Generating Colored Images from Black and White Images [using *pix2pix*]

Team Members

Rajat Rathi	160050015
Anmol Singh	160050107
Gurparkash Singh	160050112

Aim

- The aim of our project is to create a Neural Network model that can automatically generate colored images from corresponding black and white images.

- For this purpose, we have implemented the paper "*Image-to-Image Translation with Conditional Adversarial Networks*" published in November 2017 by Berkeley AI Research Lab.

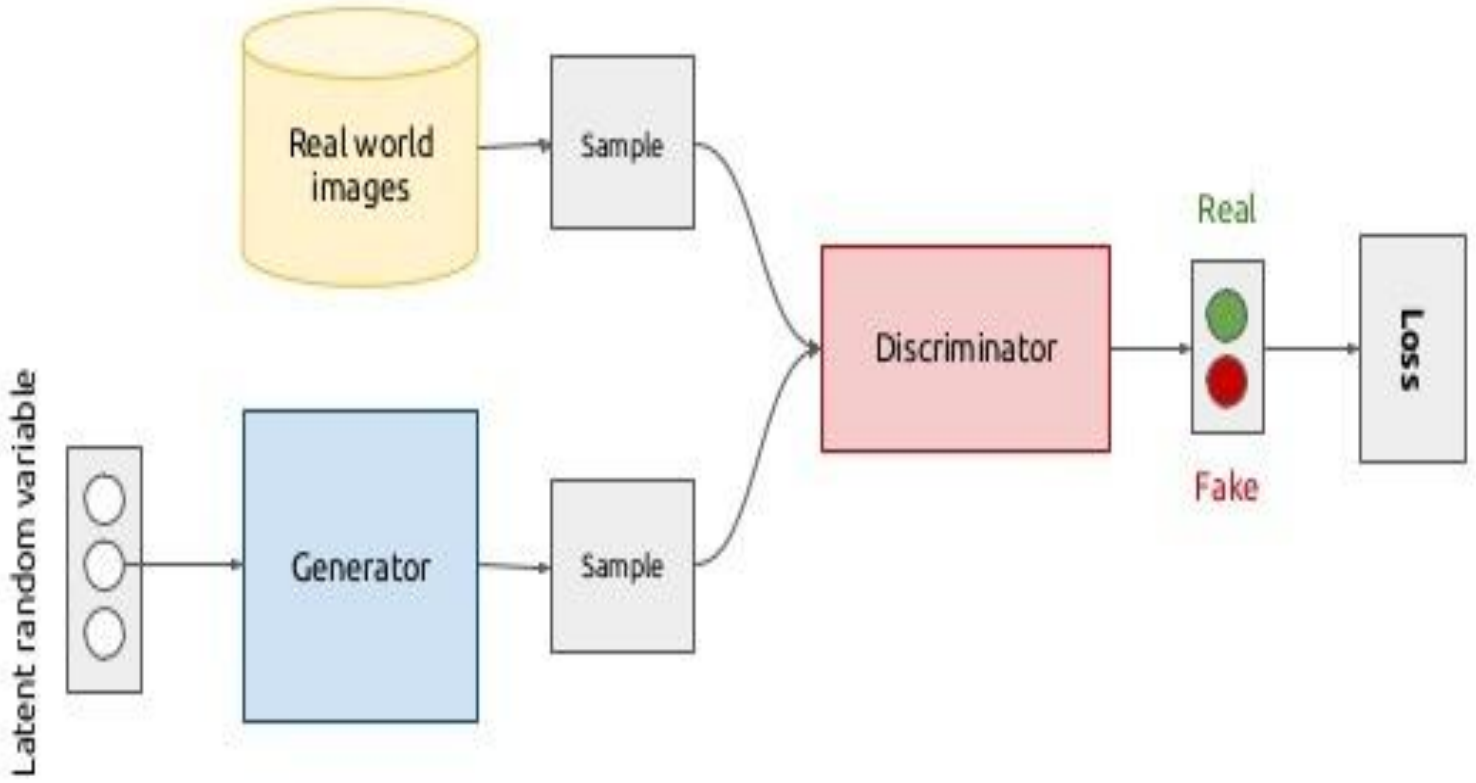
- In the paper, the authors have designed a Generative Adversarial Network (GAN) for Domain Transfer tasks such as Edges to Photos, Day to Night, Summer to Winter, Black and White to Color etc.

GAN Overview

- GAN can be thought of as the combination of a counterfeiter and a cop in a game of cat and mouse, where the counterfeiter is learning to pass false notes, and the cop is learning to detect them. Both are dynamic; i.e. the cop is in training, too and each side comes to learn the other's methods in a constant escalation.

- With more training, the discriminator gets better at distinguishing between real and fake samples whereas the generator gets better at generating real looking samples.

- One of the primary challenges in training a GAN is that since the discriminator has a relatively easier task, it becomes too strong too quickly which causes the generator to stop learning too.


Real world images
Sample
Real
Discriminator
Loss
Latent random variable
Generator
Sample
Fake

Loss Functions

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))],$$

CONDITIONAL GAN LOSS

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1].$$

L1 LOSS

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

FINAL LOSS FUNCTION

Results

INPUT

OUTPUT

TARGET

Results (contd.)

INPUT

OUTPUT

TARGET

Results (contd.)

INPUT

OUTPUT

TARGET

Results (contd.)

INPUT

OUTPUT

TARGET

Results (contd.)


INPUT


OUTPUT


TARGET

Results (contd.)

INPUT

OUTPUT

TARGET

Transition with time

View GIF in Drive
[https://docs.google.com/presentation/d/14NOG9s6Z6HCzzzb3QHYWQvooin-G1ZrVAn9ZzMy
T9-8/edit?usp=sharing]

Transition with time (contd.)

View GIF in Drive
[https://docs.google.com/presentation/d/14NOG9s6Z6HCzzzb3QHYWQvooin-G1ZrVAn9ZzMyT9-8/edit?usp=sharing]

Experimentation - Inversion of Images

Experimentation - Effect of Lambda

Lambda=0.01

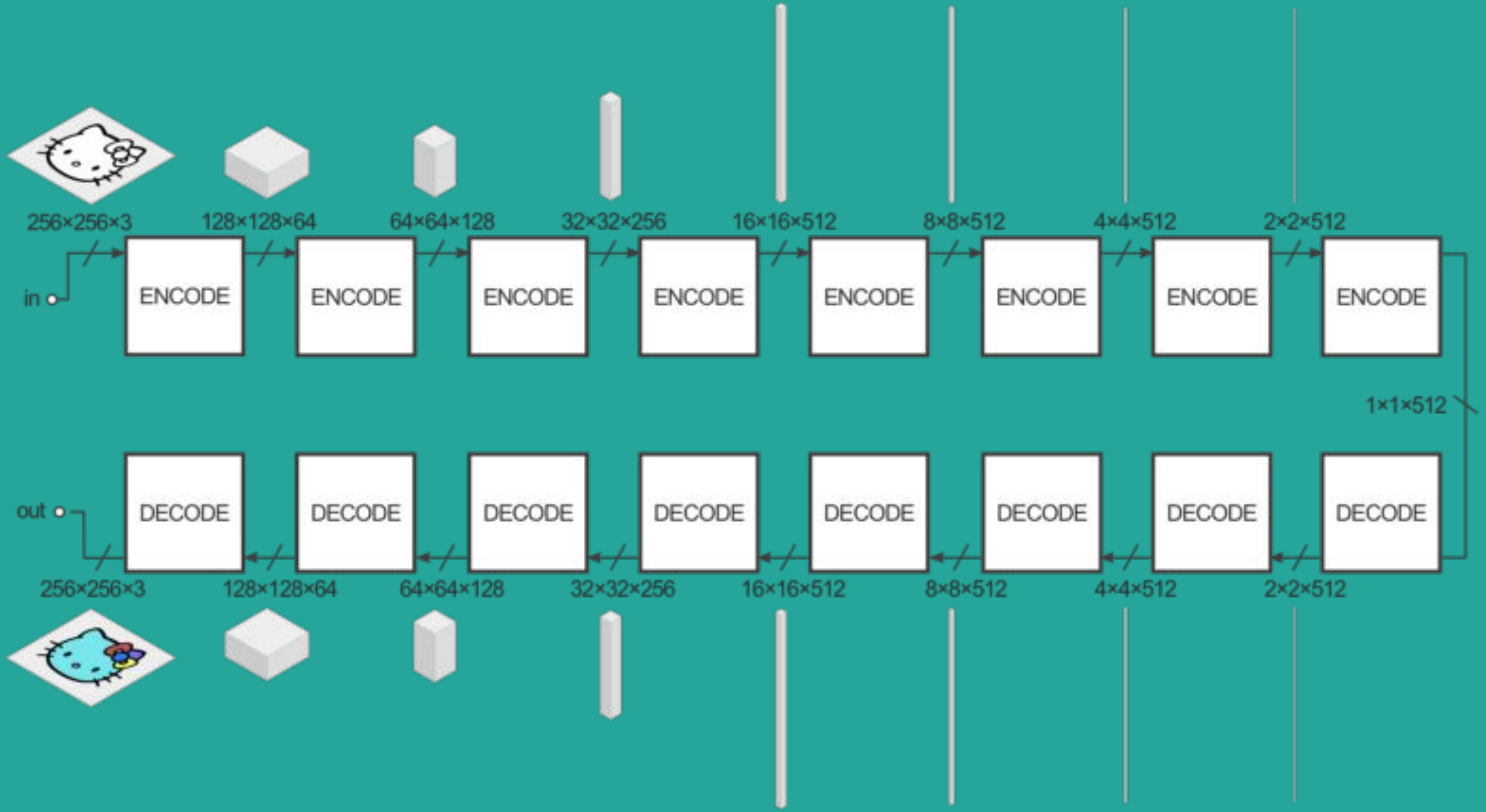
Lambda=20.0

Lambda=200.0

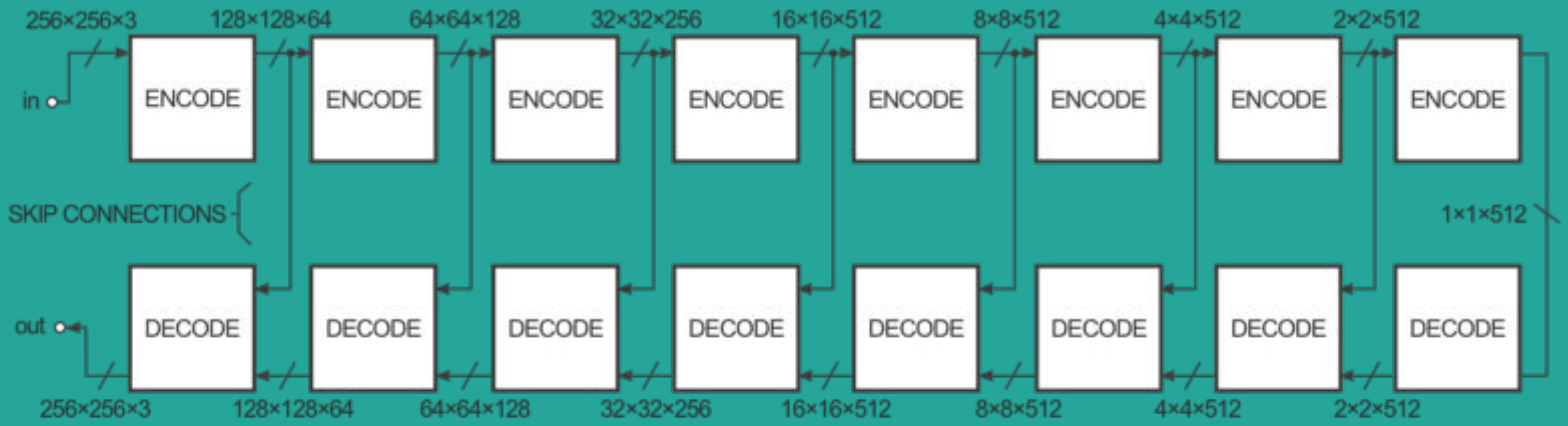
Other Experimentations / Tuning

- Some other experiments that helped :)
  - Using U-Net Generator Network instead of simple Encoder-Decoder Network.
  - Data Augmentation by randomly cropping and inverting images (results shown above)
  - Using Fuzzy target labels!! (Most helpful)
  - Having Batch Normalization and dropout in Discriminator and Decoder layers of Generator.

- And some that didn’t :(
  - Training one of discriminator/generator more per epoch.
  - Training generator for higher number of epochs, keeping discriminator fixed.
  - Having higher learning rate for generator.



ENCODER-DECODER NETWORK



U-NET GENERATOR

Evaluation Metrics

- Open Country Dataset
  - Avg. SSIM : 0.8792889071250597
  - Avg. RMSD : 0.6651818876595532

- Forests Dataset
  - Avg. SSIM : 0.8884571322784819
  - Avg. RMSD : 0.6913058699378285

Performance on other tasks

- Facades
- Cityscape

Conclusion: We need to do hyperparameter tuning again for different tasks!

References

- Image-to-Image Translation with Conditional Adversarial Networks
  [https://arxiv.org/pdf/1611.07004.pdf]

- Original Generative Adversarial Networks Paper
  [https://arxiv.org/pdf/1406.2661.pdf]

- MIT-CVCL Images Dataset
  [http://cvcl.mit.edu/database.htm]

- Berkeley AI Research pix2pix Dataset
  [http://efrosgans.eecs.berkeley.edu/pix2pix/datasets/]