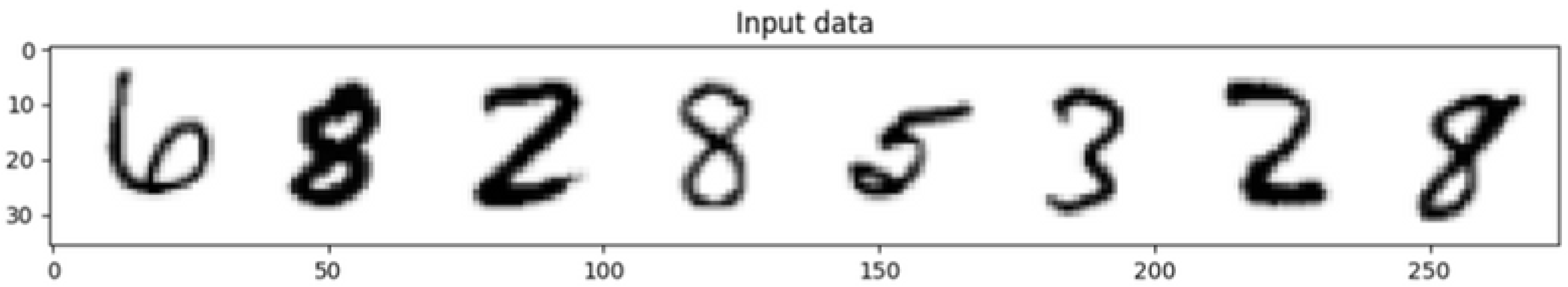

Normal
Normal
Normal
Normal
Normal
Adv
Adv
Adv
Adv
Adv
APE-GAN
APE-GAN
APE-GAN
APE-GAN
APE-GAN


Normal
Normal
Normal
Normal
Normal
Adv
Adv
Adv
Adv
Adv
APE-GAN
APE-GAN
APE-GAN
APE-GAN
APE-GAN

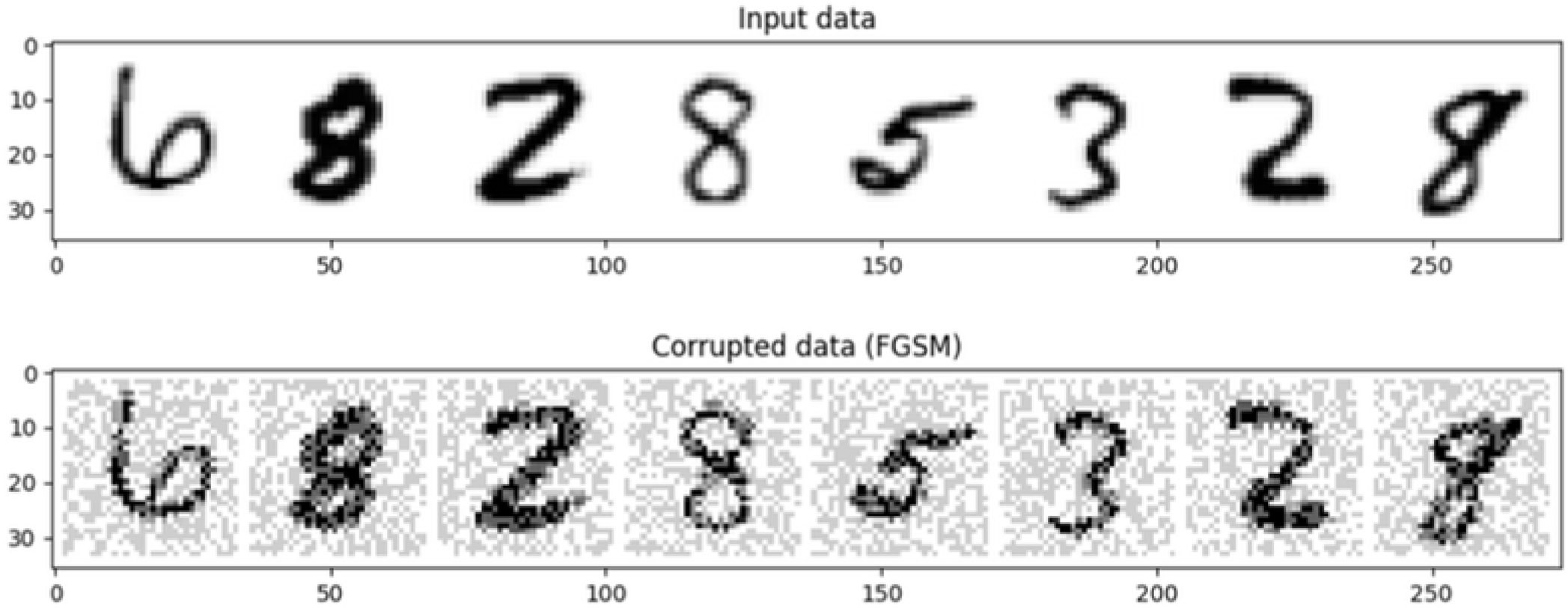
Results of U-Net Defense

FGSM Training U-Net

FGSM Training U-Net

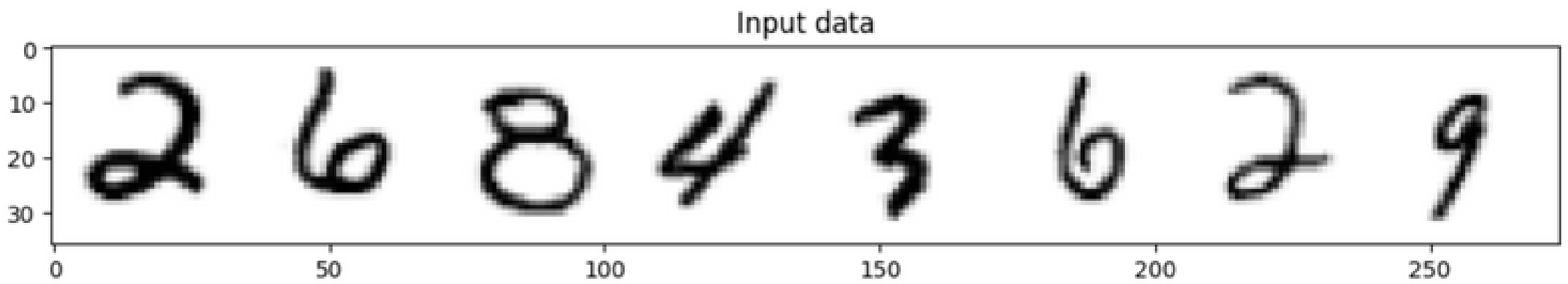


FGSM Training U-Net

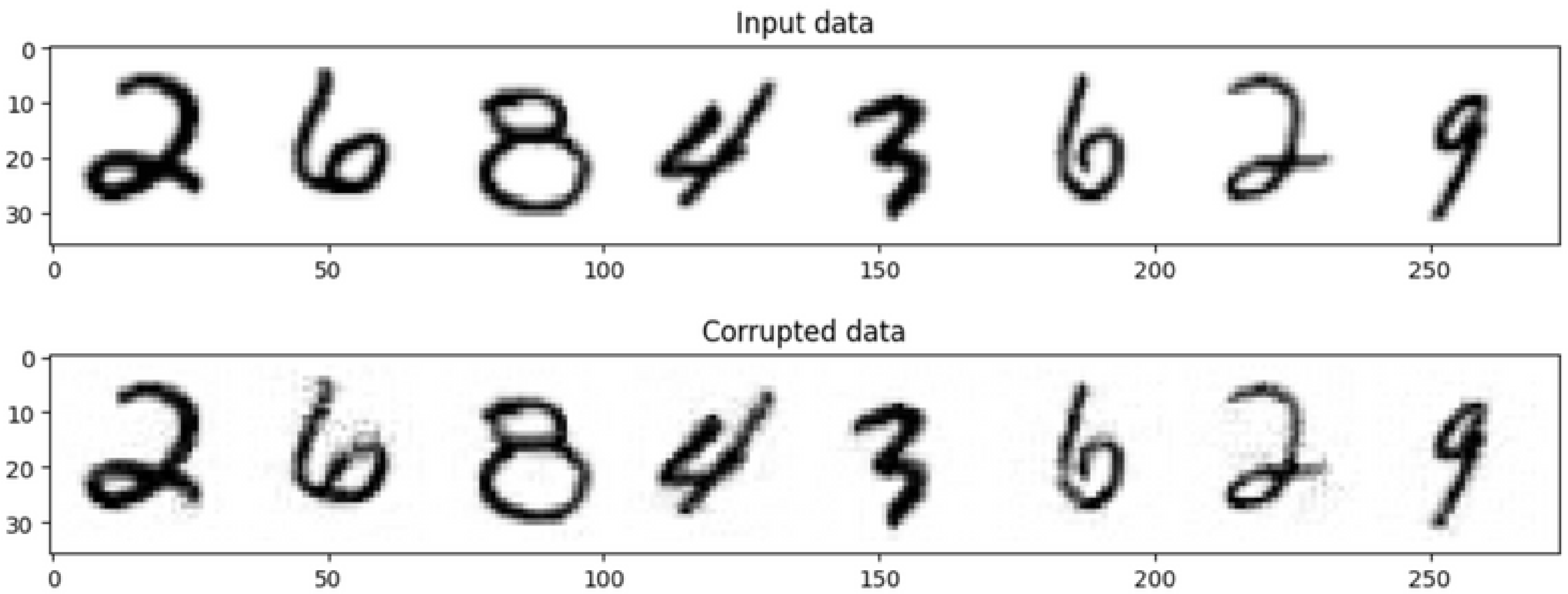


DeepFool Attack

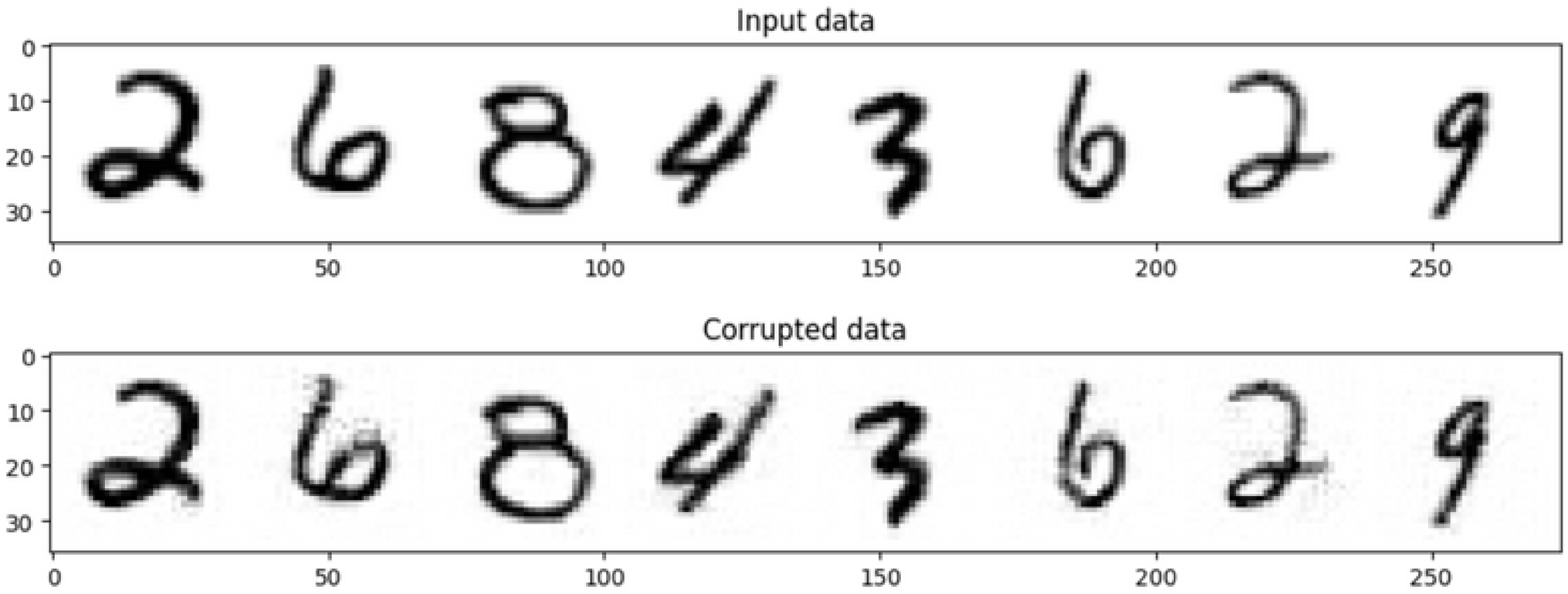
DeepFool Attack



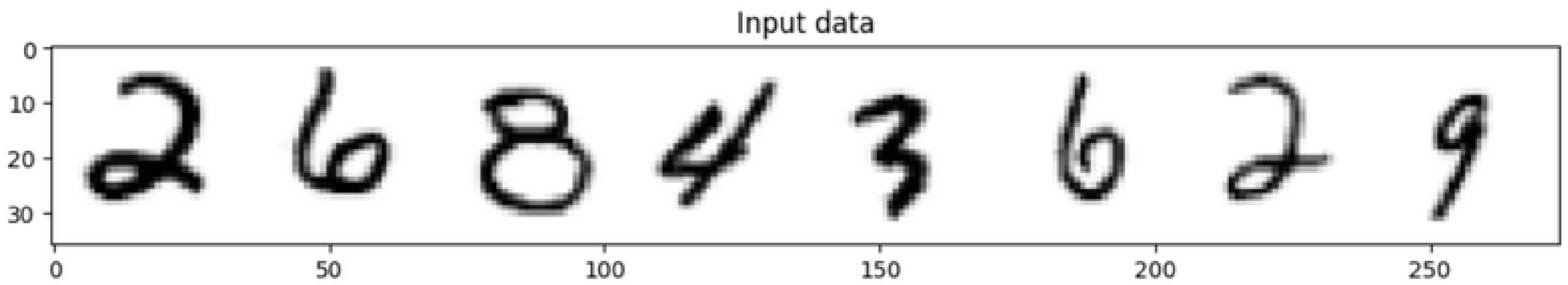
DeepFool Attack

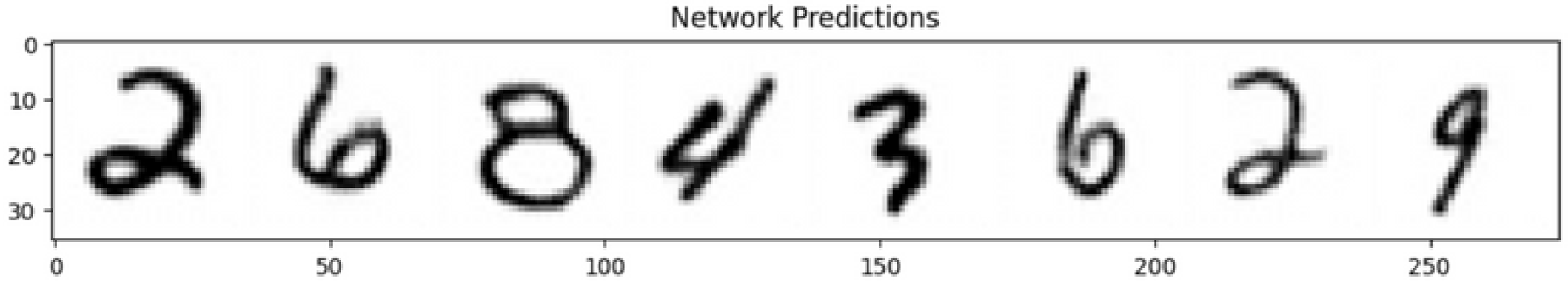


U-Net Prediction



U-Net Prediction



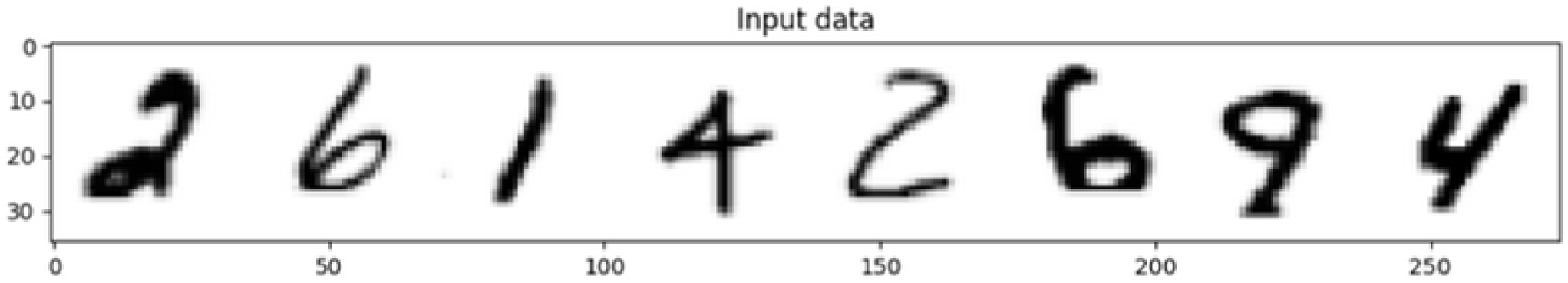


FGSM Training & Deepfool Attack

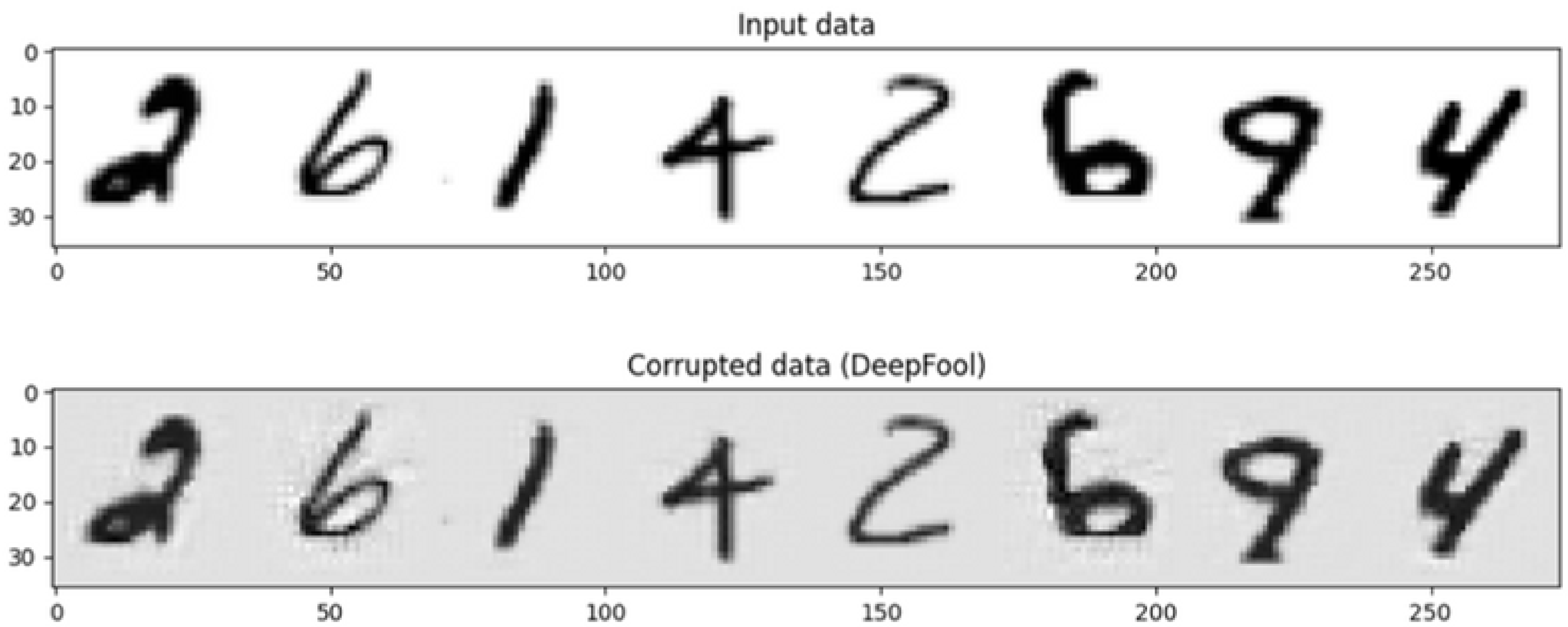
	Original Data	Deepfool Data	U-Net Data
accuracy	97.4%	1.9%	96.8%

DeepFool Training U-net

DeepFool Training U-net

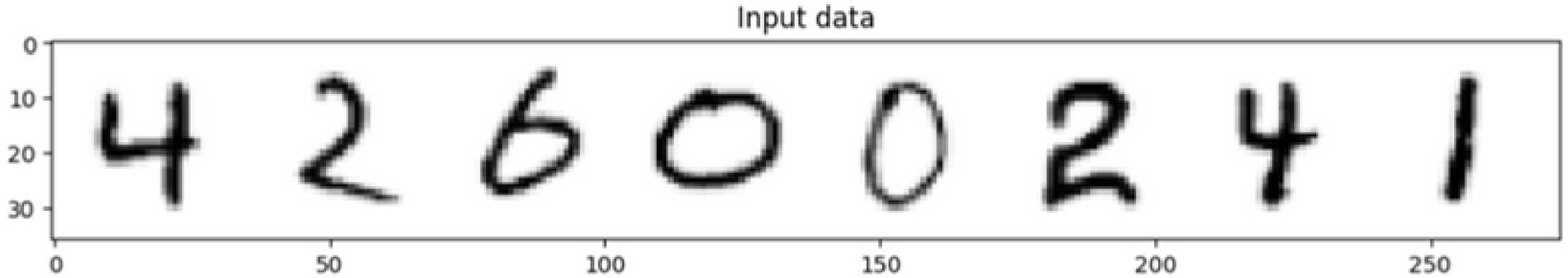


DeepFool Training U-net

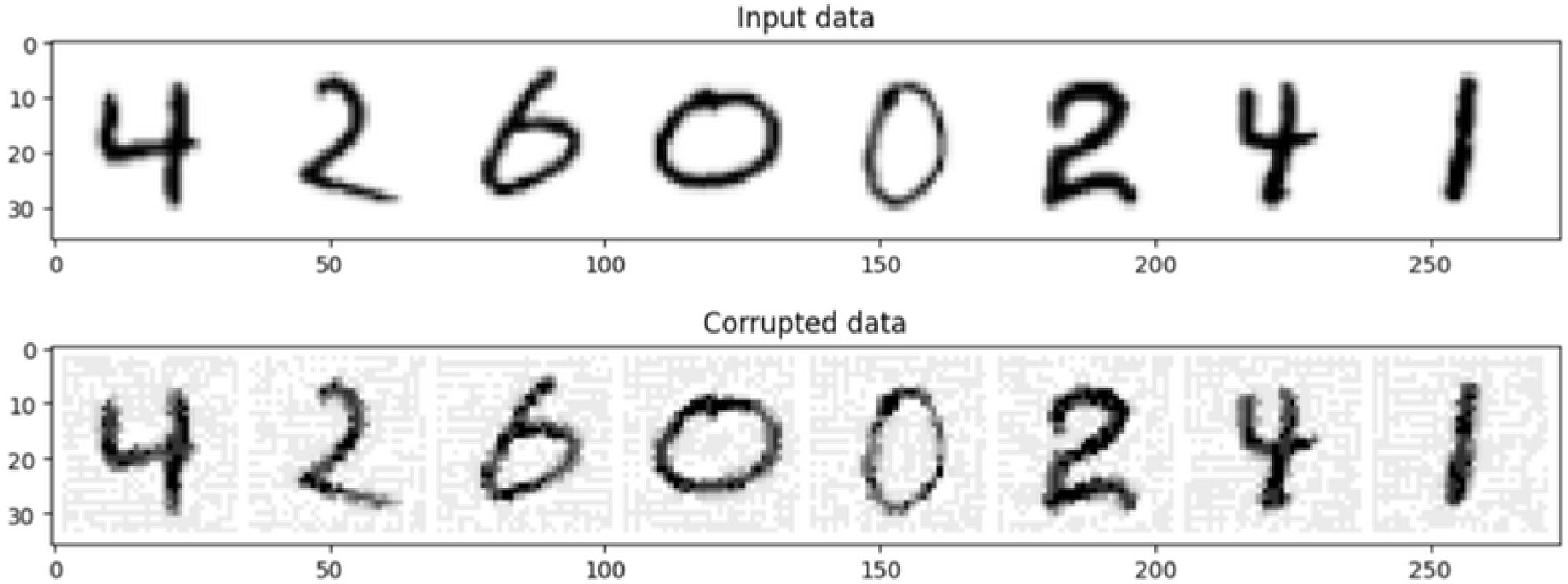


FGSM Attack

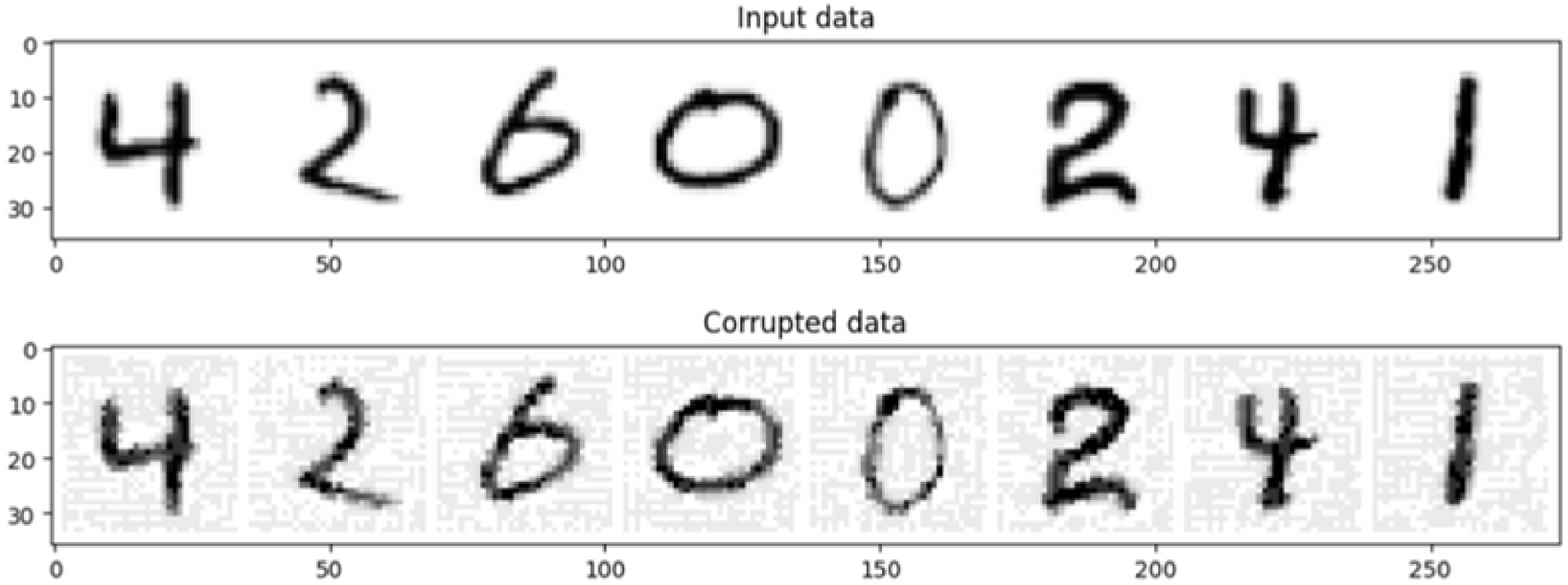
FGSM Attack



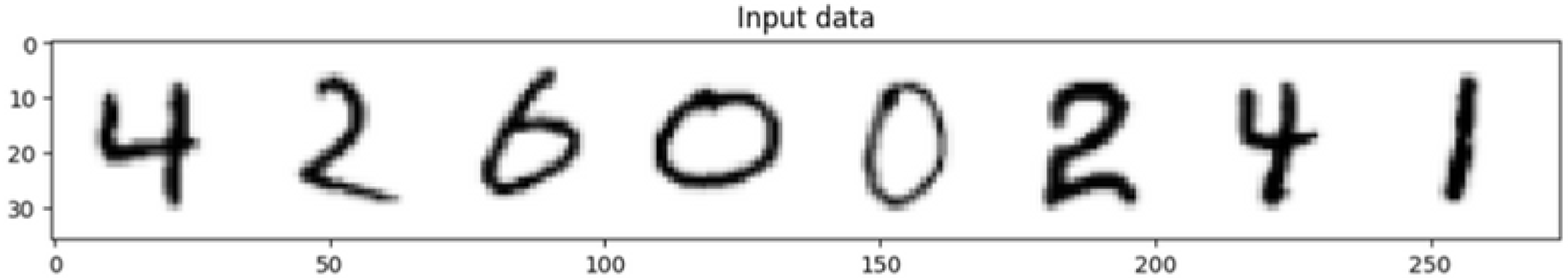
FGSM Attack

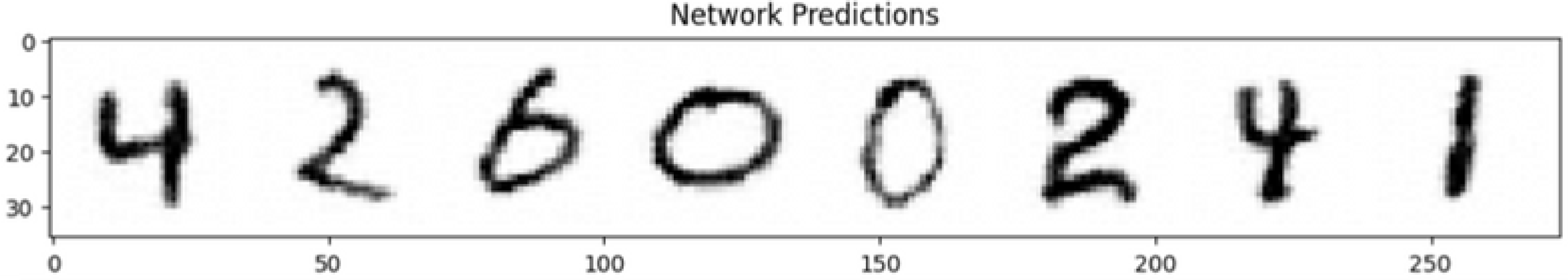

Input data
0
10
20
30
0
50
100
150
200
250
Corrupted data
0
10
20
30
0
50
100
150
200
250

U-Net Prediction



U-Net Prediction





DeepFool Training & FGSM Attack

	Original Data	FGSM Data	U-net Data
accuracy	98.1%	22.4%	80.3%

Attack Overview

	accuracy
Original	97.9%
Attacked by FGSM	22.5%
Attacked by Deepfool	1.9%

U-Net Defense Overview

	training by FGSM	training by Deepfool
attack by FGSM	82.6%	80.3%
attack by Deepfool	96.8%	98.1%

Summary

Summary

1

Generate adversarial images by FGSM model and purify those images with APE-GAN & Diffusion model

Summary

1. Generate adversarial images by FGSM model and purify those images with APE-GAN & Diffusion model

2. Try different model (Deepfool) to generate adversarial images

Summary

1. Generate adversarial images by FGSM model and purify those images with APE-GAN & Diffusion model
2. Try different model (Deepfool) to generate adversarial images
3. We found that U-net is a strong model to purify image, better than APE-GAN

Furture Work

Furture Work

1

Try FGSM & Deepfool to attack on different datasets, and find the purification result of different dataset

Furture Work

1. Try FGSM & Deepfool to attack on different datasets, and find the purification result of different dataset

2. Find other models that can defend attack more effectively

Furture Work

1. Try FGSM & Deepfool to attack on different datasets, and find the purification result of different dataset
2. Find other models that can defend attack more effectively
3. Try to defend different data type, such as video and text

Thanks for Listeing !