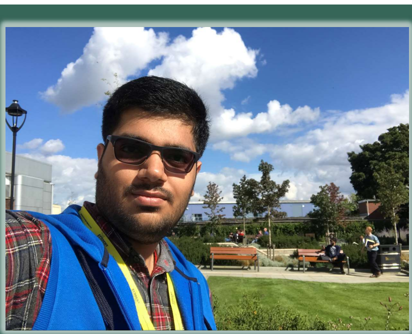
UNDERSTANDING AND IMPLEMENTING GENERATIVE ADVERSARIAL NETWORKS (GANs): ONE OF THE BIGGEST BREAKTHROUGHS IN THE DEEP LEARNING REVOLUTION

GeoPython, Basel, Switzerland | 2019 { **Tweet: #GeoPythonConf @greatdevaks** }



Anmol Krishan Sachdeva
Platform Software Engineer - Bigbasket, India
MSc Advanced Computing, University of Bristol, UK
Email Id: greatdevaks@gmail.com
LinkedIn: https://www.linkedin.com/in/greatdevaks
Twitter: https://www.twitter.com/greatdevaks
GitHub: https://www.github.com/greatdevaks

1

About Me

- Currently, working as a Platform Software Engineer at Bigbasket, India (India's largest online food and grocery store).
- MSc in Advanced Computing (Machine Learning, Artificial Intelligence, Robotics, Cloud Computing, and Computational Neuroscience) from University of Bristol, United Kingdom.
- International Tech Speaker (spoke at PyCon Thailand 2019, GeoPython 2018, EuroPython 2018, and many more).
- Former Software Developer Intern at IBM & an ALL STACK DEVELOPER capable of designing and developing solutions for Mobile, Web, Embedded Systems, and Desktop.
- Represented India at International Hackathons like Hack Junction'16, Finland and Hack the North'16, Canada. Got invited for numerous prestigious International Hackathons (PennApps'17, HackNY'17, Hack Princeton'17, and many more) and Conferences.
- Microsoft Certified Professional, Microsoft Technology Associate, IBM Certified Web Developer, and Hewlett Packard Certified Developer.
- 8+ International Publications [Latest work got published in ACM CHI 2018. The project was exhibited in Montreal, Canada.].
- Received 6 Honours and Awards (International and National level).

2

Flow of the Talk

GeoPython 2019 Muttenz Basel Switzerland June 24-26

- A Succinct Prelude to Adversarial Training.
- Understanding Discriminative and Generative Models.
- Architecture of Generative Adversarial Networks (GANs).
- Quick Code Walkthrough (implementing DCGAN model using Python).
- Tips to Train GANs Better.
- Applications of GANs.
- Roadmap to Further Study About GANs.
- End of Talk followed by Questions and Answers Session.

20 Minutes

10 Minutes

Pre-requisites

GeoPython 2019 Muttenz Basel Switzerland June 24-26

- Good understanding of Python language and TensorFlow.
- Decent knowledge of Artificial Neural Networks (better if having experience with Convolutional Neural Networks).
- Elementary Linear Algebra, Calculus, and Statistics.

The Important Part Starts…

A BRIEF INTRODUCTION TO ADVERSARIAL TRAINING

Story of a Counterfeiter

GeoPython 2019 Muttenz Basel Switzerland June 24-26

Mary

First Sketch

Mona Lisa

John

Story Continued…

GeoPython 2019 Muttenz Basel Switzerland June 24-26

Mary

Tenth Sketch

Mona Lisa

John

UNDERSTANDING AND IMPLEMENTING GENERATIVE ADVERSARIAL NETWORKS (GANS) Tuesday, June 25, 2019 7

7

Mary Gaining Perfection…

GeoPython 2019 Muttenz Basel Switzerland June 24-26

Mary

Fiftieth Sketch

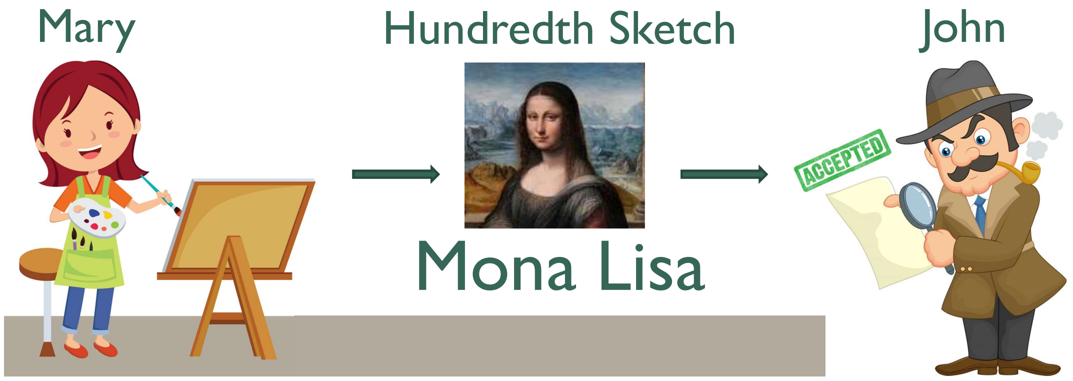
Mona Lisa

John

UNDERSTANDING AND IMPLEMENTING GENERATIVE ADVERSARIAL NETWORKS (GANS) Tuesday, June 25, 2019 8

8

Finally, Mary fooled John



Discriminative and Generative Models

- A ***Discriminative Model*** is a ***Supervised Learning*** based model which observes the data and learns to classify the data. These models don't stress much on the data distribution but on the quality of data.
- A ***Generative Model*** can be defined as a model (think of it to be like a mathematical representation of a real-world process) that is used to generate new data (which seems real) given the training data.
- Generative Models are able to learn mostly any kind of data distribution using unsupervised learning.
- A large amount of data in some domain (like images, audio, sequences) is used for training the generative model and the model learns the underlying data distribution. After learning the data distribution, the model is able to generate the data which looks real.

Continued…

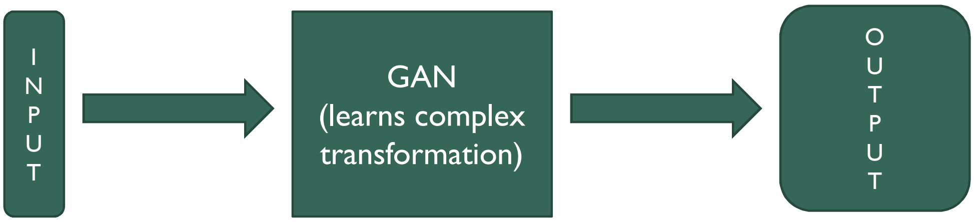
- In nutshell, *a Discriminative Model models the decision boundary between the classes*. On the other hand, a Generative Model models the actual data distribution of each class.
- A Discriminative Model is good at classifying/categorizing the data, whereas, a Generative Model is good at producing very similar data distribution.

Generative Adversarial Networks

- ***Generative Adversarial Networks*** are a type of ***Unsupervised Learning*** model.
- Generative Adversarial Networks are formed by combining a ***Discriminator*** (network using a Discriminative Model) and a ***Generator*** (network using a Generative Model) in form of a network.
- The goal of a GAN is to sample from a simple distribution (say a Normal Distribution) and then learn a complex transformation from simple distribution to the training distribution (training distribution is provided in form of bulk training data). Estimating the underlying distribution of data is termed as ***Density Estimation***.
- GANs use ***Implicit Density Estimation*** (there is no need to explicitly calculate probability distributions and then based on that generate samples; the goal is to generate real-looking samples given the training data).

GANs Continued…

GeoPython 2019 Muttenz Basel Switzerland June 24-26



Sample Distribution [m dimensional]
Z ~ N(0, I): Normal Distribution (say)

Generated Sample [n dimensional]

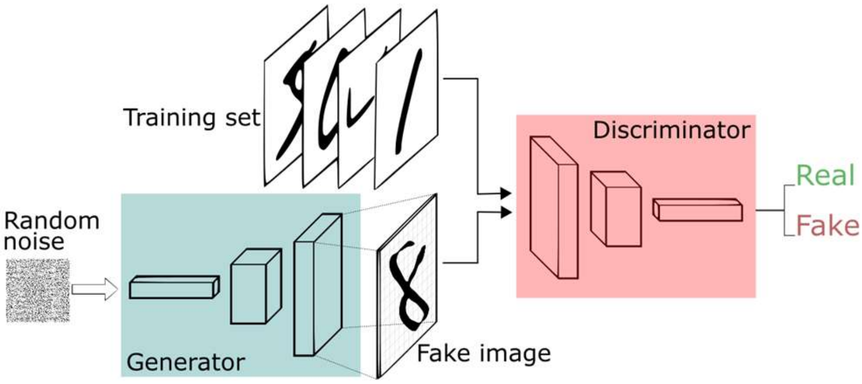
Training a GAN

GeoPython 2019 Muttenz Basel Switzerland June 24-26

- We use a two players game in order to train a GAN (a ***game-theoretic approach***).
- Generator is the first player. The job of the generator is to take a sample distribution (latent space) and through a complex transformation (using a neural network), learn the training distribution i.e. generate a real looking sample.
- Discriminator is the second player. The job of the discriminator is to get better and better at distinguishing between the ***"REAL"*** sample and the ***"FAKE"*** sample.
- The generator tries to produce samples that look so natural that the discriminator thinks the samples came from the real data distribution.
- Try to correlate this scenario with the story of "Mary the Counterfeiter" and "John the Investigator". Mary can be considered as the Generator and John can be considered as the Discriminator.
- GAN training becomes successful when the Discriminator is not able to distinguish between the real and the generated samples. This happens because the Generator learns the training data's distribution very well.

The Intuition and Setup

GeoPython 2019 Muttenz Basel Switzerland June 24-26

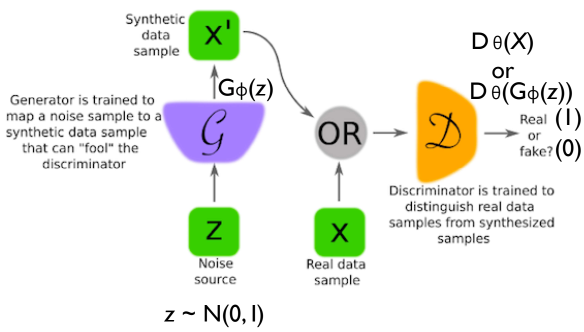


Source: https://skymind.ai/wiki/generative-adversarial-network-gan

Diving into Mathematics of GANs

GeoPython 2019 Muttenz Basel Switzerland June 24-26

- Let $G\phi$ denote the Generator and $D\theta$ denote the Discriminator. Here, ϕ and θ are the parameters of G and D, respectively.
- The Discriminator assigns its input a score between 0 and 1 (means the output is going to be sigmoid) => this tells the probability of the sample being real or fake.
- The Generator always wants the score given by the Discriminator to be highest i.e. equal to 1.



Mathematics of GANs

- Objective/Cost/Loss function for the Generator is the log likelihood:

$$\text{maximize} \log D_\theta(G_\phi(z))$$

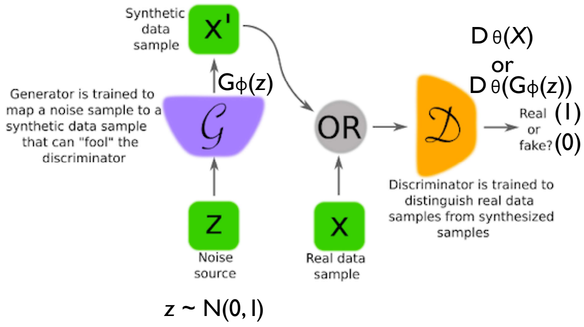
or

$$\text{minimize} \log (1 - D_\theta(G_\phi(z)))$$

Note:

[1] This is the objective function for just one sample z.

[2] Generator is a deterministic function i.e. for a sample z, it always gives the same result.

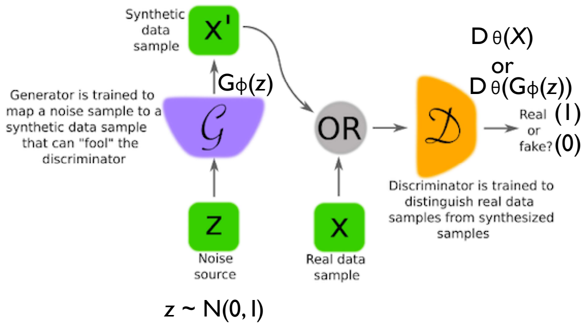


17

Mathematics of GANs

- Since z is continuous and normal, the objective function for all the samples taken together would be:

$$\underset{\phi}{\text{minimize}} \int p(z)\ (1 - \log D_\theta(G_\phi(z)))$$



18

Mathematics of GANs

- Discriminator's loss function has two components since Discriminator's task is two-fold (it should assign highest score to the real samples and the lowest score to the fake samples).

$$\underset{\theta}{\text{maximize}} \int p(x) \log D_\theta(x) + \int p(z) \left[\log\left(1 - D_\theta(G_\phi(z))\right)\right]$$

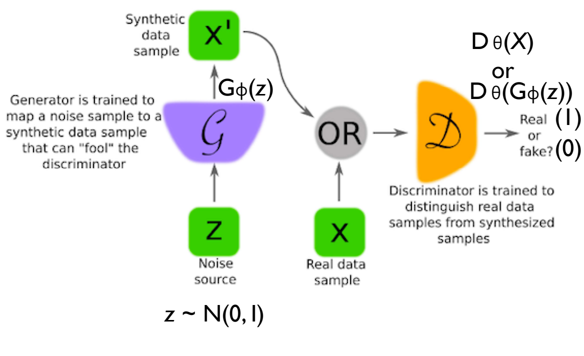
- The overall objective function (also called the ***minimax objective function***) comes out to be:

$$\underset{\phi}{\text{minimize}}\ \underset{\theta}{\text{maximize}} \left[\int p(x) \log D_\theta(x) + \int p(z) \left[\log\left(1 - D_\theta(G_\phi(z))\right)\right]\right]$$

Note: The Discriminator wants to maximize the second term and the Generator wants to minimize the second term.

Main Steps Involved in Training GANs

- The training happens in two steps:
  - **Step 1:** Perform Gradient Ascent (maximize the likelihood function) on the Discriminator.
  - **Step 2:** Perform Gradient Descent (minimize the likelihood function) the on Generator.



Algorithm for Training GANs

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

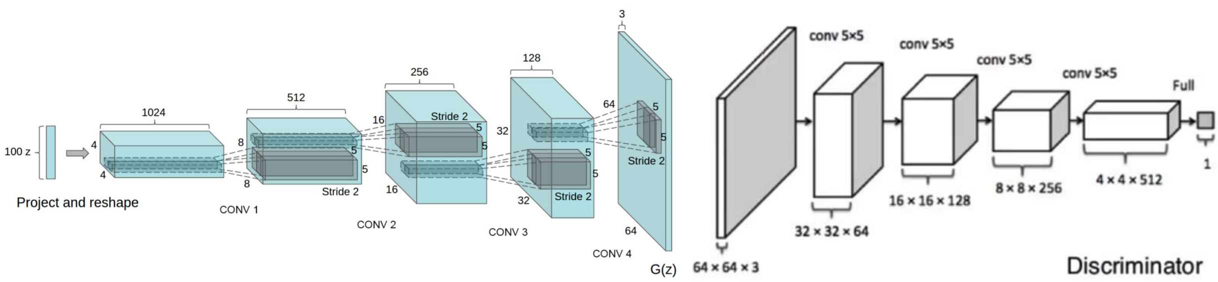
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Source: https://papers.nips.cc/paper/5423-generative-adversarial-nets

Architecture of DCGAN



Source: https://arxiv.org/pdf/1511.06434.pdf

Time for a Quick Hands-on

GeoPython 2019 Muttenz Basel Switzerland June 24-26

- Let's build our DCGAN (Deep Convolutional Generative Adversarial Network).
- This GAN will try to generate digits like the ones found in MNIST dataset.
- Code available at: https://github.com/greatdevaks/PyCon_Thailand_2019_GANs_Anmol_Krishan_Sachdeva

Best Practices for DCGAN

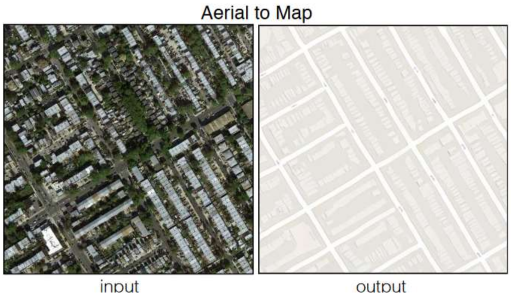
GeoPython 2019 Muttenz Basel Switzerland June 24-26

- Replace the pooling layers with strided convolutions in the Discriminator and fractional-strided/transpose convolutions in the Generator.
- Use Batch Normalization in both the Discriminator and the Generator.
- Remove Fully Connected hidden layers from deeper architectures.
- Use ReLU activation in the Generator for all the layers except for the output, which uses tanh activation.
- Use Leaky ReLU activation in the Discriminator for all the layers.

Applications of GANs

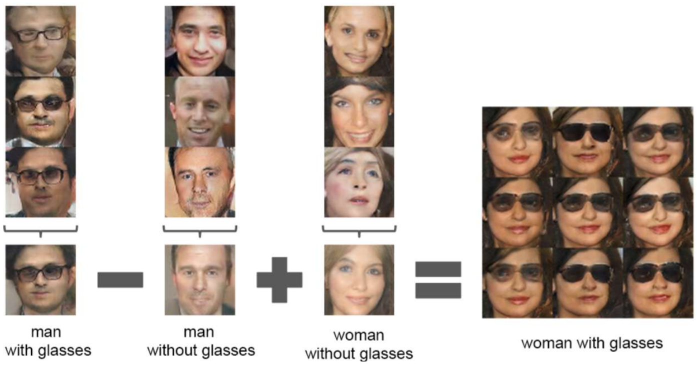
~ Satellite View ⇔ Map View:

- GeoGAN [https://arxiv.org/abs/1902.05611]
- Pix2Pix [https://arxiv.org/pdf/1611.07004.pdf]



Some more Applications of GANs

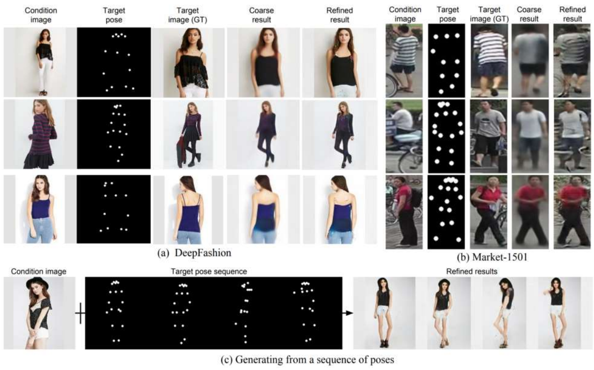
~ Vector Operations on Visual Objects:



Continued…

GeoPython 2019 Muttenz Basel Switzerland June 24-26

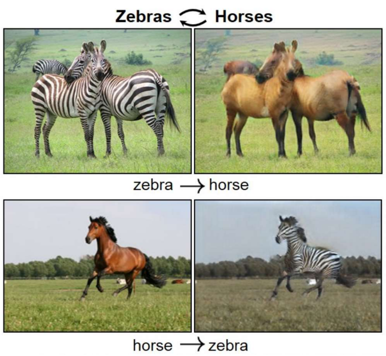
~ Pose Guided Image Generation [LINK]:

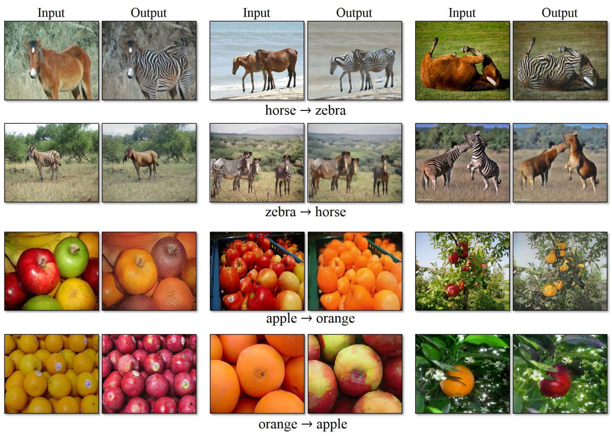


Continued…

GeoPython 2019 Muttenz Basel Switzerland June 24-26

~ CycleGAN [Link]:





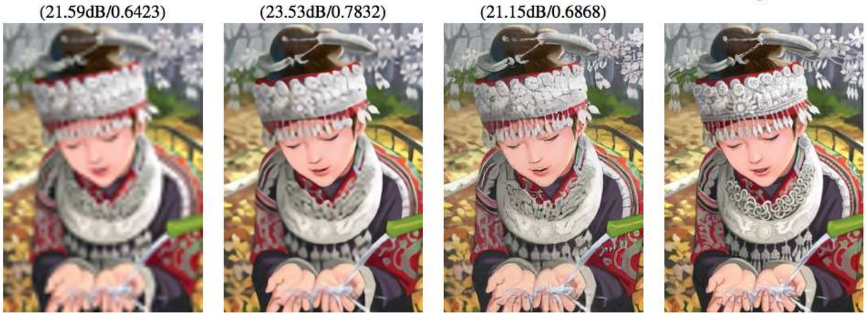
Continued…

~ PixelDTGAN - Suggesting Merchandise based on Real Photos [Link]:

Continued…

~ SRGAN [Link to Paper]:

bicubic (21.59dB/0.6423) | SRResNet (23.53dB/0.7832) | SRGAN (21.15dB/0.6868) | original



Continued…

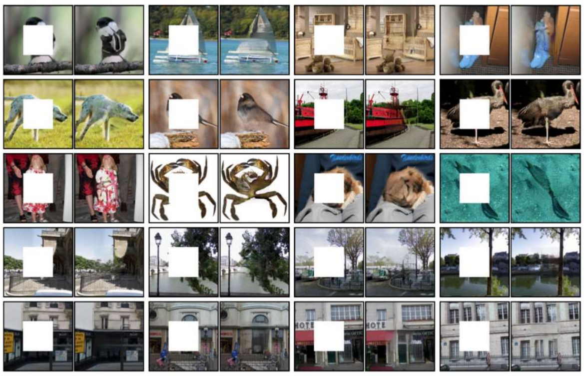
~ Progressive GAN - First Commercial-like Image Quality [Link to Paper]:

Continued…

~ Test-to-Image using Stack GAN [Link]:

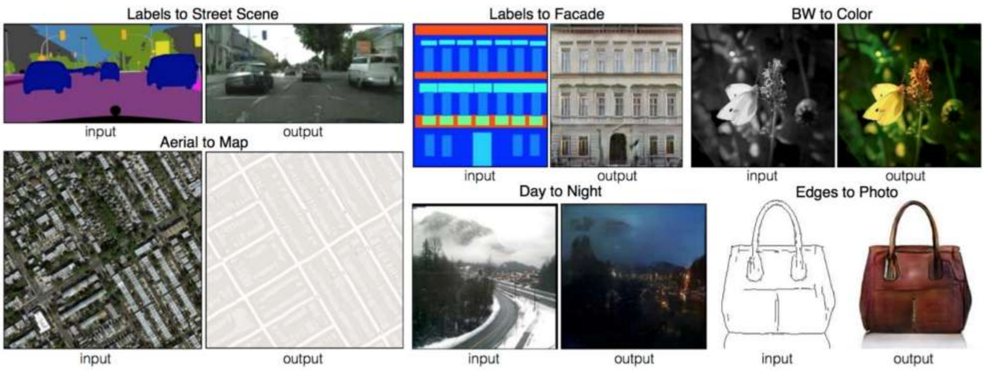
Continued…

~ Image Inpainting using Context Encoder [Link]:



Continued…

~ Image-to-Image Translation using Pix2Pix [Link]:



Acknowledgements

GeoPython 2019 Muttenz Basel Switzerland June 24-26

- I thank GeoPython's organizers and all its team members for organizing such a nice conference.
- I would like to thank Dr. Ian Goodfellow and all the researchers who have contributed towards the concept of GANs.
- My sincere thanks to Prof. Cian O'Donnell, Prof. Tilo Burghardt, and Prof. Carl Henrik Ek [all from University of Bristol, UK] for imparting invaluable knowledge to me in this domain.
- I would like to thank Bigbasket, India and all my team members for supporting and motivating me.

University of BRISTOL

big basket India's largest online supermarket

Catch Me Next At…

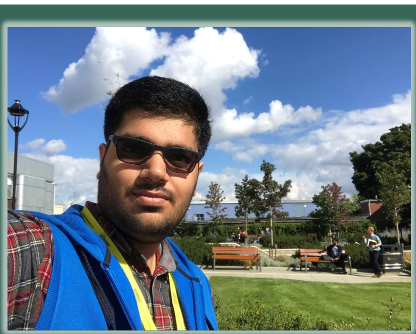
GeoPython 2019 Muttenz Basel Switzerland June 24-26

- Speaking at EuroPython, Switzerland [July 8 – 14, 2019].

Thanks a lot to everyone sitting in the hall for being such a nice audience

I WILL BE HAPPY TO ANSWER YOUR QUESTIONS…

#GeoPythonConf
@greatdevaks



Anmol Krishan Sachdeva
Platform Software Engineer - Bigbasket, India
MSc Advanced Computing, University of Bristol, UK
Email Id: greatdevaks@gmail.com
LinkedIn: https://www.linkedin.com/in/greatdevaks
Twitter: https://www.twitter.com/greatdevaks
GitHub: https://www.github.com/greatdevaks

Time For Feedback

- It will take only 2 minutes. I will be highly obliged if you fill the feedback form.
- Link: https://tinyurl.com/geo-gan-19
- Note: No personal details asked.