# GAN-based defect transfer model for steel surface defects generation


Student: Francesco Gazzola
Supervisor: Emanuele Menegatti, Stefano Totaro
Co-supervisor: Alberto Bacchin


*Padova, 27-11-2023*


DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

# Table of contents

# Introduction

- During the steel manufacturing process, diverse defects might occur due to the different processing techniques such as the rolling equipment
- Defects can lead to safety hazard and reduce the product performance and quality

  → we need to detect the defects

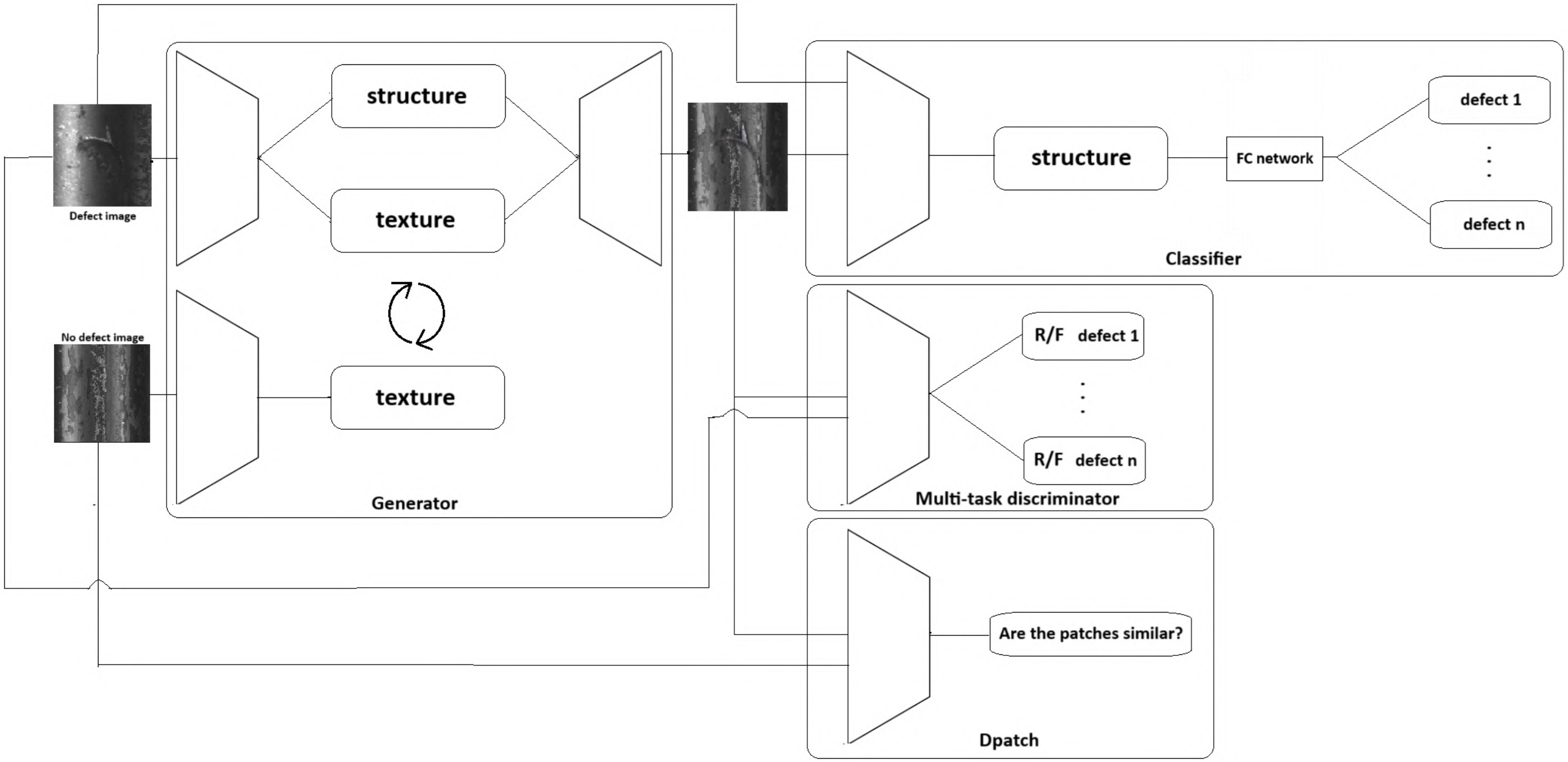
  - But little data in a real-world scenario: it is easy to collect no-defective images, but defects images are very few. This leads to bad accuracy of a classifier.

    → we need data-augmentation

**THESIS OBJECTIVE**: given images of defects on materials of texture X, transfer them to images of surfaces with a different texture Y in order to enhance diversity

# Method – model architecture

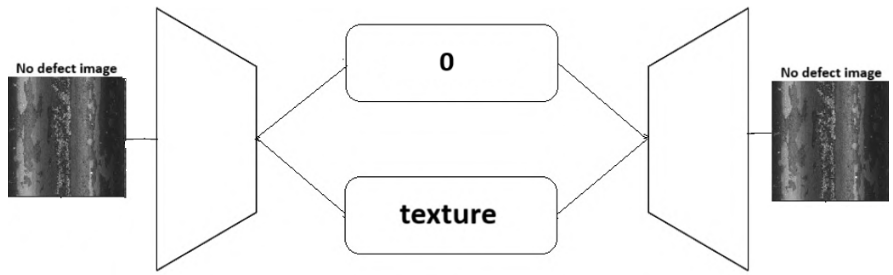
- Based on the swapping autoencoder
  - Disentangled learning



# Method – disentangled learning

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

- Anchor domain hypothesis




- Given an image $X_i$ s.t. $(z^i_s, z^i_t)$=Encoder($X_i$) are the structure and the texture code, and $X_1$=defective image, $X_2$=no-defective image, $X_3=G(z^1_s, z^2_t)$ i.e. the hybrid image, $X_4=G(0, z^2_t)$ i.e. the reconstruction of $X_2$
    - **Feature reconstruction loss**

$$L_{feature}(E,G) = MSE(z_s^3, z_s^1) + MSE(z_t^3, z_t^2) + MSE(z_t^4, z_t^2)$$

    - **Structure consistency loss**

$$L_{Structure}(E,G) = E_{x \sim X}[||(x_1 - G(0,z_t^1)) - (G(z_s^1,z_t^2) - x_2)||_1] + E_{x \sim X}[||((G(z_s^1,z_t^1) - G(0,z_t^1)) - (G(z_s^1,z_t^2) - x_2)||_1]$$

# Results - Dataset

- Dataset statistics

| scratch | shell | mark | scoring | tear | notching | normal |
|---|---|---|---|---|---|---|
| 105 | 28 | 163 | 25 | 167 | 40 | 7262 |

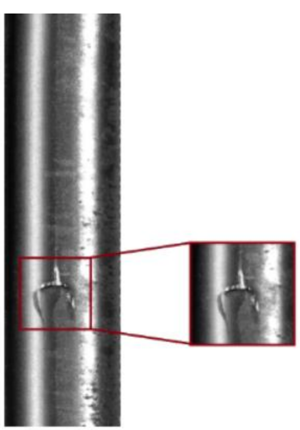
**Table 4.1:** Original industrial set

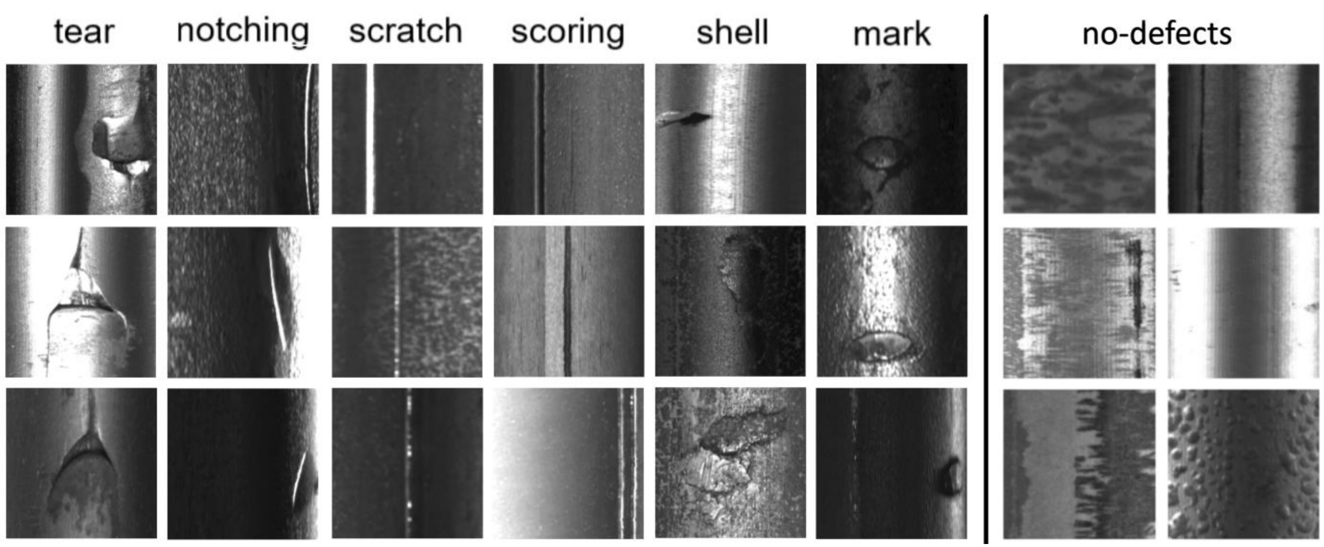
+ horizontal and vertical flip

| scratch | shell | mark | scoring | tear | notching | normal |
|---|---|---|---|---|---|---|
| 105 | 84 | 163 | 75 | 167 | 120 | 727 |

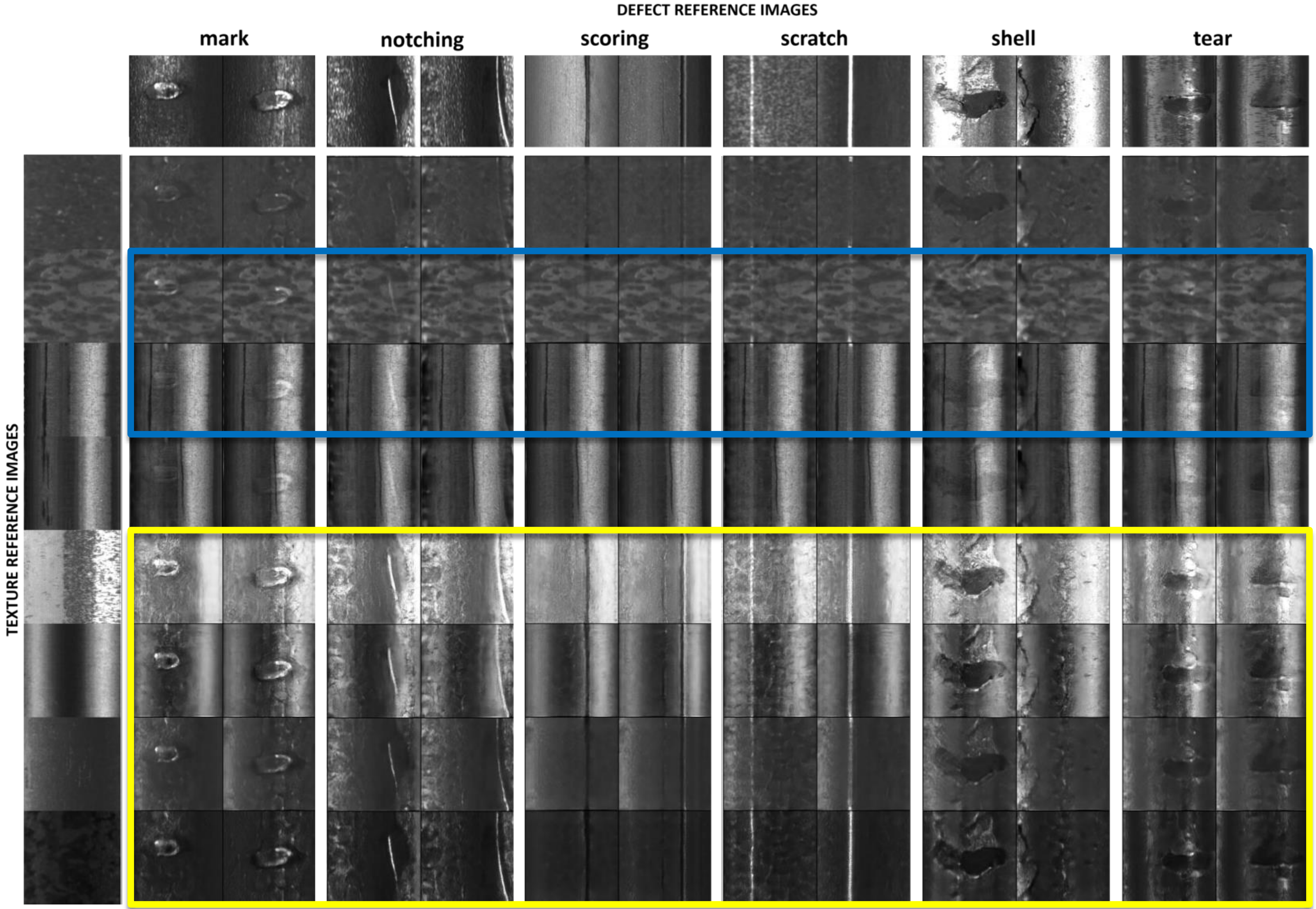
**Table 4.2:** Training set

- Training dataset samples example:





# Results – Qualitative research



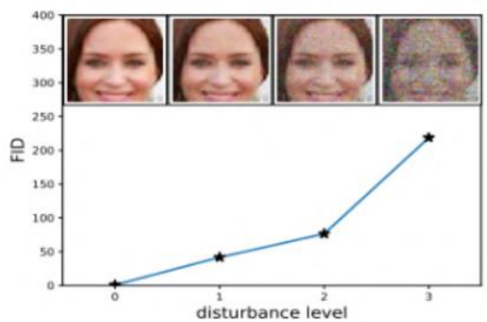
# Results – Quantitative research

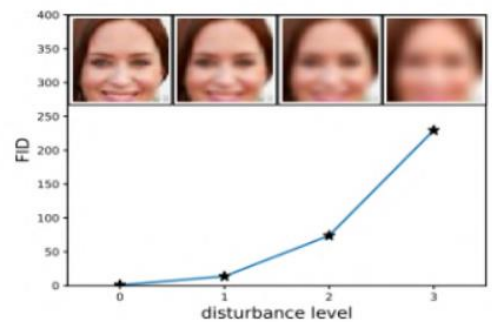


- Fréchet Inception Distance (FID) and Kernel Inception distance (KID) score
  - They are based on statistics: mean and covariance
  - The lower, the better the realness and the diversity of the generated images

$$FID = ||\mu_r - \mu_g||^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

$$KID = ||\mu_r - \mu_g||^2$$




| | scratch | shell | mark | scoring | tear | notching |
|---|---|---|---|---|---|---|
| FID | 2.45 | 4.93 | 0.72 | 7.12 | 0.74 | 1.15 |
| KID | 0.10 | 0.07 | 0.13 | 0.11 | 0.06 | 0.19 |

**Table 4.3:** Quantitative evaluation

# Results – data augmentation

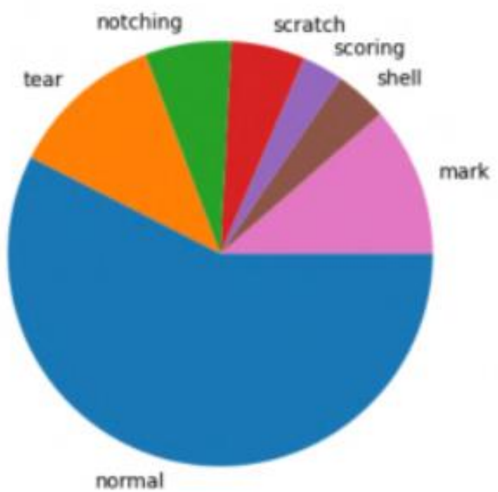
- Dataset

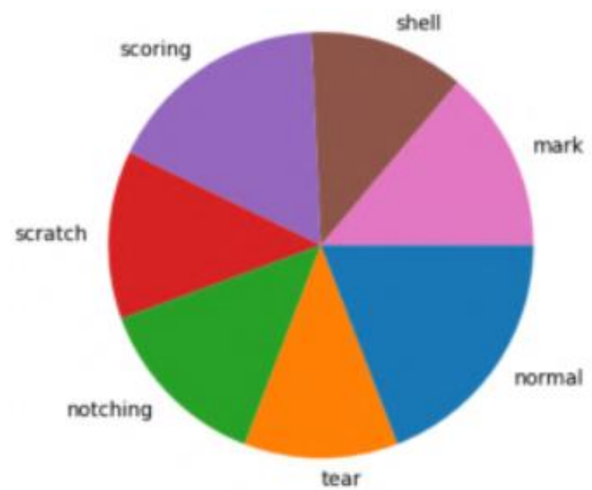
| scratch | shell | mark | scoring | tear | notching | normal |
|---|---|---|---|---|---|---|
| 59 | 42 | 120 | 33 | 124 | 69 | 604 |

**Table 4.4:** Training set for the classifier

| scratch | shell | mark | scoring | tear | notching | normal |
|---|---|---|---|---|---|---|
| 403 | 374 | 438 | 541 | 370 | 428 | 604 |

**Table 4.7:** Augmented training set for the classifier



# Results – data augmentation

- Better generalization on the **validation set**

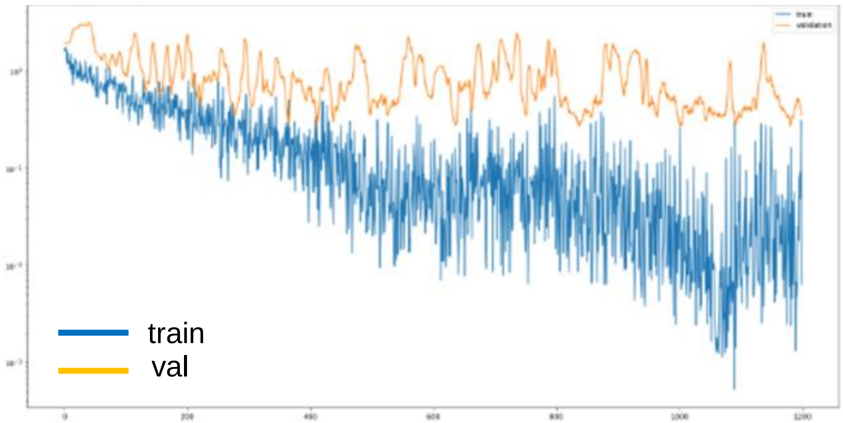



**Figure 4.19:** Loss ResNet50 classifier on no augmented data

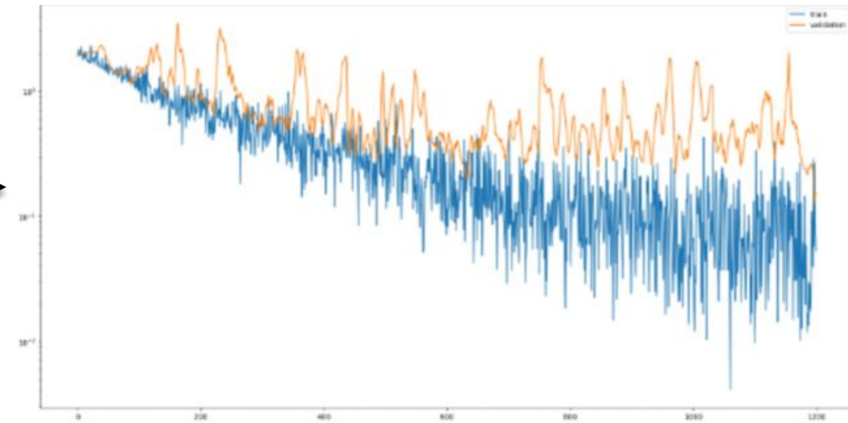


**Figure 4.20:** Loss ResNet50 classifier on augmented data

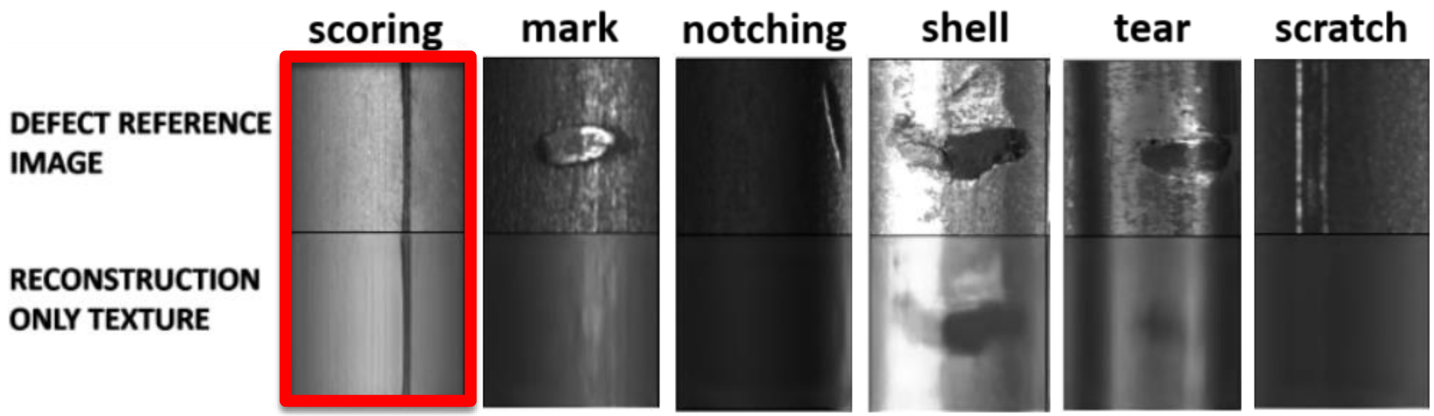
- Overall error rate diminished from 9.03% to 5.42% on the **test set**
- Single error rate per class improved too, but the scratch

| | scratch | shell | mark | scoring | tear | notching | normal |
|---|---|---|---|---|---|---|---|
| No augmented | 0% | 77.8% | 6.2% | 11.1% | 15.2% | 0% | 0% |
| Augmented | 11.4% | 11.1% | 3.1% | 0.0% | 9.1% | 0% | 0% |

# Conclusion

- Transferring the defects across multiple background or changing its style enhances the diversity of the dataset and thus counteracts the data-insufficiency problem in the real-world scenario

- Disentanglement mechanism needs to be improved



# Thanks for your attention

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

mermec

UNIVERSITAS STUDII PADUANI
MCCXXII

# Training objectives - Discriminators

- Adversarial loss

$$L^D_{adv}(E,G,D) = E_{x_1,x_2 \sim X}[log(D(x))] + E_{x_1,x_2 \sim X}[log(1 - D(G(z_s,z_t))]$$

- Texture adversarial loss

$$\begin{aligned} L^D_{CoccurGAN}(E,G,D) = & E_{x_2 \sim X}[log(D_{patch}(crop(x_2),crops(x_2)))] \\ & + E_{x_1,x_2 \sim X}[log(1 - D_{patch}(crop(G(z_s,z_t)),crops(x_2)))] \end{aligned}$$

- DISCRIMINATORS OBJECTIVE

$$L_D = L^D_{adv} + L^D_{CoccurGAN}$$

- CLASSIFIER OBJECTIVE

$$L_C = \mathrm{CrossEntropy}(P(y|X), y)$$

- GENERATOR OBJECTIVE

$$L_G = L_{rec} + 0.5L^G_{adv} + L^G_{CoocurGAN} + L_{Structure} + L_{feature} + L^G_{cls}$$

$$L^G_{cls}(E,G) = \mathrm{CrossEntropy}(P(y|X_{fake}), y)$$ Classification loss

$$L^G_{CoccurGAN}(E,G) = E_{x_1,x_2 \sim X}[-log(D_{patch}(crop(G(z_s,z_t)), crops(x_2))]$$ Texture adversarial loss

$$L^G_{adv}(E,G) = E_{x_1,x_2 \sim X}[-log(D(G(z_s,z_t)))]$$ Non-saturating adversarial loss

$$L_{rec}(E,G) = E_{x \sim X}[||x_1 - G(z^1_s, z^1_t))||_1] + E_{x \sim X}[||x_2 - G(0, z^2_t))||_1]$$ Image reconstruction loss

# Issue – mode collapse

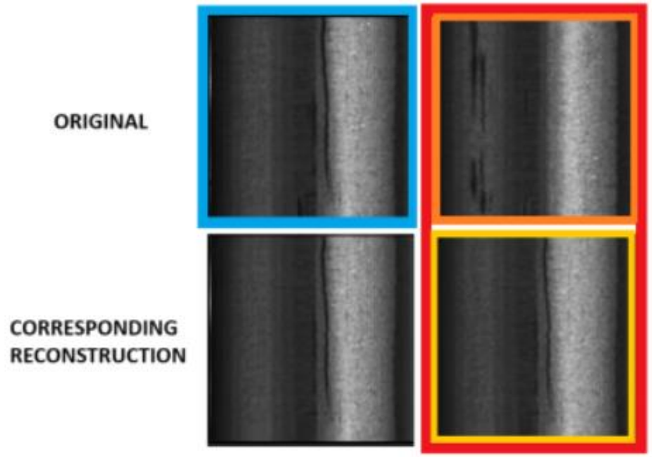



**Figure 4.4:** Highlighted in red is the suspected case of mode collapse. Given the texture reference image in the the orange box, the model reconstructs it as shown in the yellow box. However its reconstruction is identical to another texture image in the training dataset (shown in the blue box).

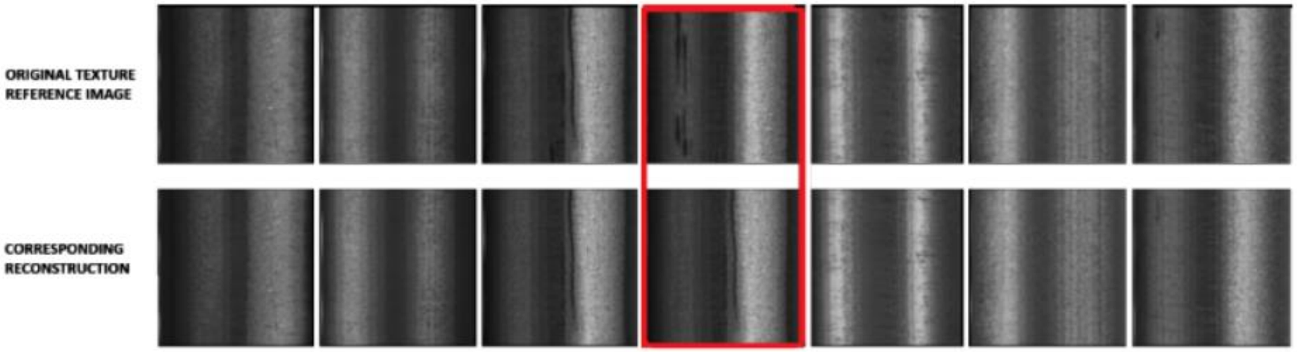
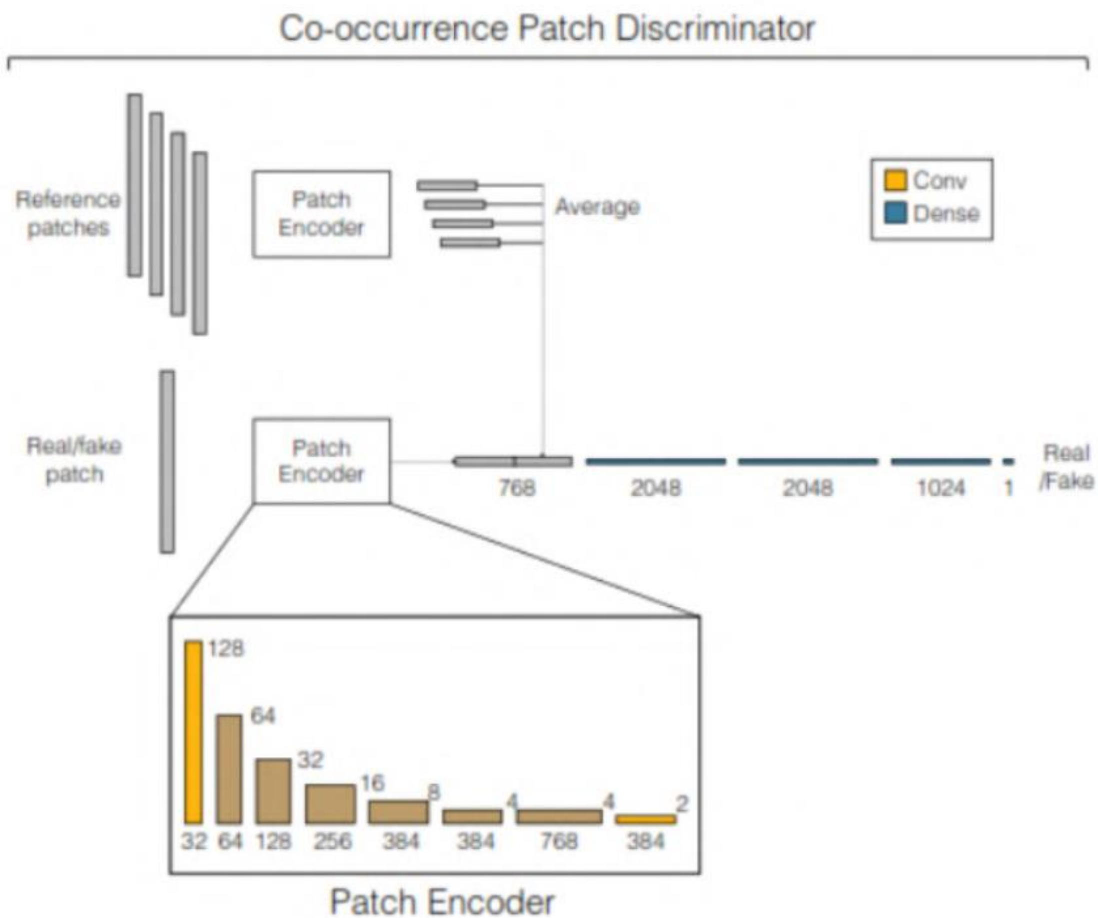



**Figure 4.5:** Mode collapse study. The first row reports the original image, while the second row represents the corresponding generated image. We can see that although all the images share the same style and texture, only the image at the top in the red box is reconstructed wrongly.

# Patch discriminator



# Encoder and generator

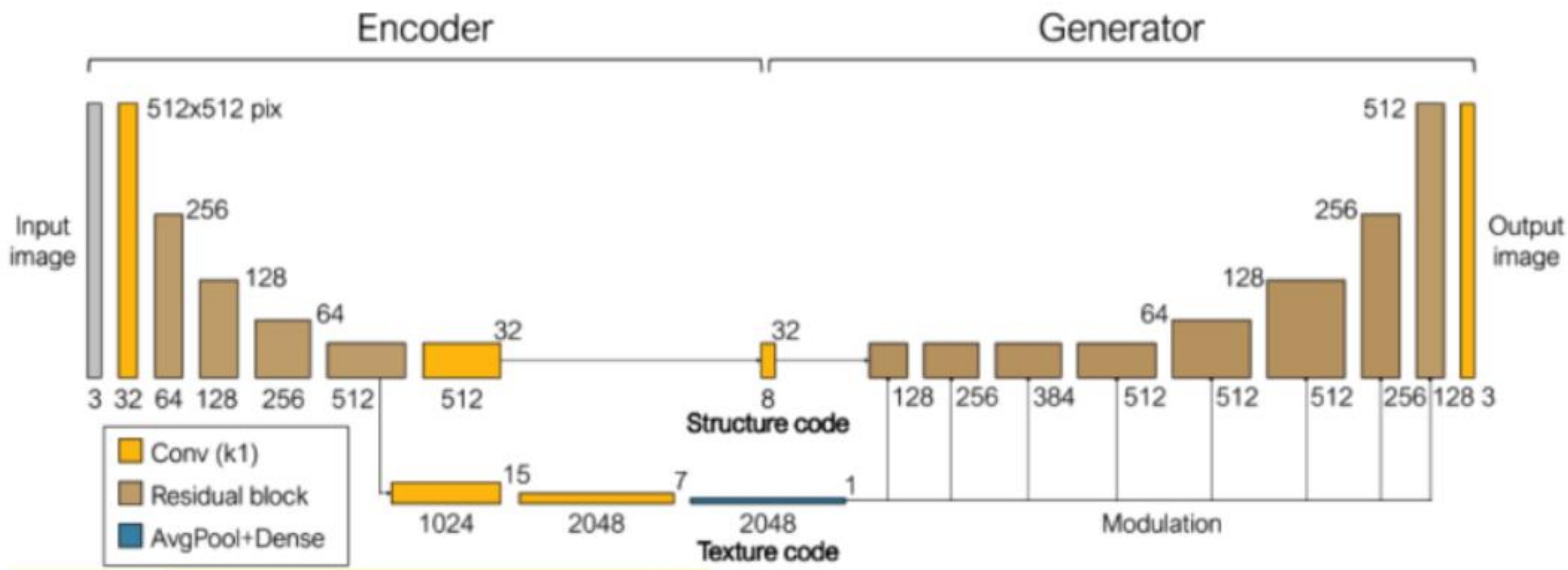



Figure 3.2: Encoder and generator

# Anchor domain hypothesis verification

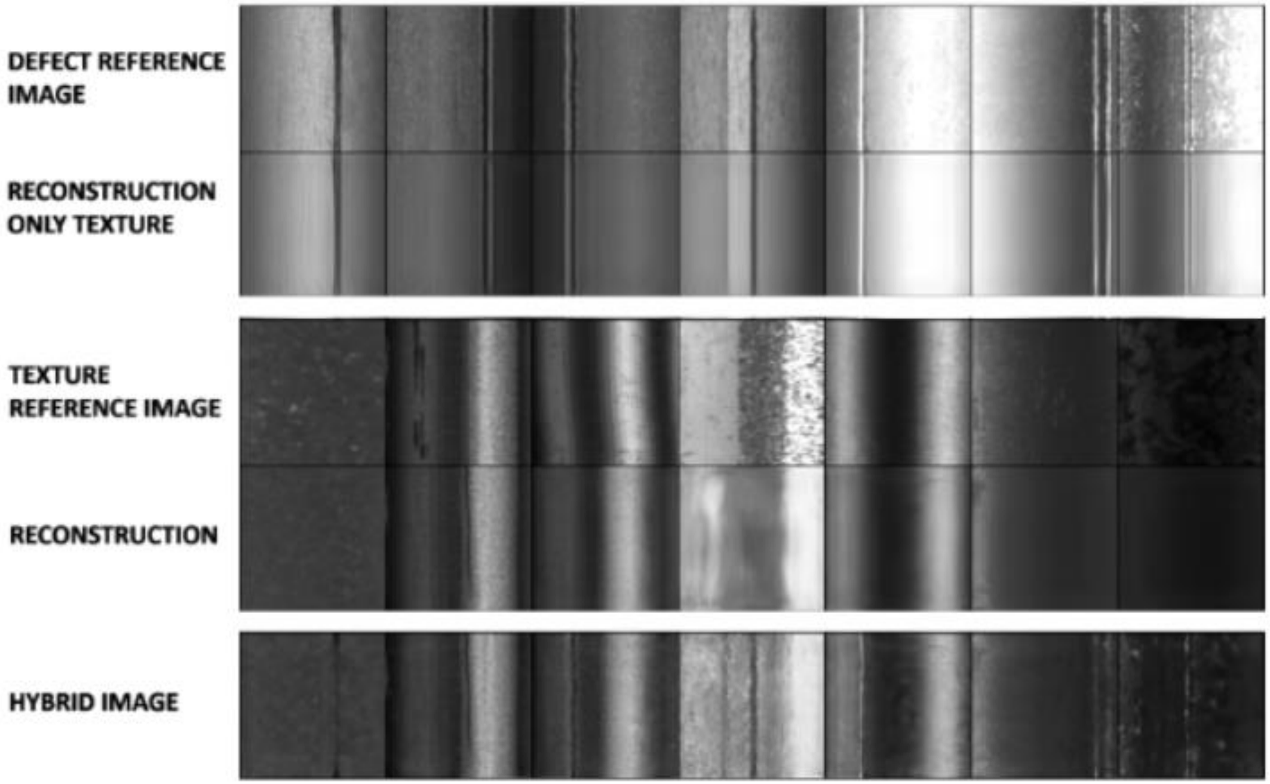



**Figure 4.16:** Texture reconstruction for the scoring defects. First and third row represents the defect and normal reference images respectively. The second and the fourth row reports the texture reconstruction for the defect and no-defect reference image respectively. The last row instead illustrates the images obtained by swapping the texture code of the defective image with the normal reference image.