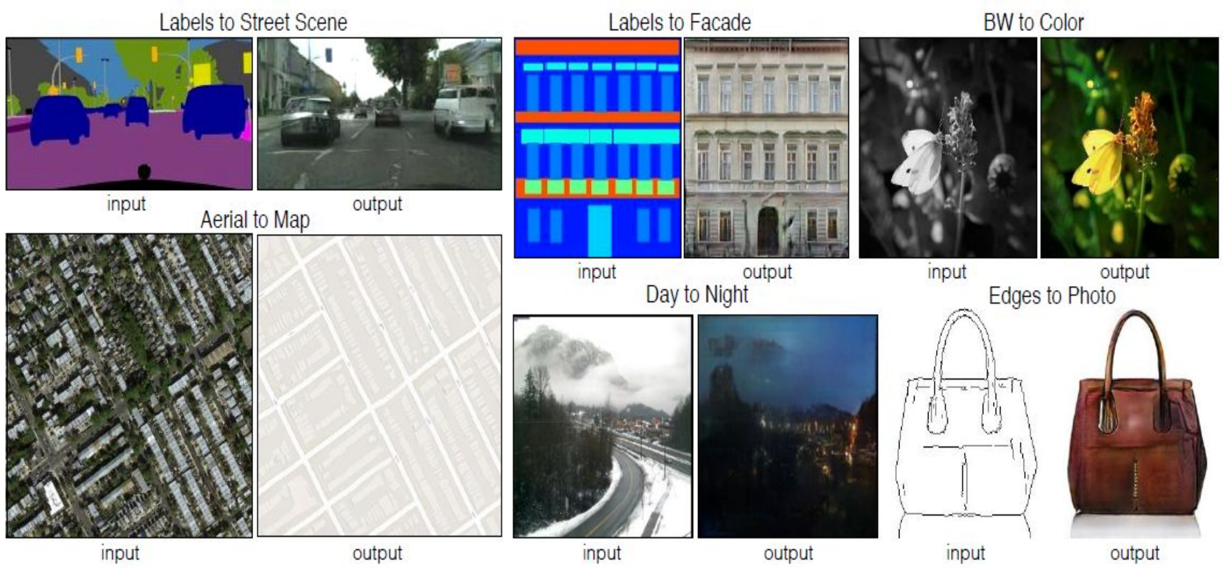
Project: Update Report I

Image to Image translation using cGANs

Mentor: Dr. Pawan Kumar

Aman Krishna - 2018201070

Gyanshu Azad Singh - 2018201073



Introduction:

The paper talks about a general approach to image-to-image translation using Conditional Generative adversarial Networks. In this report we have discussed our understanding of the paper. It's approach, the architecture used and the loss function. Following it we have mentioned what is the next thing we will be working on in the future. The report includes:

- cGAN vs GAN
- Objective Function
- Architecture of the Network
- Future work

cGANs (Conditional GANs) vs GAN

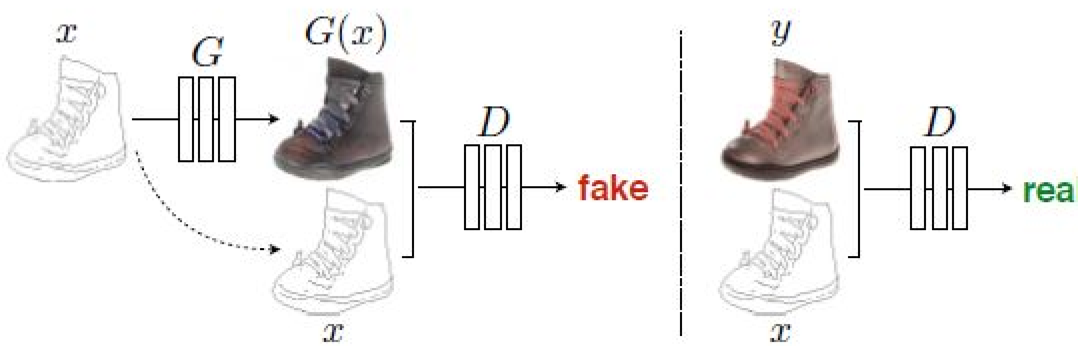
- In traditional GANs the generative model learn a mapping from random noise vector ***z*** to output image ***y***

G: *z* → *y*

- Conditions GANs learn a mapping from an observed input image ***x*** and random noise vector ***z*** to produce output image ***y***

G: { *x, z* } → *y*

- Unlike traditional GANs in cGANs both the discriminator also observes the input the image during Fake/No Fake decision



Objective Function

- The objective of a conditional GAN can be expressed as

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

Here, G tries to minimize the objective function while adversarial D tries to maximize it

$$G^* = \min_G \max_D L_{cGAN}(G,D)$$

- Since the task of the Generator is not only to fool the Discriminator but also to be near the ground truth a L1 loss is also added

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

- Hence the final objective function is

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

- In the final model the noise is provided in the form of dropouts applied on several layers of the Generator at both training and test time

Network Architecture

- Like any GAN the Conditional Adversarial Network consists primarily of two parts
    - Generator
    - Discriminator

Generator

- Traditionally an Encoder-Decoder model is used for Generators. However, for many image translation problems there is a great deal of low level information shared between the input and output
- Hence it is desirable to shuttle this information directly across the network
- Therefore, the paper uses a U-Net

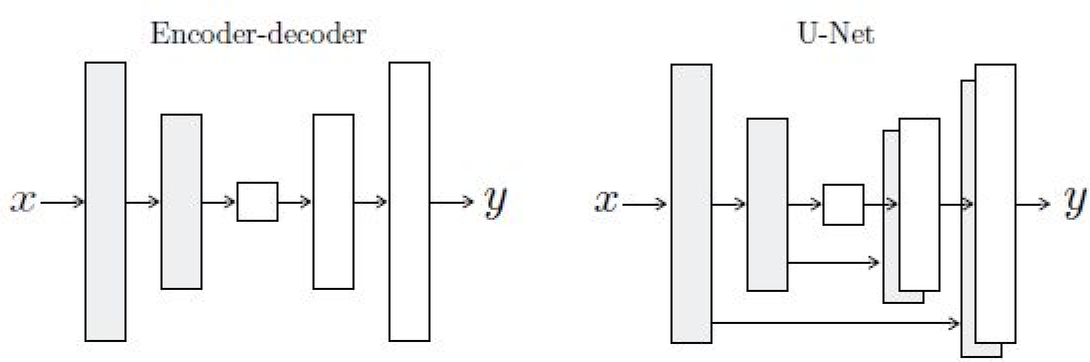


Figure 3: Two choices for the architecture of the generator. The "U-Net" [50] is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

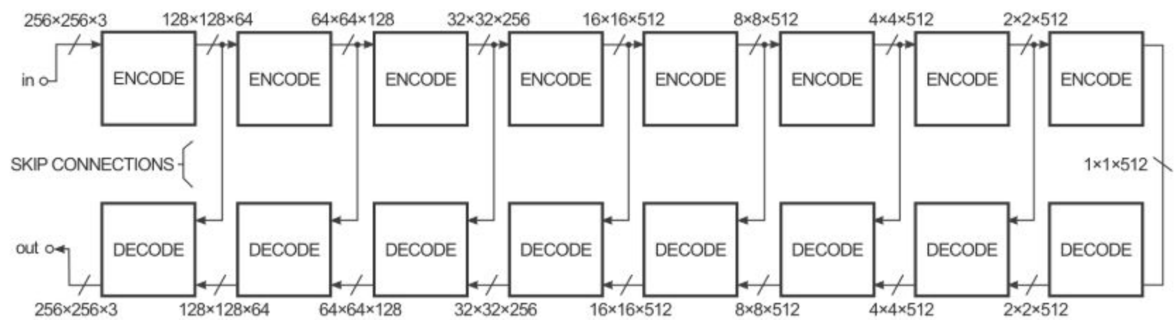
Discriminator

- The L1 loss introduced in the objective function accurately capture the low frequencies (low pixel intensity variation) in the output image
- Therefore, the work of the Discriminator is limited to model the high-frequency structure
- The paper uses a PatchGAN which penalizes structure at the scale of patches
- This discriminator tries to classify if each **NxN** patch in an image is real or fake
- The Discriminator is run convolutionally across the image, averaging all responses to provide the ultimate output of D
- Size of **N** can be varied

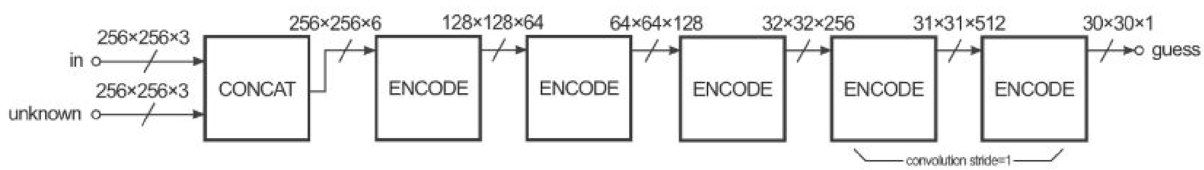
Output for different values of N

Block Diagram of Generator and Discriminator

- Following is the block diagram of the generator



- Following is the block diagram of the discriminator



Future Work:

- Though we know the structure of the model we are still in the dark about how the actual model is built
- Things we do not know:
    - How the dropouts introduce the random z vector
    - How to build (code) a simple generator and discriminator and train them
- Therefore, for the next step we will be building a simple GAN network and then use it to have a better understanding of the model given in the paper