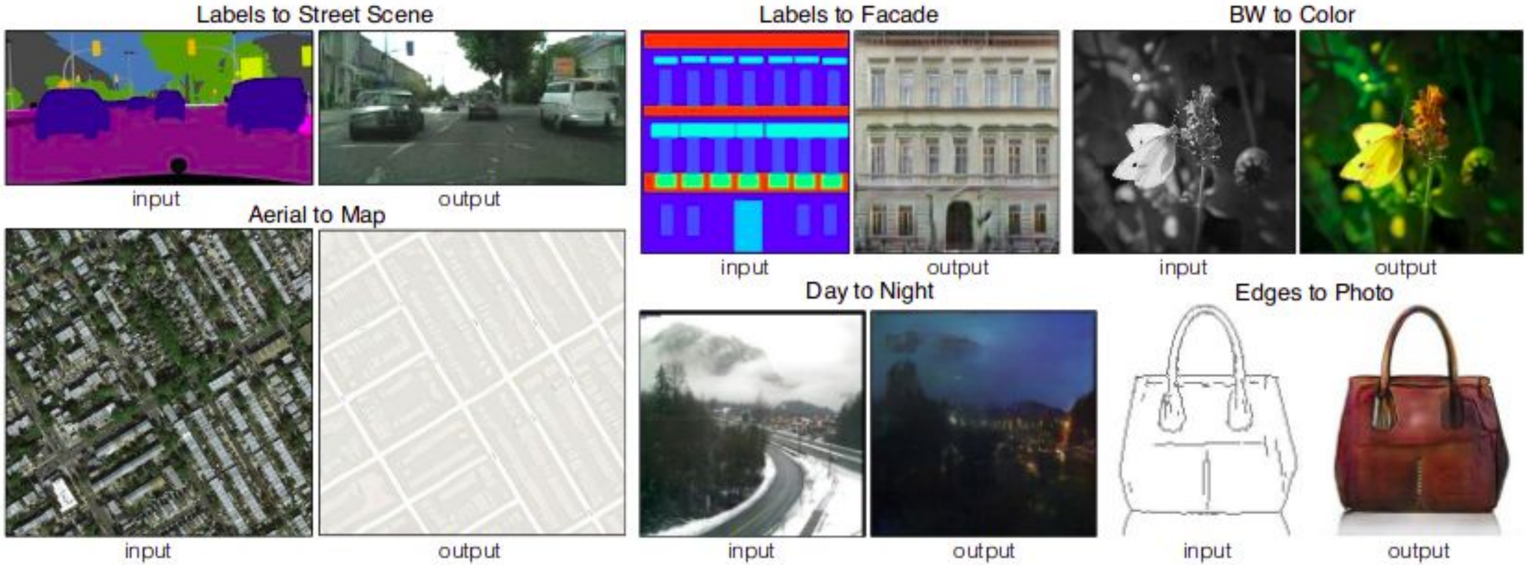
Image-To-Image Translation using Deep Learning



Outline

- INTRODUCTION
- INSTITUTE PROFILE
- WORK
- Model
- Dataset
- Requirements
- Training and Testing
- OUTCOME
- CERTIFICATE SCAN COPY
- CONCLUSION
- REFERENCES

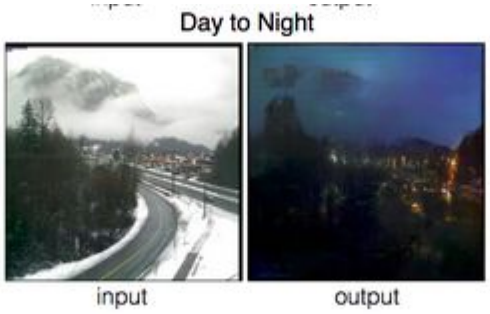
Abstract

- In Image-to-Image Translation, the task is to convert an input image into another corresponding image, like translating black and white scene to color one, labelled image to synthesized street or façade image, restoring objects photo from edge map, etc.

- Used the concept of pix2pix based on the structure of CGANs, which learns a structured loss that strives to classify the output image as real or fake one, while simultaneously training a conditional generative model to minimize this loss.

Introduction

- Image-To-Image Translation can be thought of as a problem which requires image processing, deep learning and computer vision.
- I presented here a simple framework of “Coverting Daylight image into Night image and vice versa“ which achieved a reasonable results, and visualized the effects of several foremost architectural alternatives.



Institute Profile

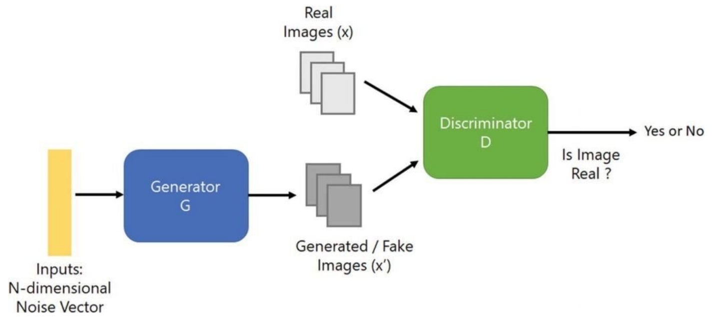
- **Indian Institute of Information Technology, Allahabad** (**IIIT-Allahabad**) is a public university located in Allahabad, Jhalwa in Uttar Pradesh state in northern India.
- The Institute has engaged students under a wide range of heads and disciplines within the Information and related technology domain ranging from pure **IT fields** such as -Soft Computing, artificial intelligence, Graphics animation and multimedia, Data Mining, Wireless and Sensory Networks, Service Oriented Architectures to **Electronics and Communication Engineering** topics such as VLSI Fiber Optics, Photonics, Robotics, Computer vision and Embedded systems.

Work

1. Model
2. Dataset
3. Requirements
4. Training and Testing

Model

- The image-conditional model is based on the architecture of generator and discriminator.



- For the generator, I used a “U-Net“- based architecture, and for the discriminator, I used a convolutional “ PatchGAN “ classifier.
- The aim of CGANs can be denoted as:

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

- I used the concept of L1 to get the image less blurry:

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

- For the final model, I provided noise merely in the form of dropout, applied on several layers of the generator nets at both training and test time.
- Both generator and discriminator uses modules of the form of ‘convolution-BatchNorm- ReLu’ .
- For optimization of Discriminator D, I alternated between one gradient descent step on D, and then one step on Generator G. Here training is done to maximize log(D(x,G(x,z)))

Dataset

- Training have been done on around 20K natural scene images from ‘Transient Attributes Dataset’. The day to night training set consists of images from 91 webcam videos.
- Here I have also used the concept of transfer learning where I used the pre-trained model “day2night”, trained from 101 webcams, containing 8521 images all annotated with 40 attribute labels.
- I have also trained on my own datasets which includes images of outdoor scenes recorded from my accommodation using 1 Logitech Webcam (having maximum resolution of 30 fps) saving 96 images having resolution as 640x480.
- This dataset was distributed uniformly amongst three classes namely- train, test, and validation.

Requirements

- Linux
- CPU or NVIDIA GPU + CUDA CuDNN
- torch >= 0.4.1
- torchvision >= 0.2.1
- dominate >= 2.3.1
- opencv >= 3.4.1
- Pillow == 5.0.0
- numpy == 1.14.1
- scipy==1.1.0

Training and Testing

- Training is based on the concept of transfer learning, where the weights were initialized from a Gaussian distribution with mean 0 and standard deviation 0.02.
- I have used 60% of total data for training and 40% of total data for testing. Every time two images corresponding to the same scene have to be taken during day and night.
- I used batch normalization for better results.
- Size of batch: 1, Learning rate: 0.002, Starting epoch count: 1 and Total no. of epochs: 400.

Outcome

- I got the following results when I tested my model on my test dataset:

Daylight Image:

Night Image:

Output Image:

- When I trained my model for 200 epochs, it resulted in too much blurry image. But on increasing the total no. of epochs and varying the learning rate to 0.003, I got slightly better image.

Daylight Image:

Night Image:

Output Image:

- But when I tested the images using the pretrained model ‘day2night_pretrained’ , the daylight images seemed to be converted to night images pretty nicely.

Daylight Image:

Night Image:

Output Image:

Certificate Scan Copy

भारतीय सूचना प्रौद्योगिकी संस्थान, इलाहाबाद

Indian Institute of Information Technology, Allahabad

An Institute of National Importance by Act of Parliament

Deoghat, Jhalwa, Allahabad-211015 (U.P.) INDIA

Ph.: 0532-2922025, 2922067, Fax : 0532-2430006, Web : www.iiita.ac.in, E-mail : contact@iiita.ac.in

F. No. : IIIT-A /HOD-IT/ 1927 /dated 24/7/19

To whom it may concern

This is to certify that **Mr. Ashutosh Kumar Verma** S/O **Mr. Sarju Prasad Verma has** successfully completed Summer Internship Program from **14-05-19 to 12-07-19** at the **Department of Information Technology.**

We wish him all the best for his future endeavours.

(Dr. Rahul Kala)
Mentor

12/7/19
(Dr. Vrijendra Singh)
HoD-IT

Indian Institute of Information Technology Allahabad
Head
Department of
Information Technology
Prayagraj

Conclusion

- The pix2pix architecture proved to be very useful in converting daylight images to night images or vice-versa, as it excelled other architectures on such real life image-to-image translation tasks.
- It can be used in a lot of applications like in self driving vehicles which requires recorded videos to be converted to another domain, in colourization and super-resolution, and for studying novel image synthesis.

References

[1] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts.arXiv preprint arXiv:1612.00215, 2016.

[2] Phillip Isola , Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004v3 [cs.CV]