Generative Adversarial Network (GAN)

EEE 511 Artificial Neural Computation
Team 17
Kevin Vora & Maitry Trivedi

23rd April 2020

Make a guess?

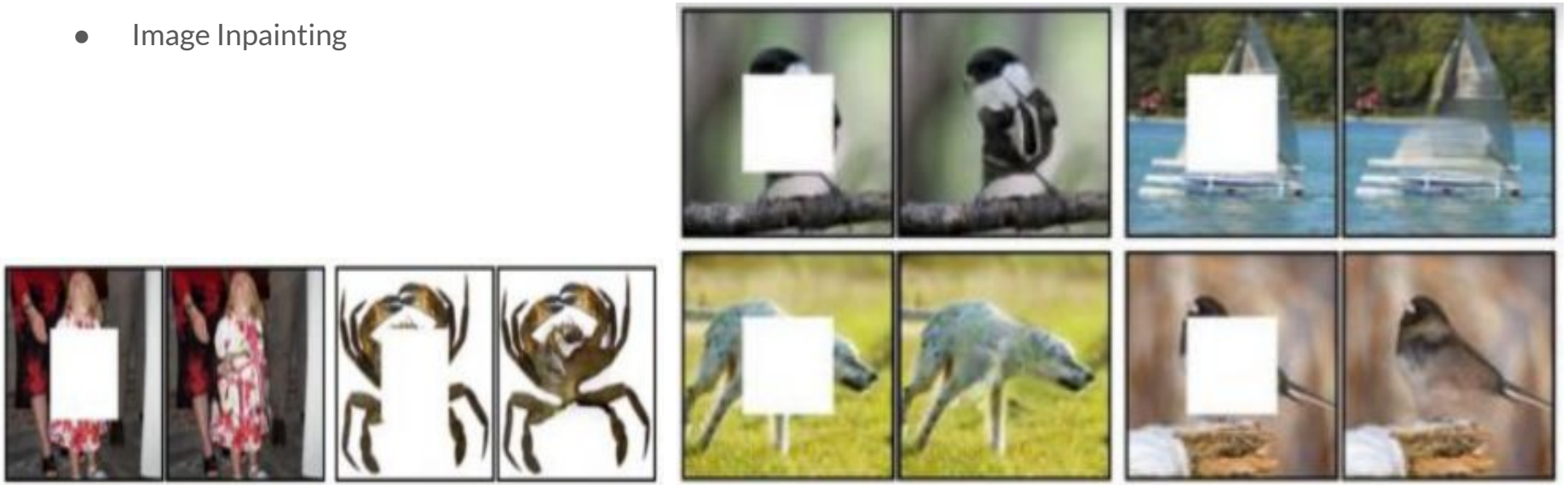
Applications of GAN

- Pose guided Image generation

(c) Generating from a sequence of poses

Applications of GAN

- Image Inpainting

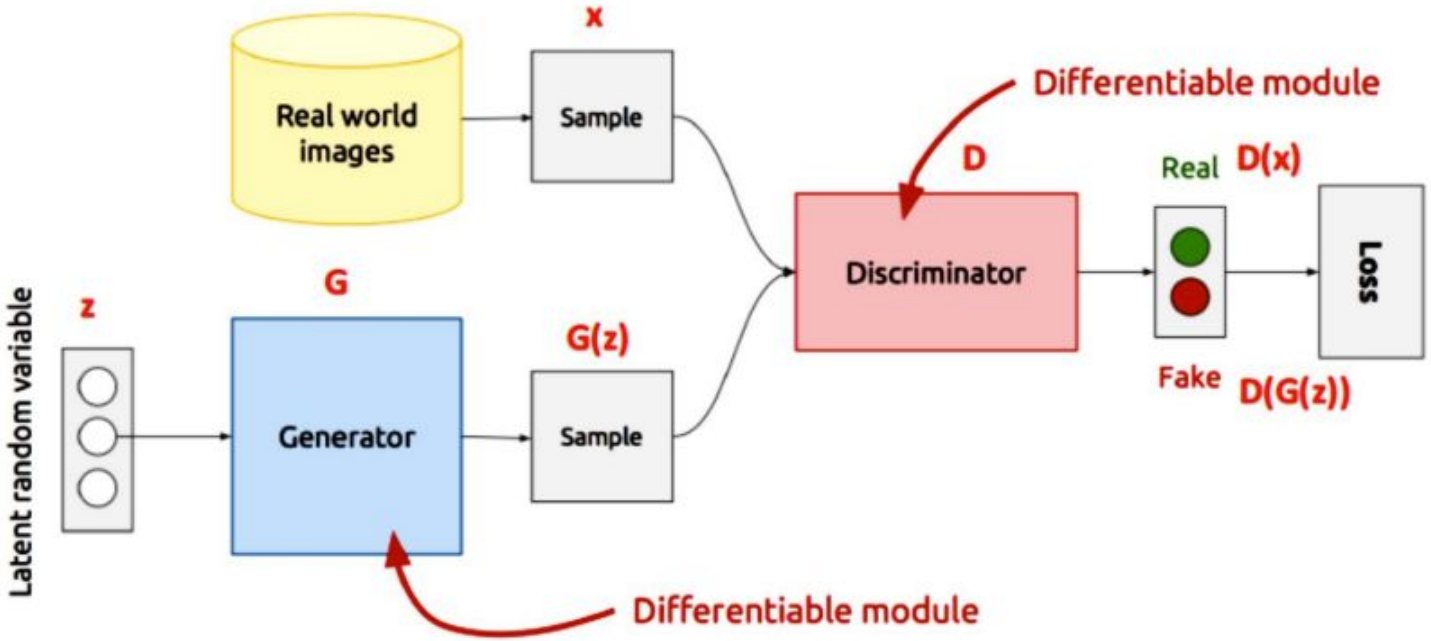


Applications of GAN

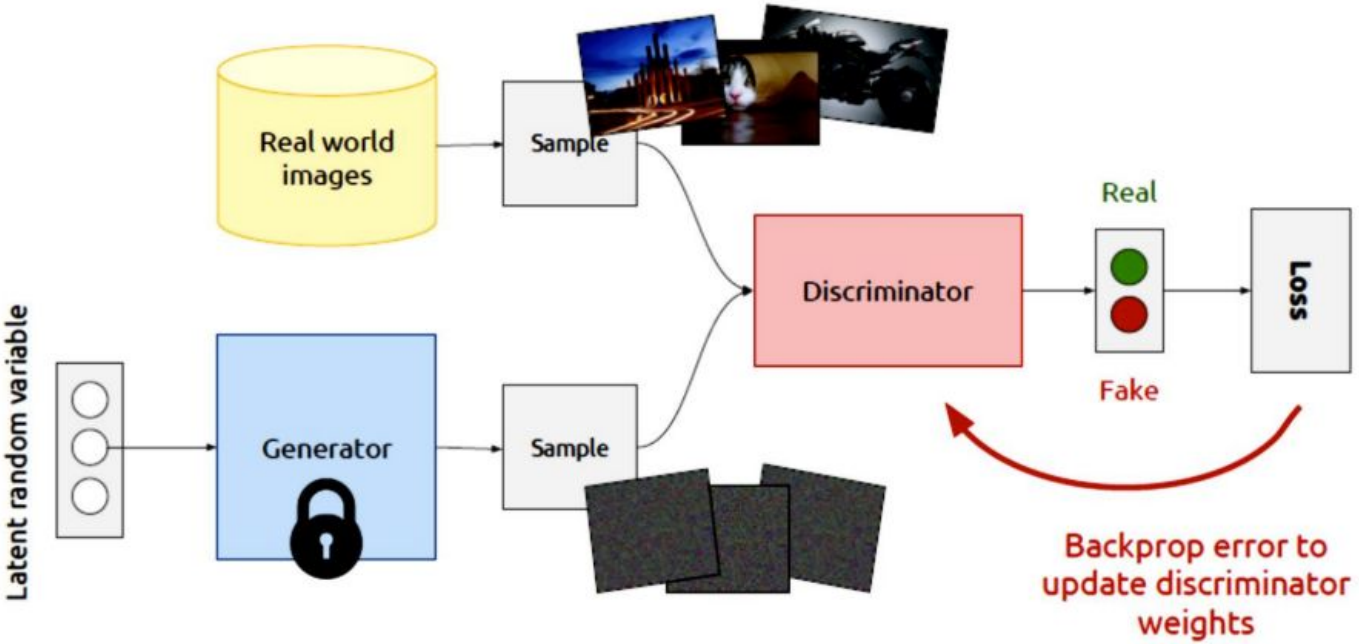
- Disco GAN

(b) Handbag images (input) & **Generated** shoe images (output)

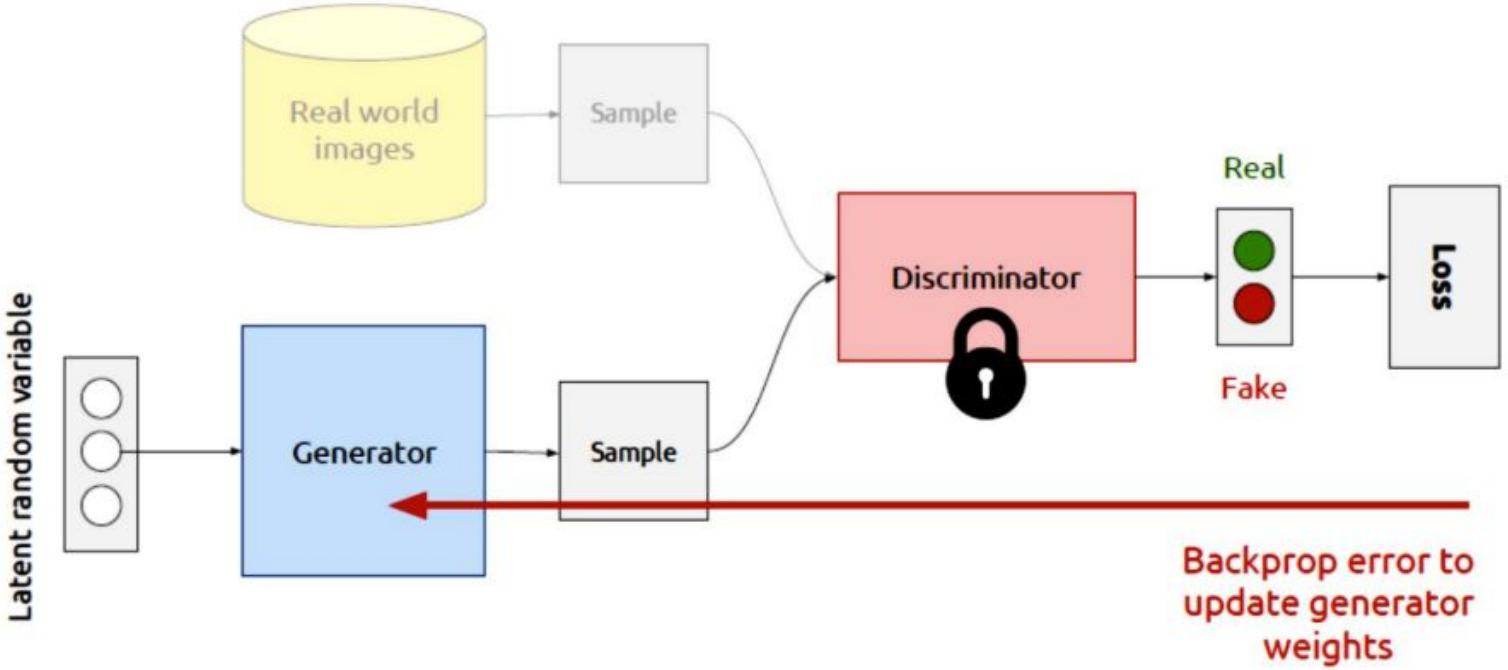
GAN Architecture



Training Discriminator



Training generator

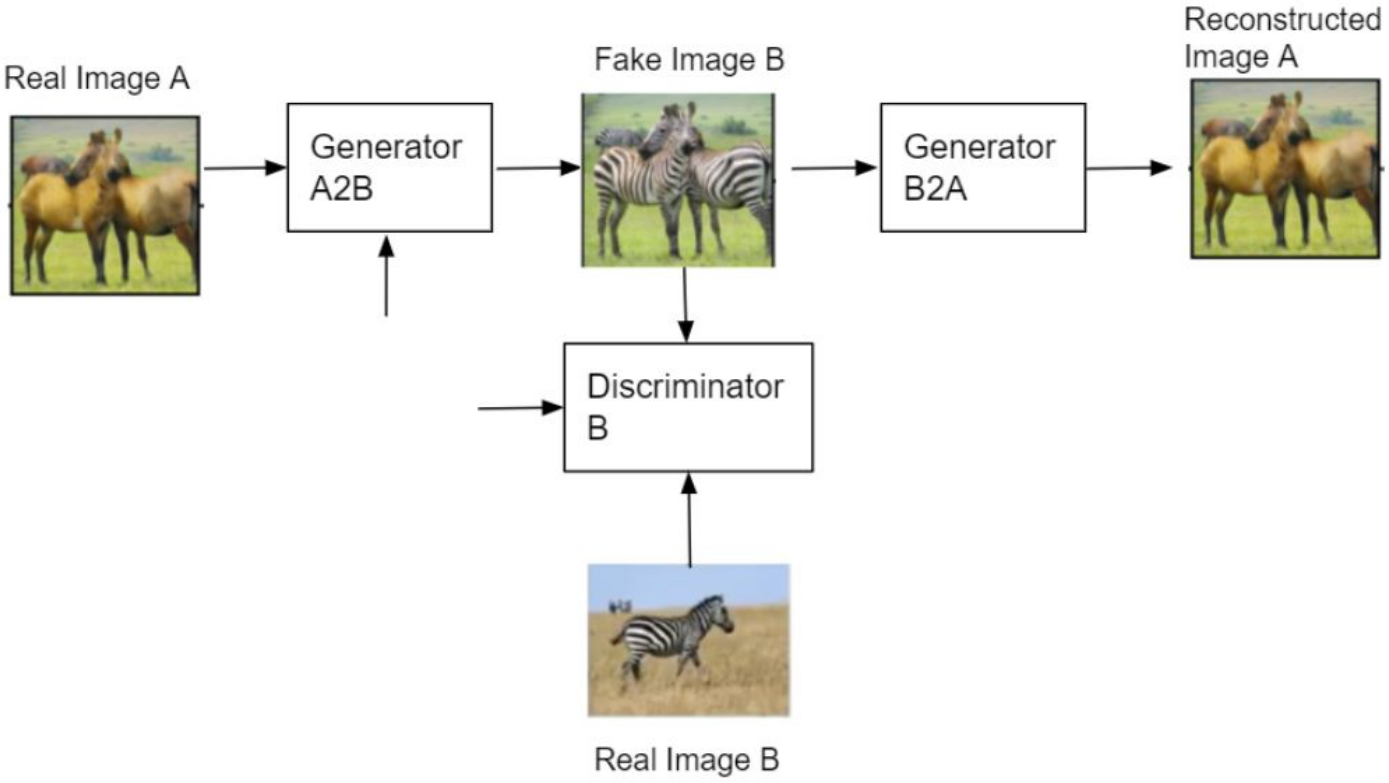


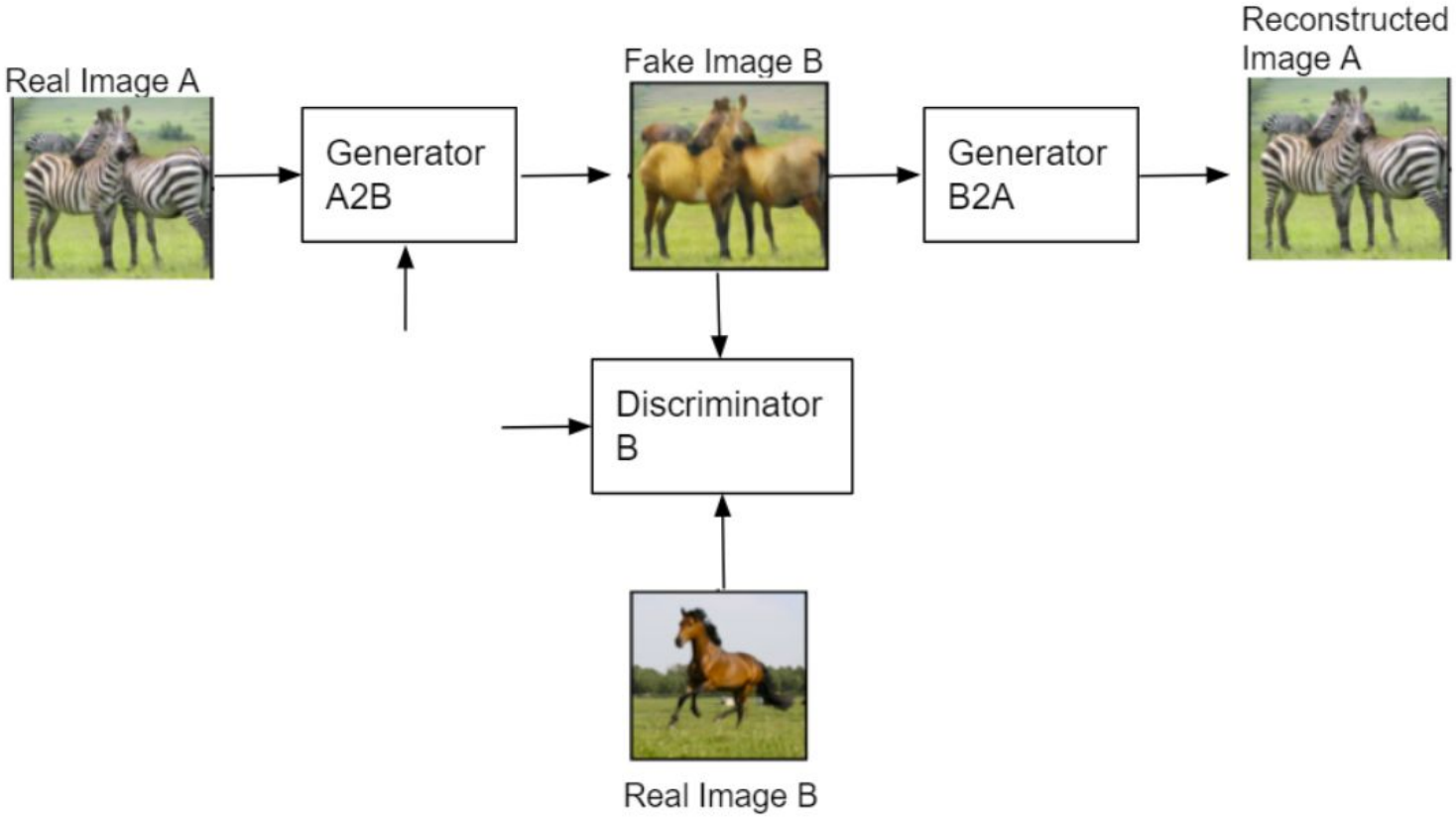
DiscoGAN

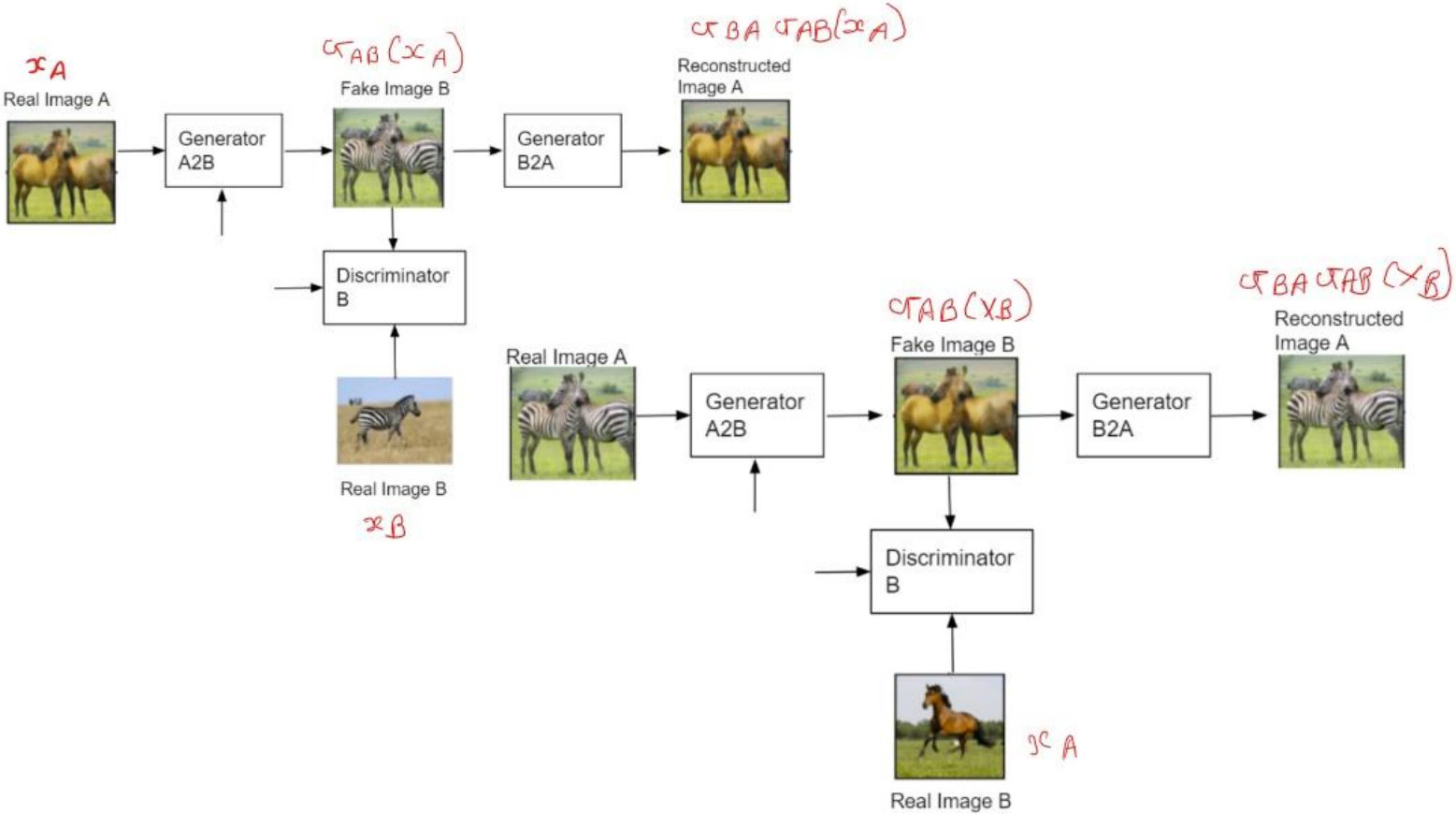
Learning to Discover Cross-Domain Relations with Generative Adversarial Networks

Problem Statement

- Our primary aim is to train a GAN model which can find the relationship between two different domains (Learning to Discover Cross-Domain Relations with Generative Adversarial Networks).
- For example, suggesting pair of shoes for one particular dress.


Real Image A
Generator A2B
Fake Image B
Generator B2A
Reconstructed Image A
Discriminator B
Real Image B


Real Image A
Generator A2B
Fake Image B
Generator B2A
Reconstructed Image A
Discriminator B
Real Image B


Real Image A
Generator A2B
Fake Image B
Generator B2A
Reconstructed Image A
Discriminator B
Real Image B
Real Image A
Generator A2B
Fake Image B
Generator B2A
Reconstructed Image A
Discriminator B
Real Image B

Cost Function of Discriminator A

$$C\_D_A = - \mathop{\mathbb{E}}_{x_A \sim P(x_A)} [\log(D_B(x_A))] - \mathop{\mathbb{E}}_{x_B \sim P(x_B)} [\log(1 - G_{BA}(x_B))]$$

Cost Function of Discriminator B

$$C\_D_B = - \mathop{\mathbb{E}}_{x_B \sim P(x_B)} [\log(D_B(x_B))] - \mathop{\mathbb{E}}_{x_A \sim P(x_A)} [\log(1 - G_{AB}(x_A))]$$

Cost Function of Generator AB

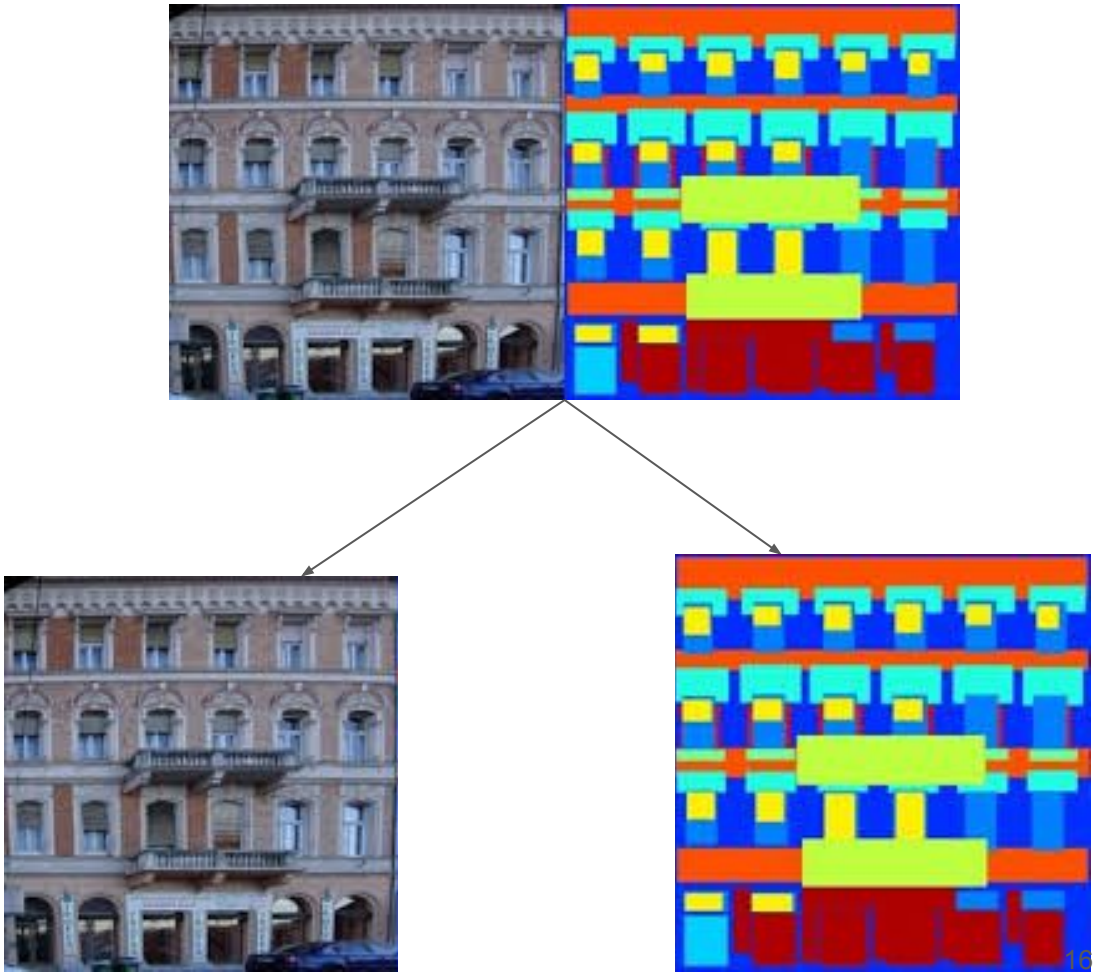
$$C\_G_{AB} = C_{reconst(ABA)} + C_{D(AB)}$$
$$= \|x_A - G_{BA}G_{AB}(x_A)\|_2^2 - \log(D_B(G_{AB}(x_A)))$$

Cost Function of Generator BA

$$C\_G_{BA} = C_{reconst(BAB)} + C_{D(BA)} = \|x_B - G_{AB}G_{BA}(x_B)\|_2^2 - \log(D_A(G_{BA}(x_B)))$$

Dataset: Facades

- Training: 399
- Testing: 106
- Validation: 100

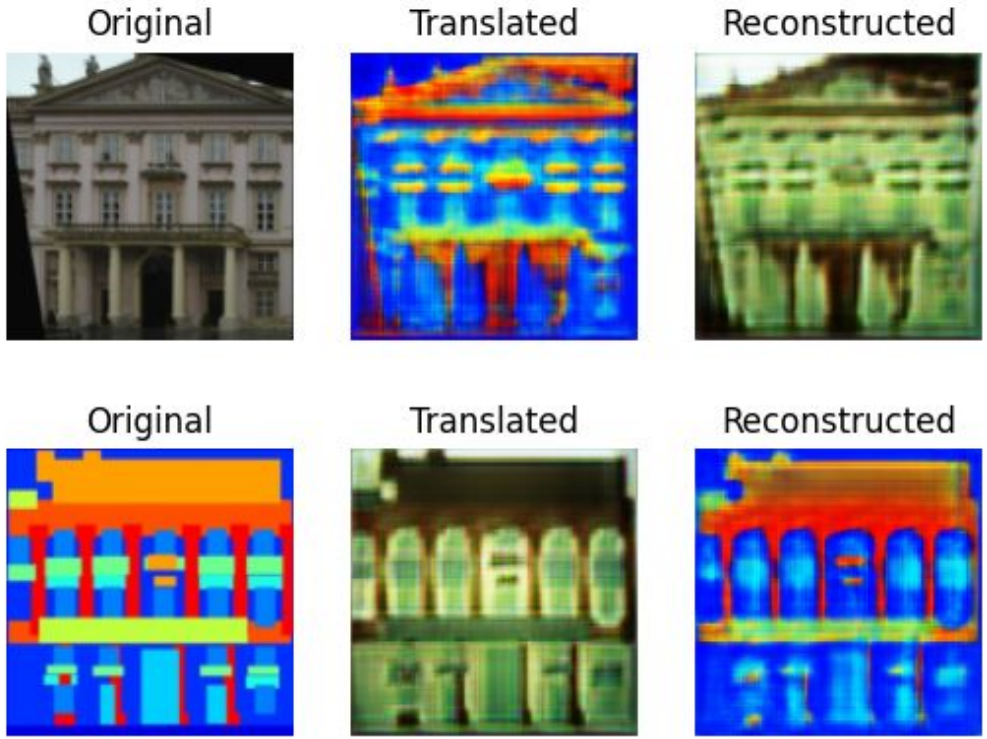


Implementation details

- Resized images to 128 x 128 pixels after cropping
- Filter size 32 x 32
- Keras based implementation:
  - Optimizer: Adam
  - Loss: MSE
  - Batch size: 1
  - Epoch: 15
- CoLab: 11 hours
  - https://colab.research.google.com/drive/1-N6vA84EOZdWfG8hHoFIps1zPHW1-dS8

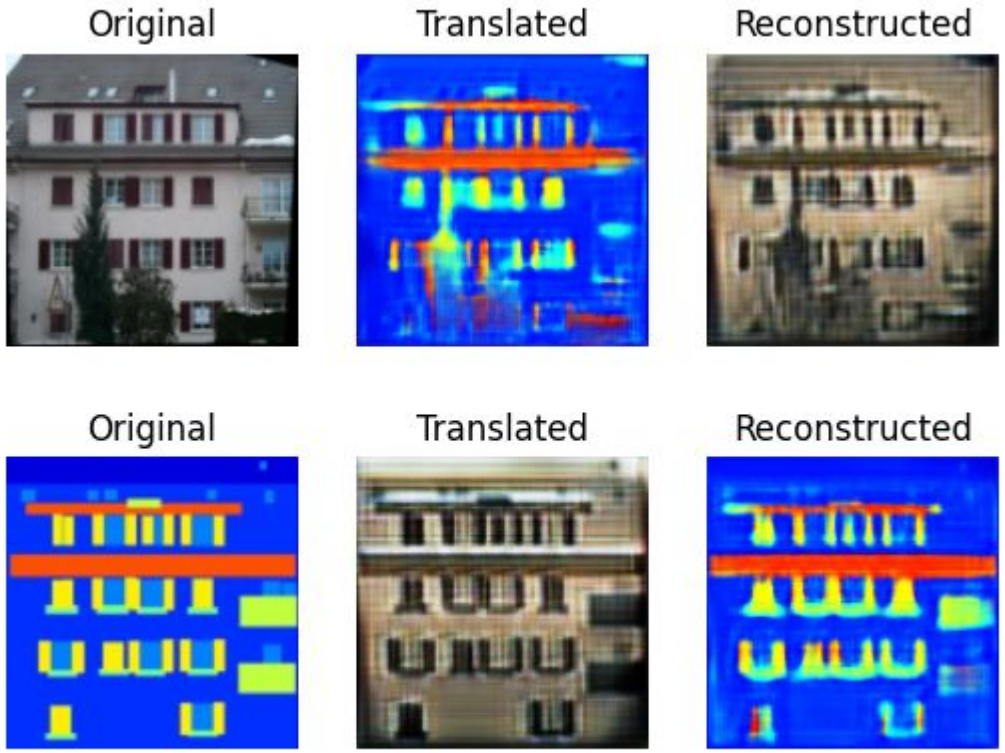
Results

- Half dataset
- 16x16 Filters



Results

- Full Dataset
- 32x32 Filters



Conclusion

- DiscoGAN is implemented to get the segmented image and it has also learned to reconstruct the image properly.
- GANs are really good at transferring cross domain characteristics.
- GANs need a lot of data and time to train
- We have come this far and with more resources we can do even better!

Questions?

References

- https://colab.research.google.com/drive/1-N6vA84EOZdWfG8hHoFIps1zPHW1-dS8
- https://medium.com/towards-artificial-intelligence/generating-matching-bags-from-shoe-images-and-vice-versa-using-discogans-8149e2cbc02
- https://github.com/eriklindernoren/Keras-GAN