# Data Augmentation Using Generative Adversarial Networks (GANs) on the NEU Dataset

Instructor: Dr. Mehrandezh
Student: Marzieh Zamani

# Why GANs?

- Problem:
  - Insufficient amount of training data

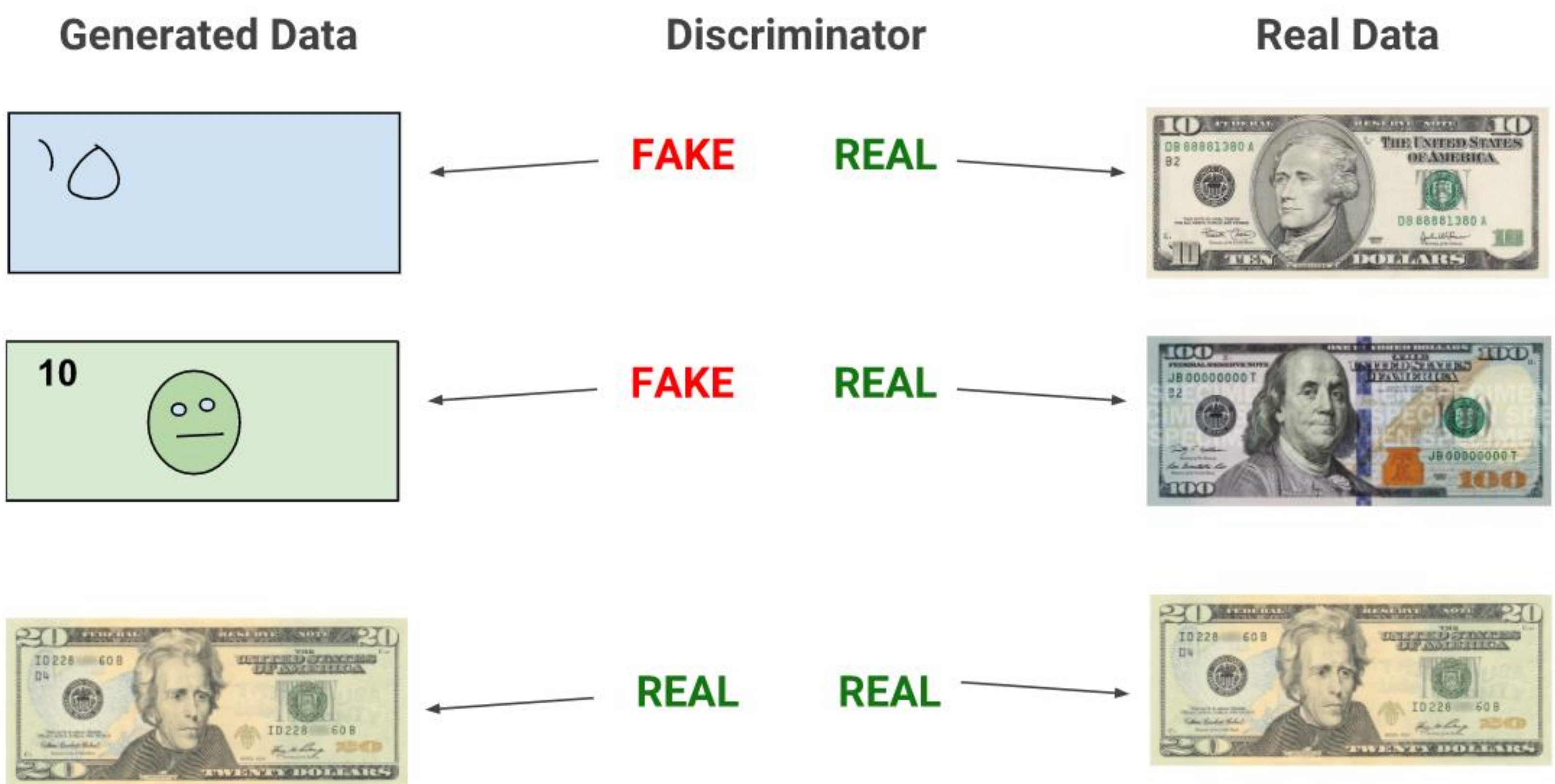
- Solution:
  - Data augmentation techniques: slightly rotating, flipping, or distorting the original data into new training data
  - GANs: generating new (synthetic) images from original images

# Generative Adversarial Networks (GANs)

- **The generator** network that takes a random noise vector as input; and produces “synthetic” images as output.

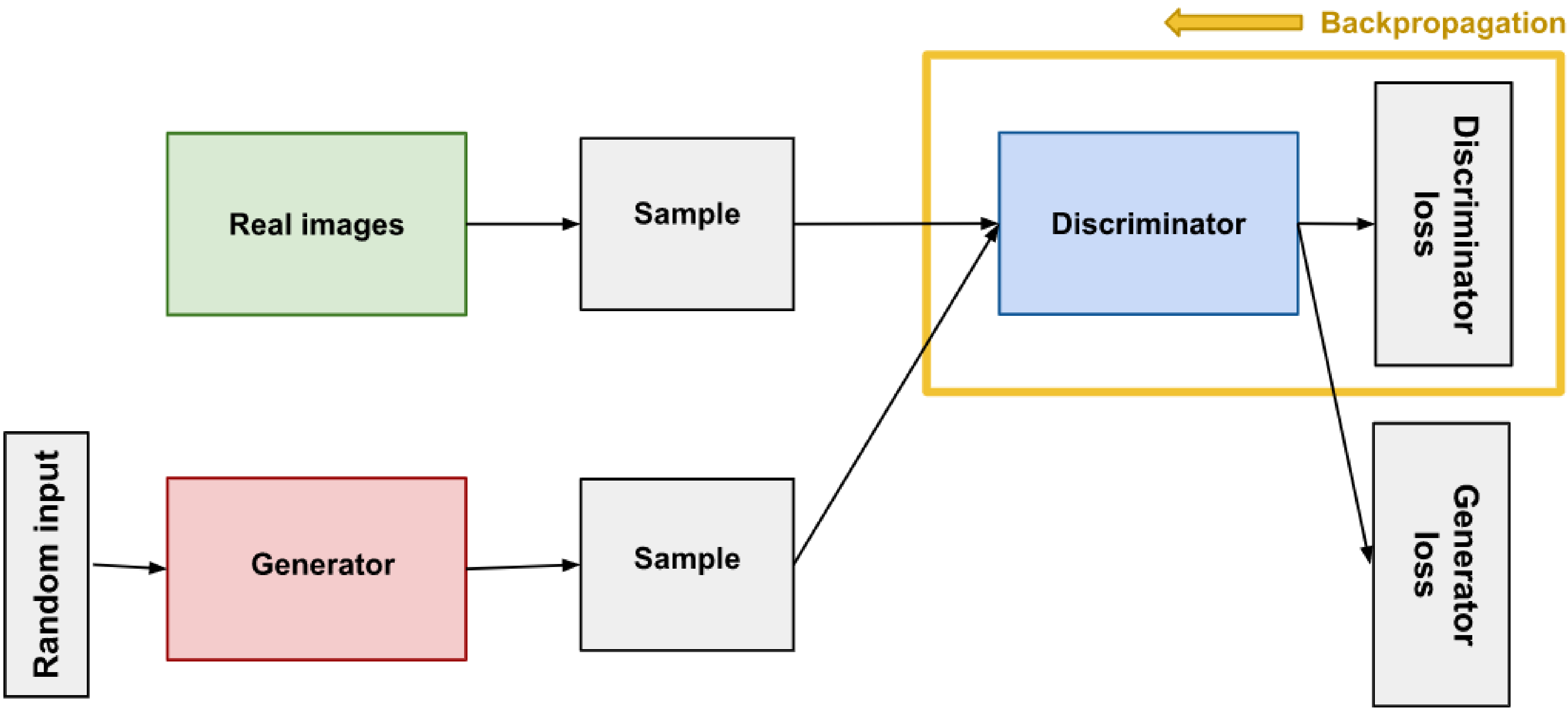
- **The discriminator** network that takes images and learns to distinguish between the “authentic” and “synthetic” images.

# Simplified GAN training process [3]

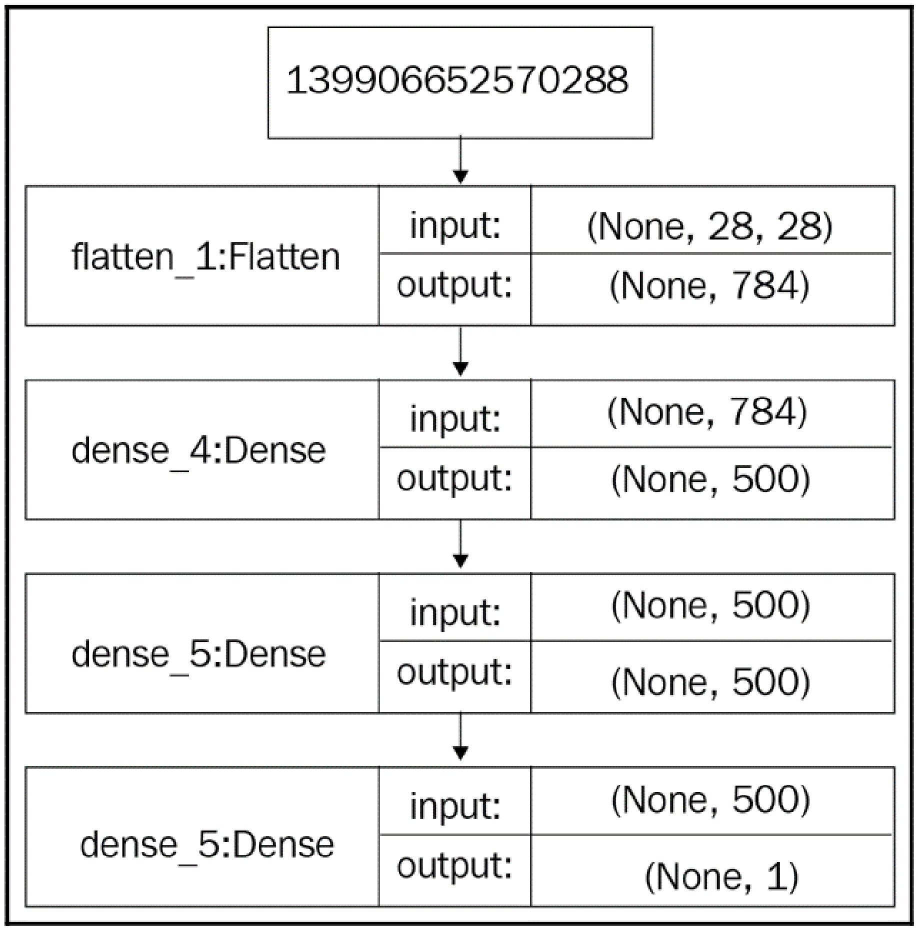


# The GAN discriminator [3]

- Input data:
  - Authentic (real) data
  - Synthetic (fake) data
- Output:
  - Classification results

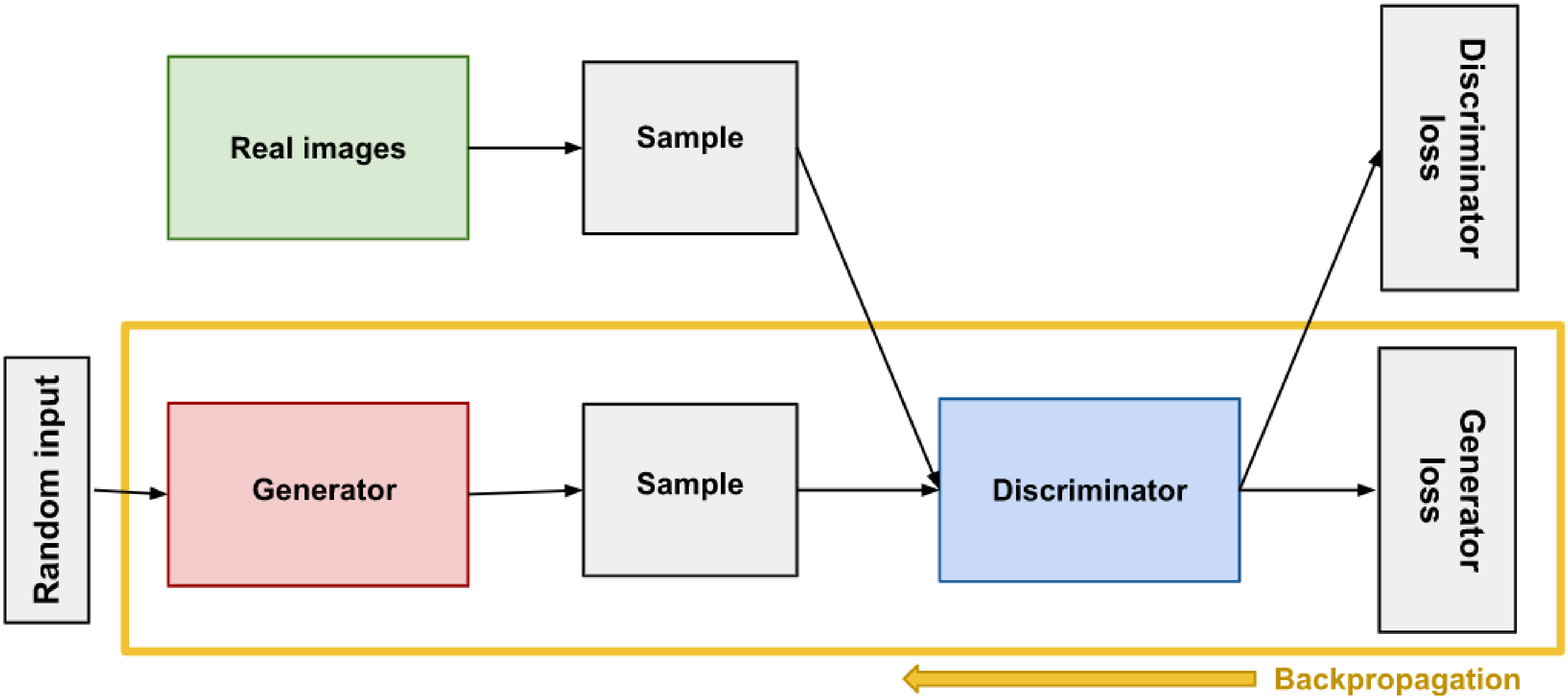


# The architecture of the discriminator [4]

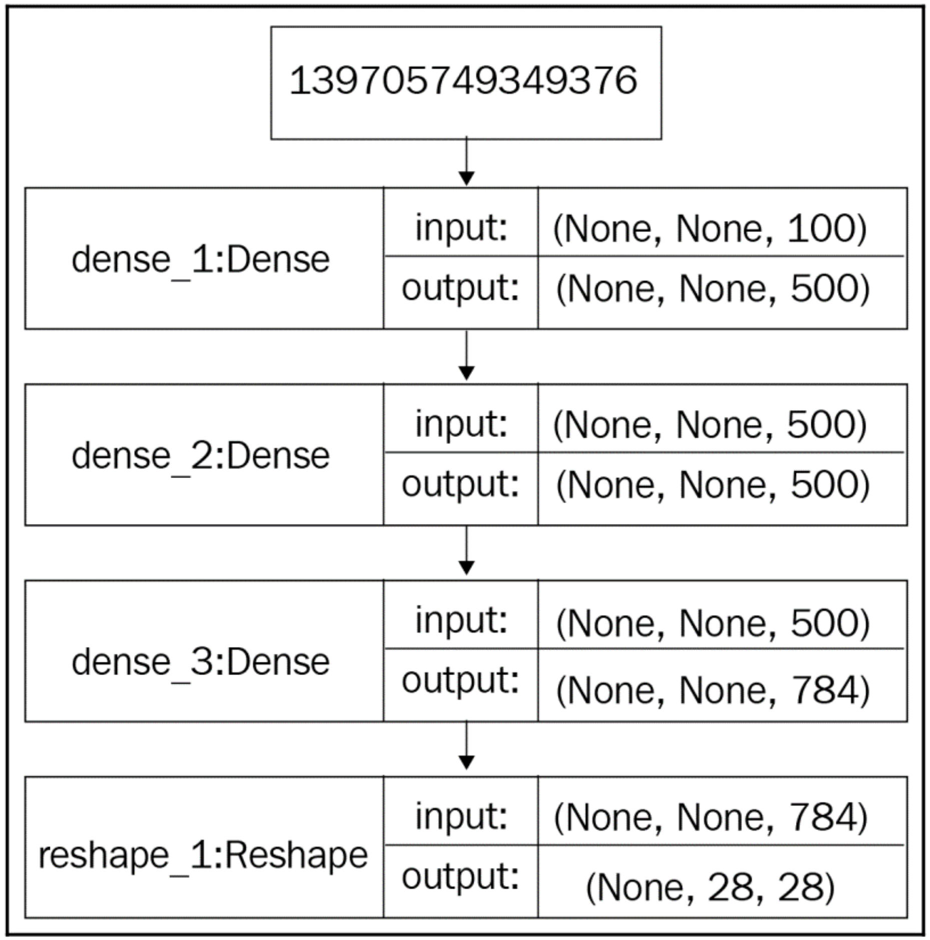


# The GAN generator [3]

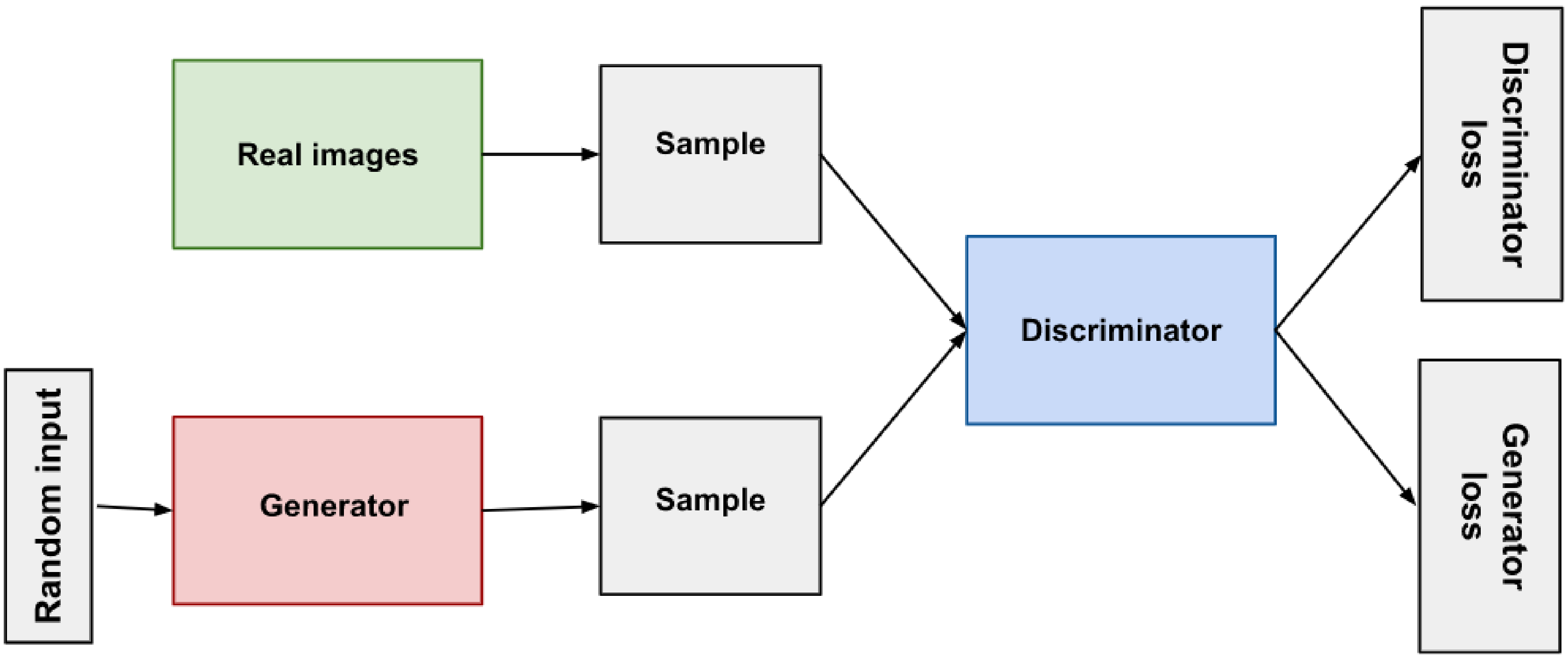
- Input:
  - Random noise vector
- Output:
  - Synthetic images



# The architecture of the generator [4]



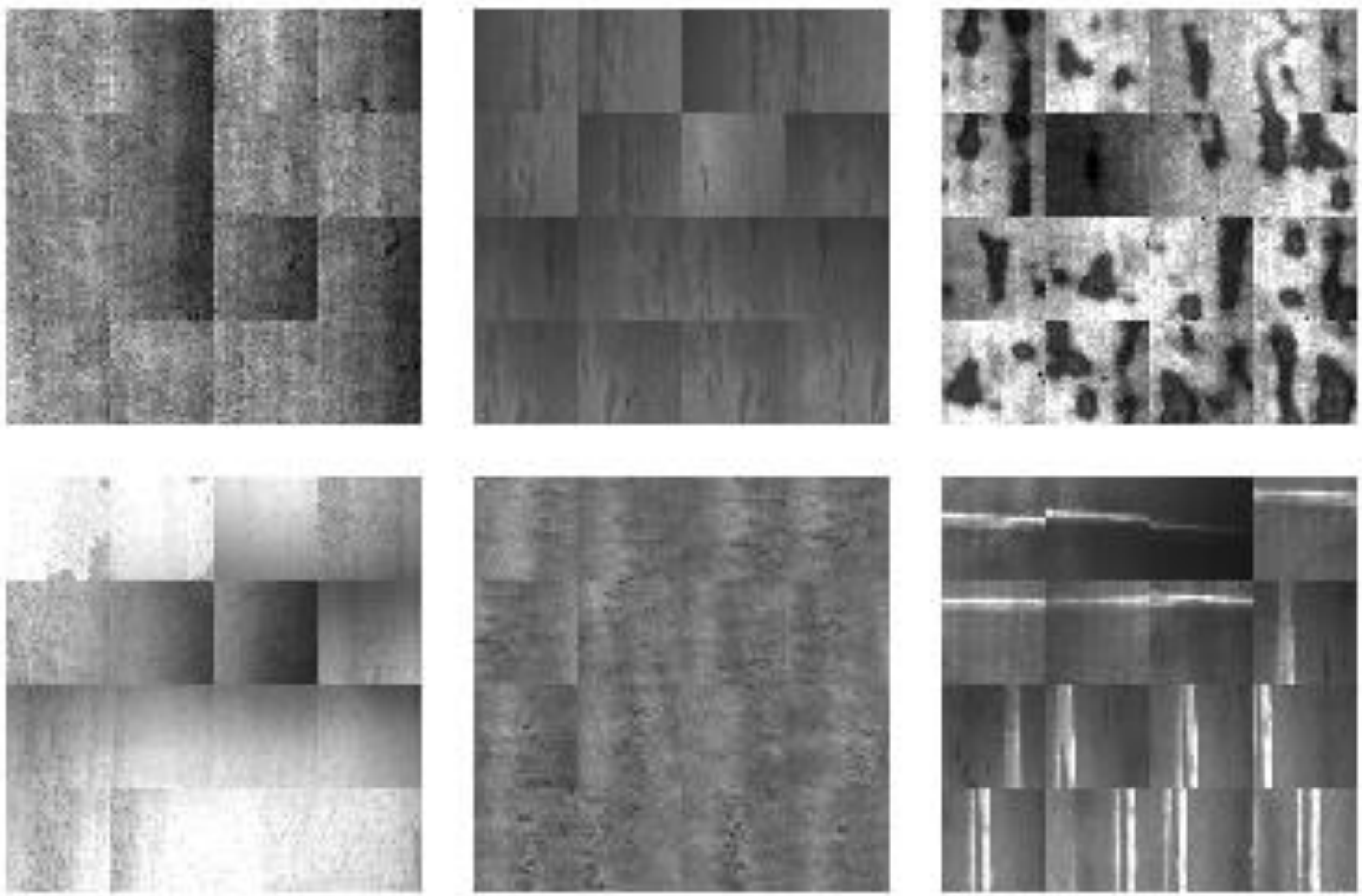
# Overall GAN training process [3]



# Dataset: The NEU surface defect database [5, 6]

- Surface defect database
- Provided by the Northeastern University (NEU)
- 1800 grayscale images
- 300 images x 6 classes:
  - Class 0: Crazing (Cr)
  - Class 1: Inclusion (In)
  - Class 2: Patches (Pa)
  - Class 3: Pitted Surface (PS)
  - Class 4: Rolled-In Scale (RS)
  - Class 5: Scratches (Sc)

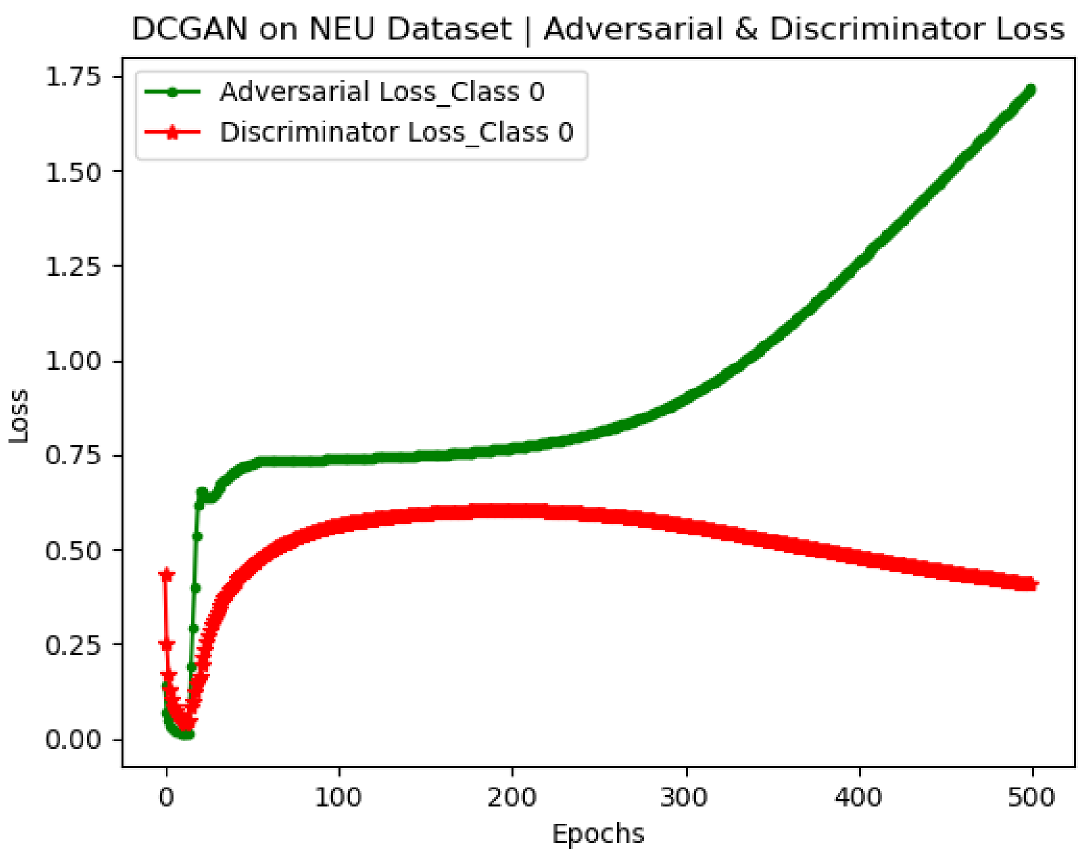
# Dataset:
# The NEU surface defect database [5, 6]



# Training DCGAN on NEU Dataset

- **Input data:**
  - 6 classes x 300 images x 28 x 28 x 1
  - Random noise 256 x 100
- **Number of epochs** = 500
- **Batch size** = 16
- **Discriminator Optimizer** = Adam(lr=0.0002, beta_1=0.5, decay=0.0002 / NUM_EPOCHS)
- **GAN Optimizer** = Adam(lr=0.0002, beta_1=0.5, decay=0.0002 / NUM_EPOCHS)
- **Outputs:**
  - Discriminator and adversarial loss
  - Generated 256 images x 28 x 28 x 1

# DCGAN on NEU Dataset
# => Adversarial & Discriminator Loss



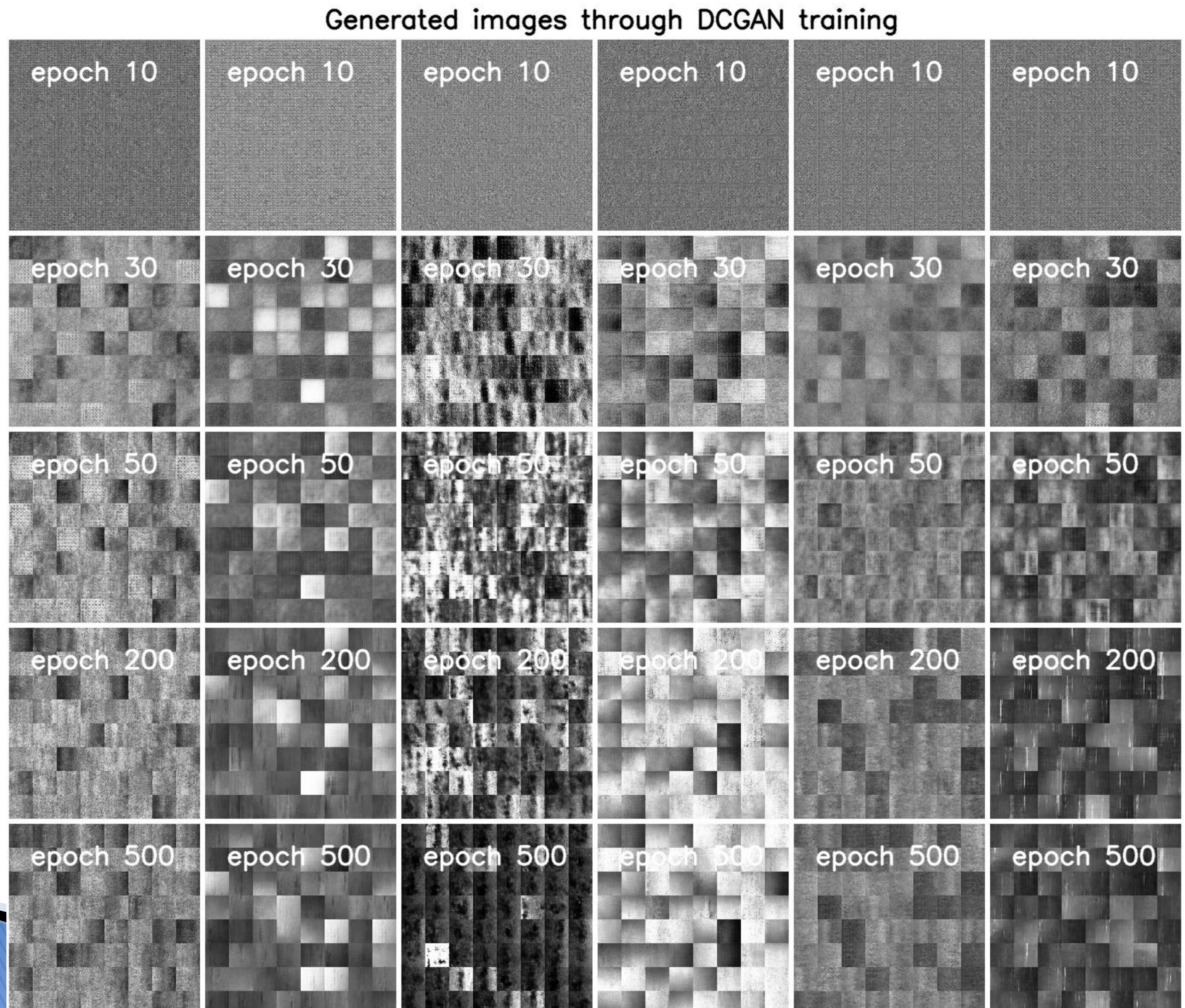
# DCGAN on NEU Dataset
# => Adversarial & Discriminator Loss

## Remarks from results:

- The discriminator loss
  - Decreases for a few epochs as it is learning.
  - Increases until epoch 200 due to the fact that generated images are more real-looking
  - After epoch 200, the discriminator loss is decreasing again due to overfitting of generator.

- The adversarial loss (overall loss)
  - Increasing at first which shows that the increasing discriminator loss outweighs the decreasing generator loss.
  - Then, it reaches some stability at around epoch 200 which shows the balance between discriminator and generator losses.
  - After epoch 200, the adversarial loss is sharply increasing due to overfitting of generator.

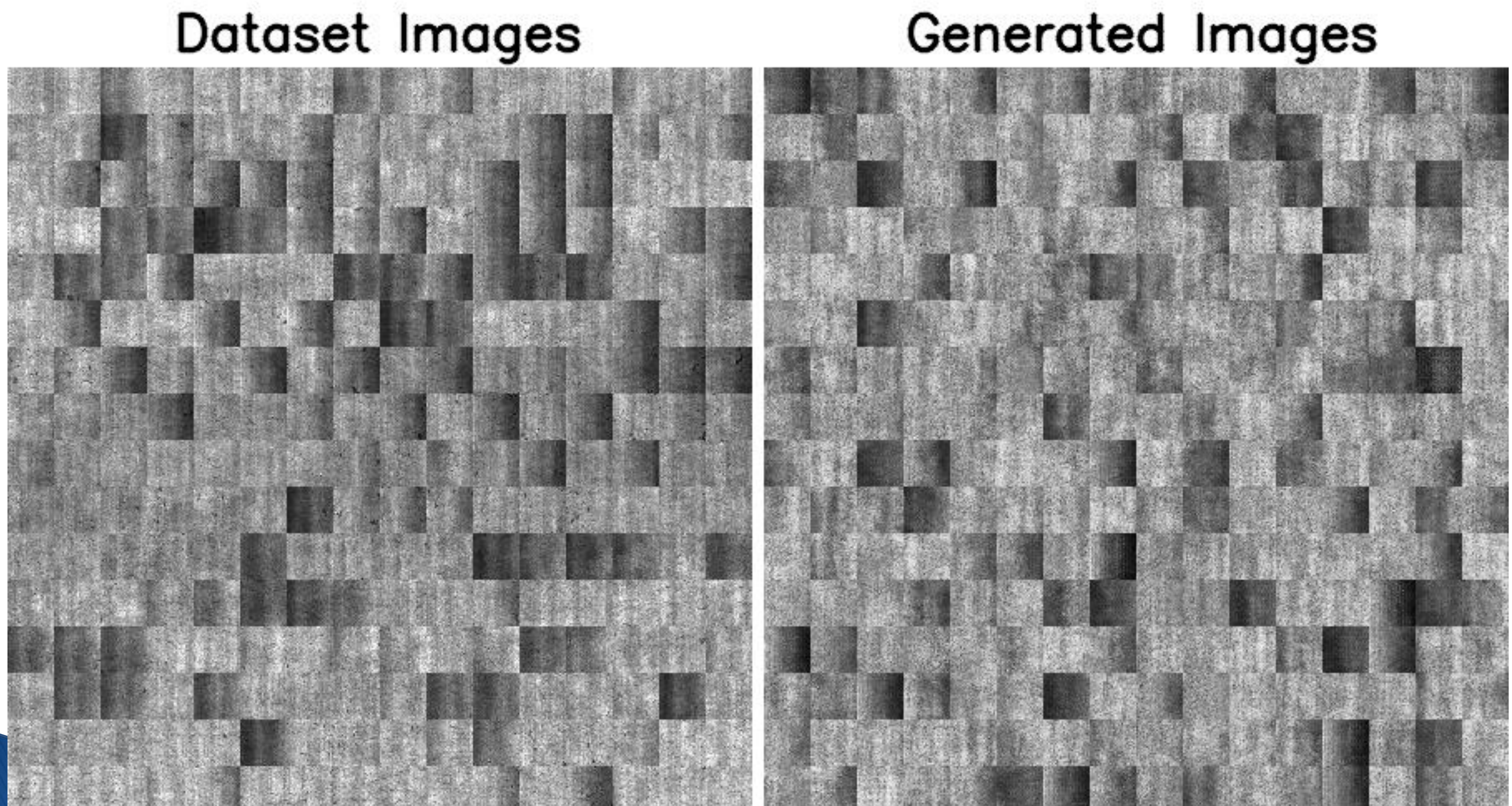
# DCGAN on NEU Dataset
# => Generated images during GAN training



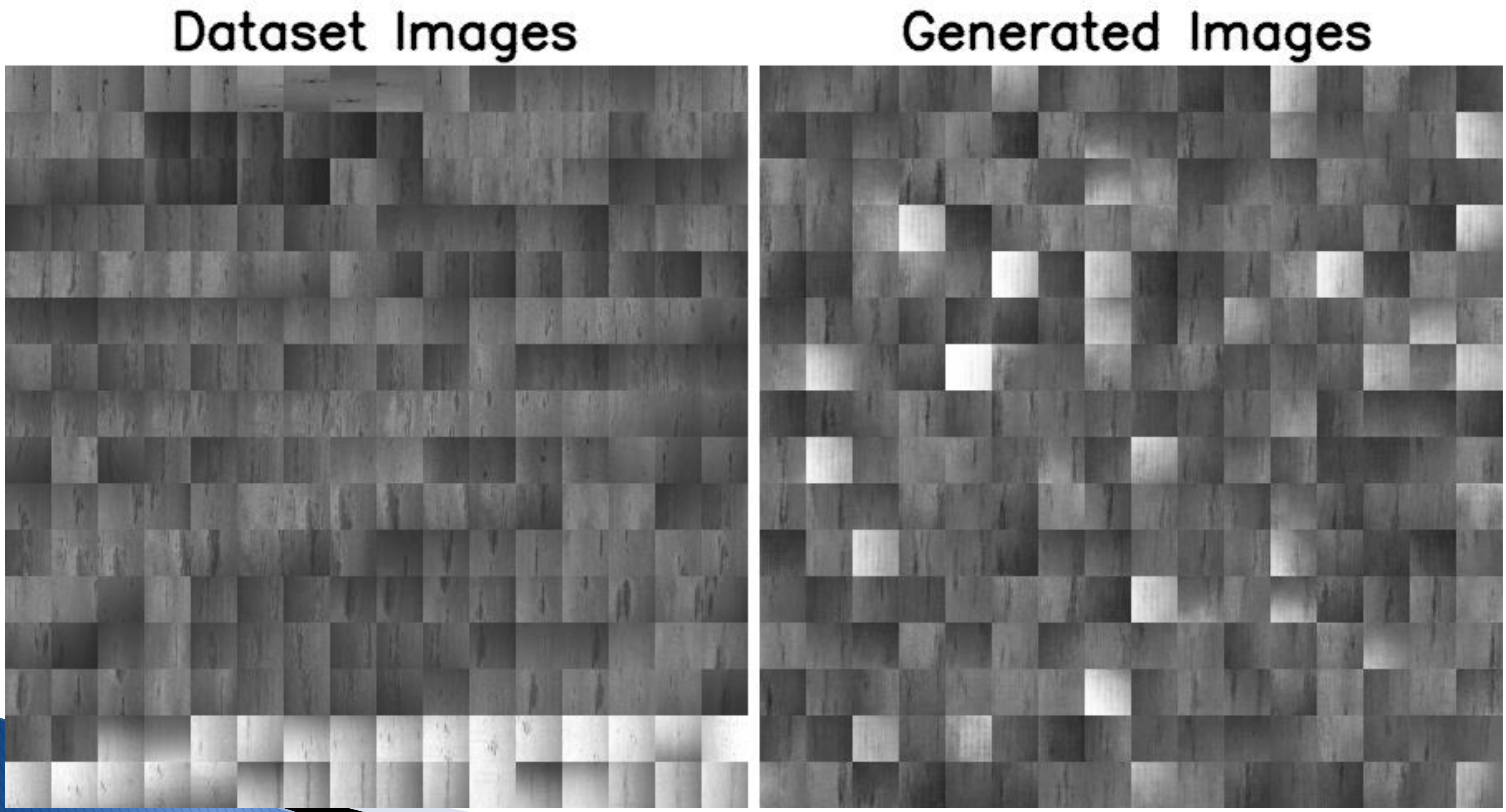
# DCGAN on NEU Dataset
# => Generated images during GAN training

- **Remarks from results:**
  - For all classes, the first generated images are just noisy grey images without any pattern but soon after epoch 20, the patterns of the original classes are showing up in generations.
  - From epoch 50, the generated images are not changing significantly but some important details are improving up to epoch 200 (best performance).
  - After epoch 200, the generator is overfitting and its outputs are drifting away from original images in some small details.

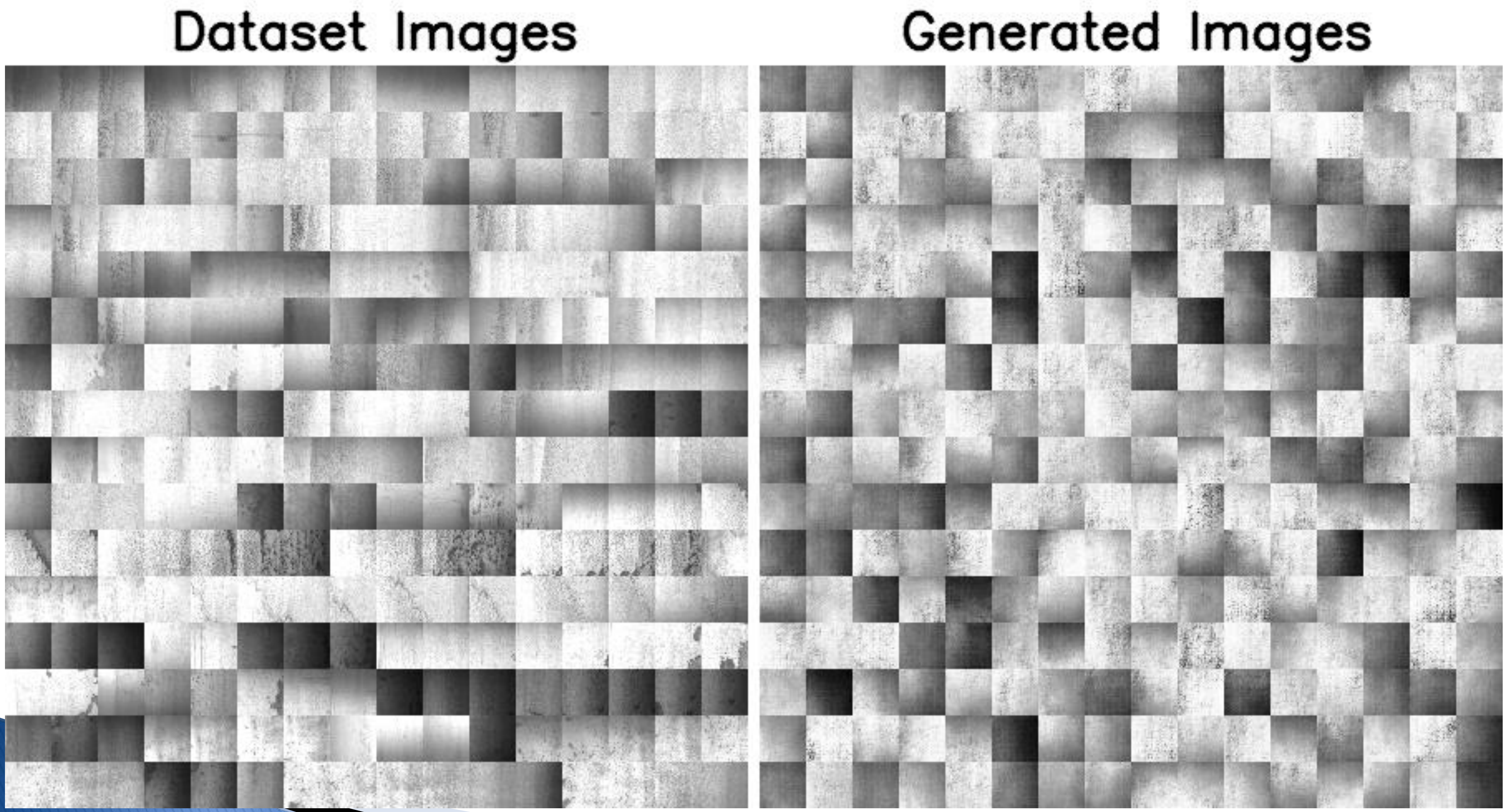
# DCGAN on NEU Dataset
# => Dataset images vs generated images for class 0



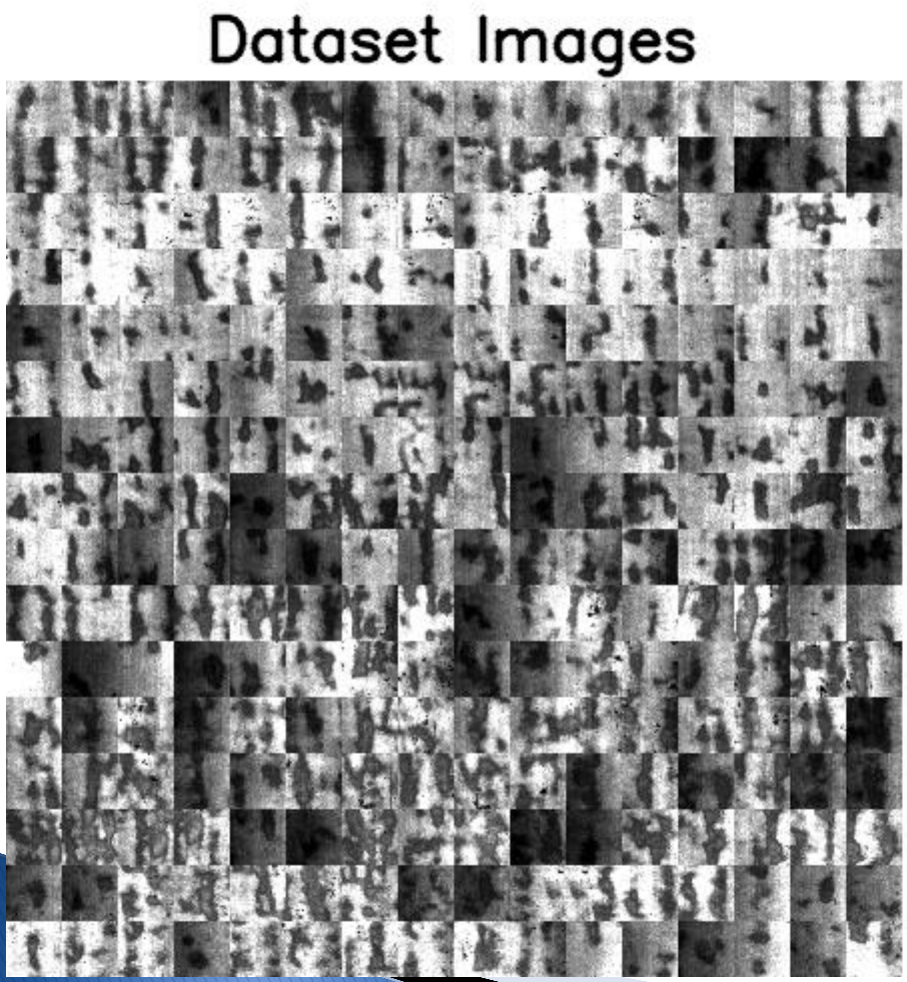
# DCGAN on NEU Dataset
# => Dataset images vs generated images for class 1

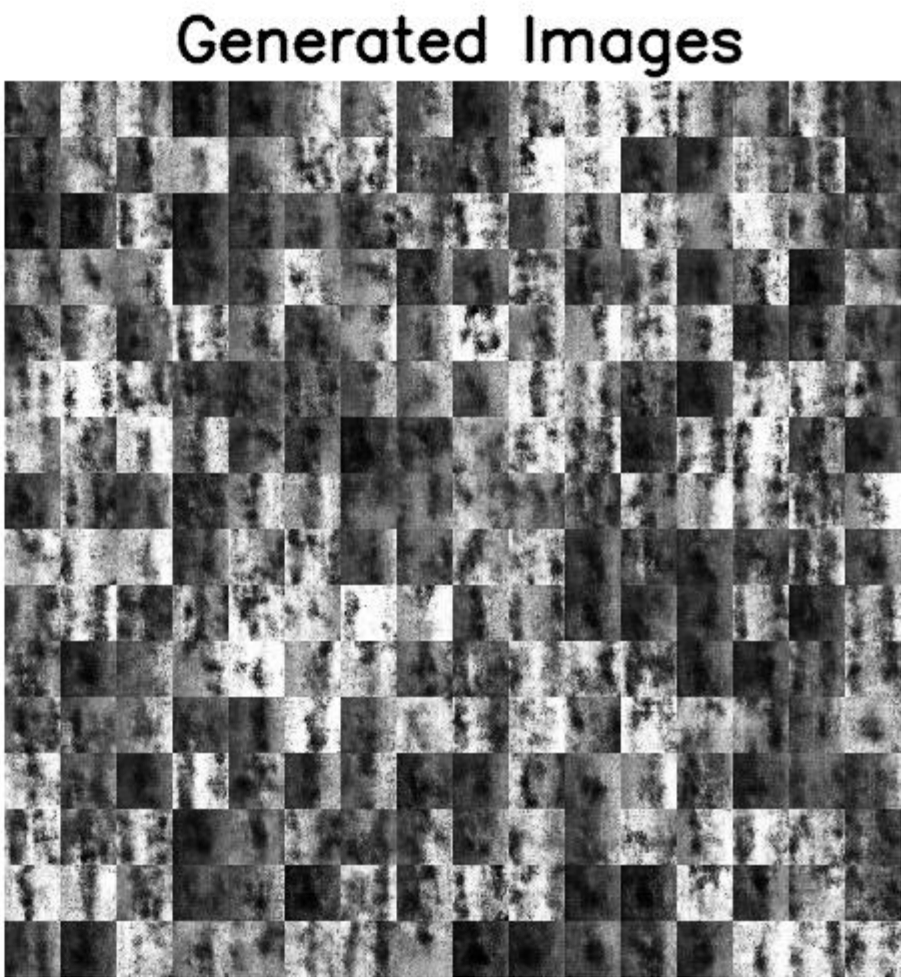


# DCGAN on NEU Dataset
# => Dataset images vs generated images for class 2

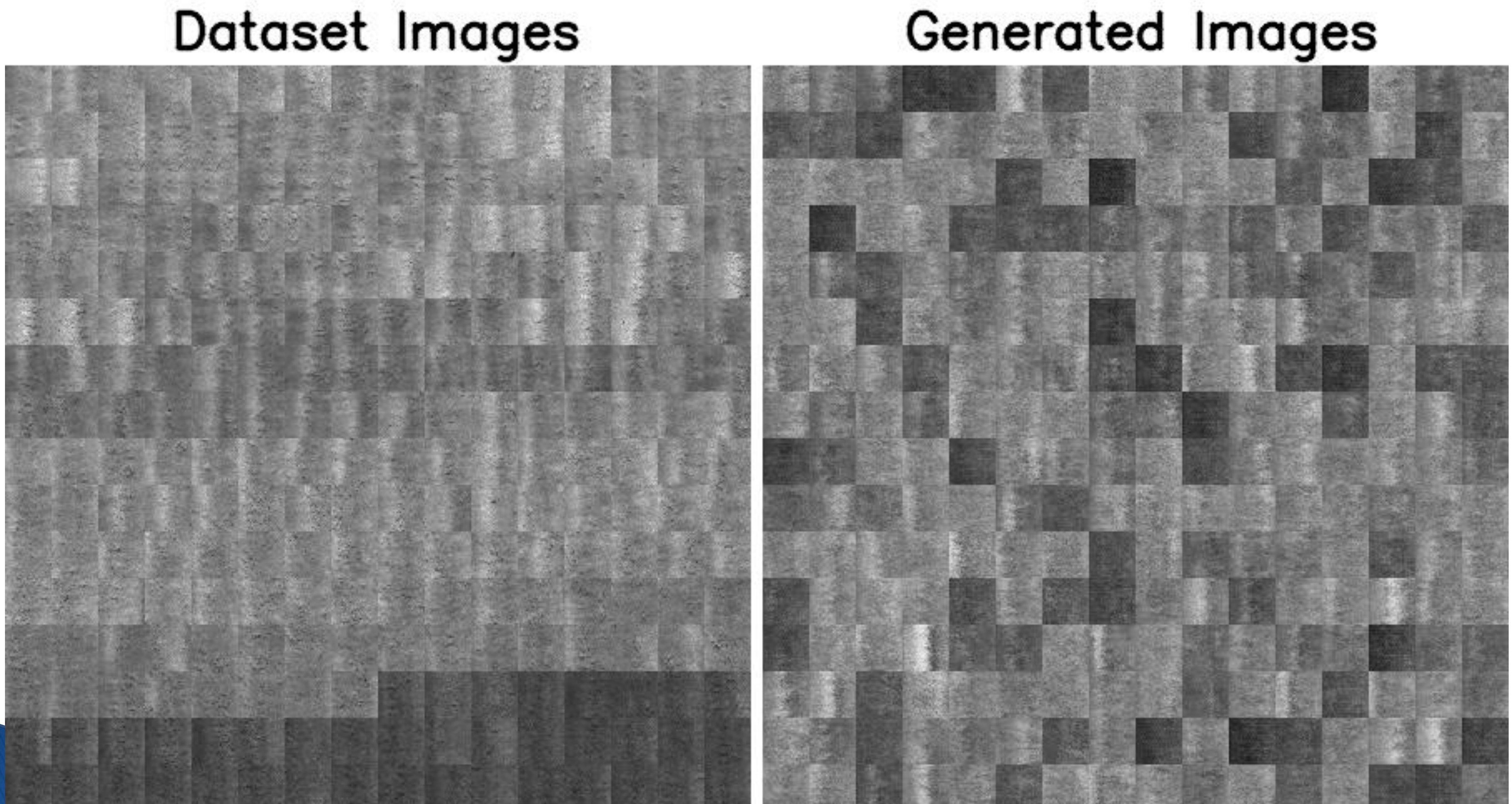


# DCGAN on NEU Dataset
# => Dataset images vs generated images for class 3

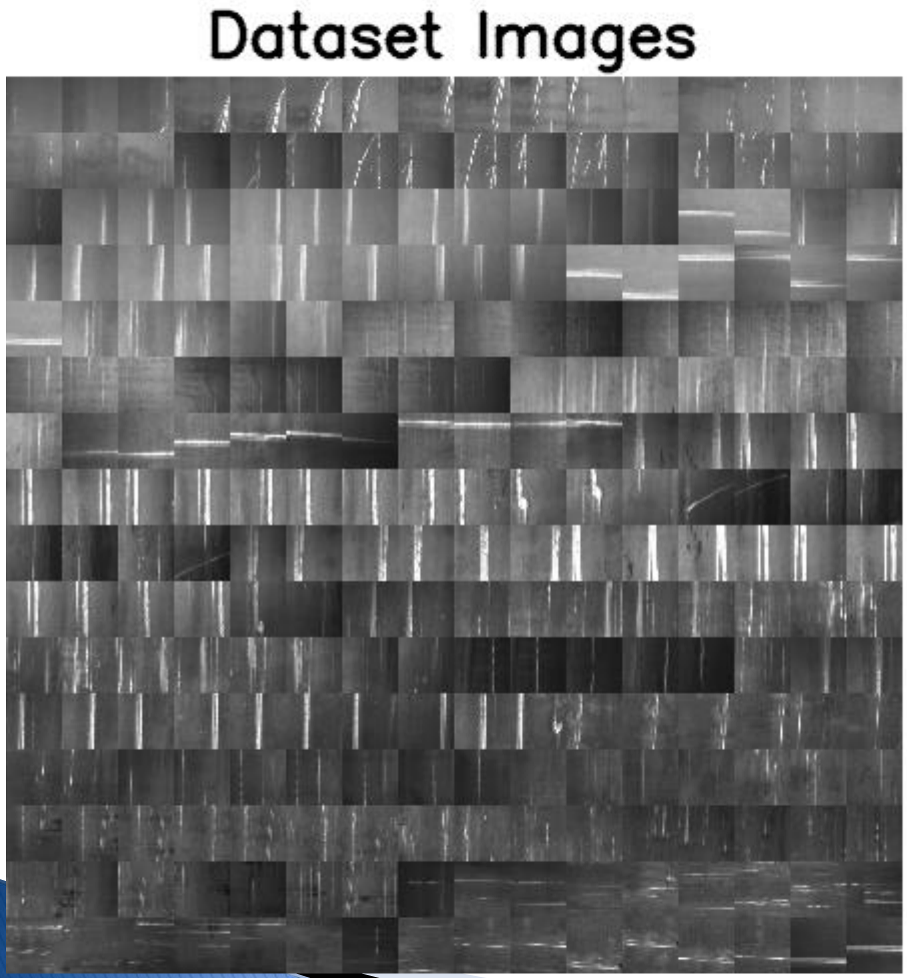


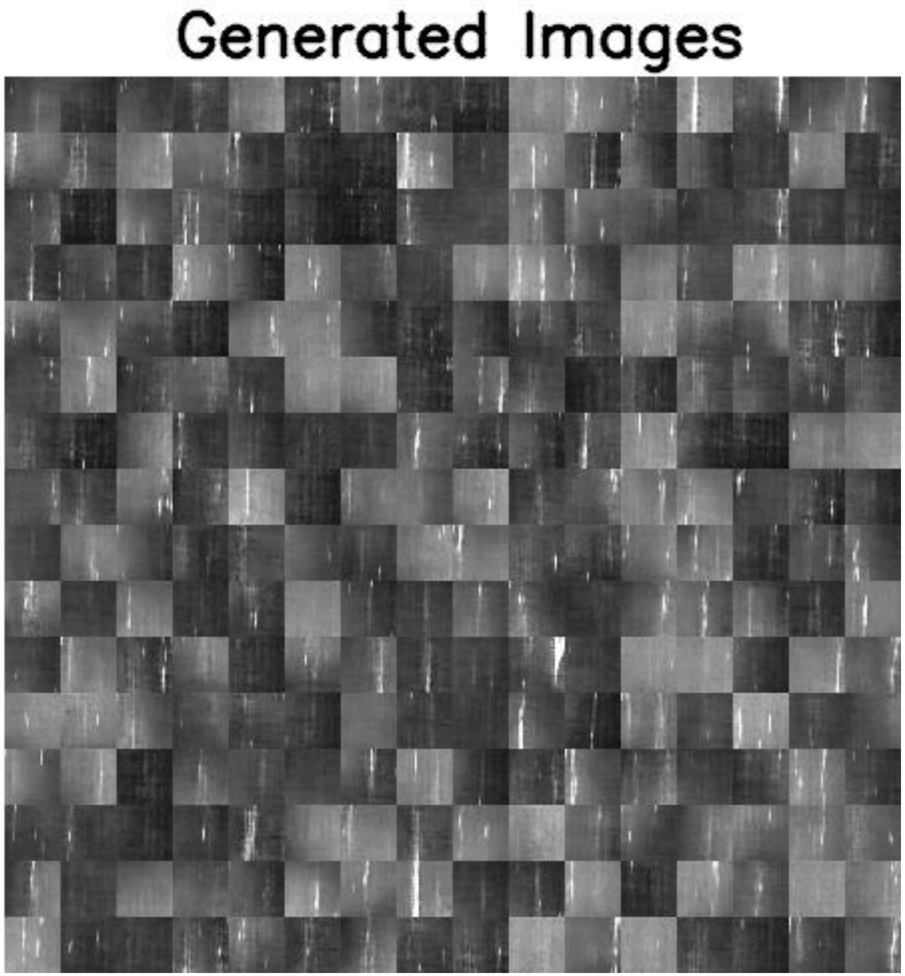



# DCGAN on NEU Dataset
# => Dataset images vs generated images for class 4



# DCGAN on NEU Dataset
# => Dataset images vs generated images for class 5



# DCGAN on NEU Dataset
# => Dataset images vs generated images

- **Remarks from results:**
  - For all classes except class 5, the generated images are almost identical to original images.
  - The generated images for class 5 are showing similar details but not exactly the same as the database images.

# References


1. https://dida.do/blog/data-augmentation-with-gans-for-defect-detection
2. Rosebrock, A. (2017). Deep Learning for Computer Vision with Python: Practitioner Bundle. PyImageSearch.
3. https://developers.google.com/machine-learning/gan/discriminator
4. Ahirwar, K. (2019). Generative Adversarial Networks Projects: Build Next-generation Generative Models Using TensorFlow and Keras. Packt Publishing Ltd.
5. http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html (Link not working)
6. K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," Applied Surface Science, vol. 285, pp. 858-864, Nov. 2013. (paper)

# Thanks for your attention

Questions are welcome