CS167: Machine Learning

Generative Modeling:
Generative Adversarial Network (GAN)

November, 6th, 2024

Can you recognize any famous person here?

A style-based generator architecture for generative adversarial networks - Tero Karras, Samuli Laine, Timo Aila - arXiv18

Today’s Topics

- **Generative Modeling**

- Introduction to generative adversarial network (GAN)

- GAN variations

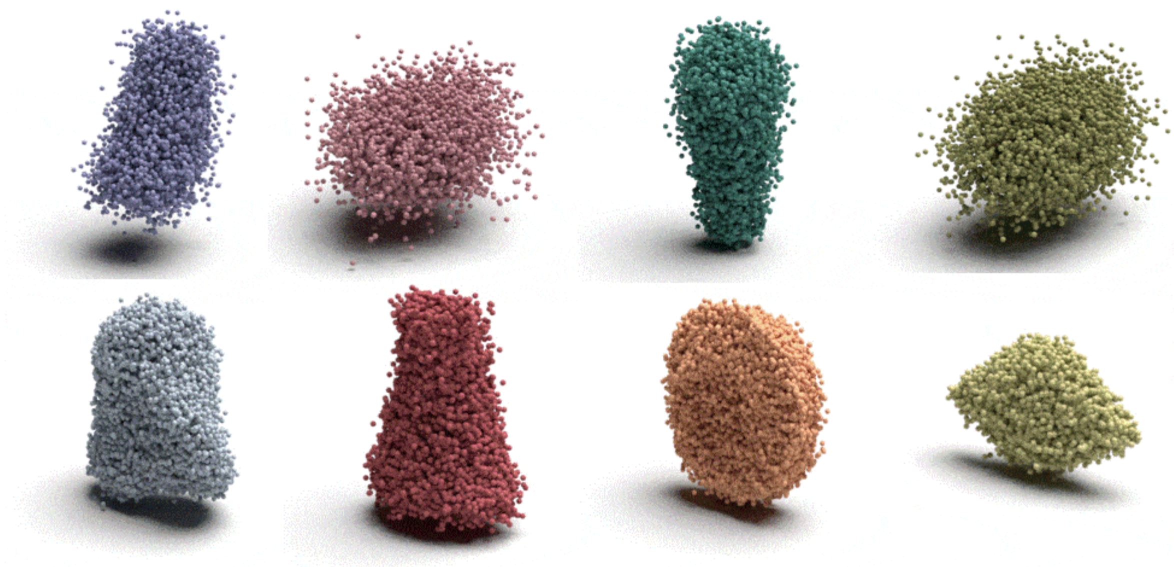
- Computer vision applications with GAN

- GAN model in my research

Generative Model

- Generative modeling have achieved state-of-the-art performance on many downstream tasks and applications

- <u>Shape generation:</u> generate shapes from 3D point-cloud



[Learning Gradient Fields for Shape Generation - ECCV'20]

Generative Model

- Generative modeling can be used on many downstream tasks and applications

- Audio synthesis: generate audio that sounds realistic

[DiffWave: A Versatile Diffusion Model for Audio Synthesis - ICML'21]

- Music generation: generate new music

[Symbolic Music Generation with Diffusion Models]

Types of Generative Models

- Generative modeling is based on representation of probability distribution
- Existing generative modeling techniques can be classified into three broad categories
  - Implicit generative models
  - Likelihood-based models
  - Score-based models (diffusion process)

Generative Model: Implicit Model

- Generative modeling is based on representation of probability distribution

- Implicit models:
  - probability distribution is represented by a model of its sampling process

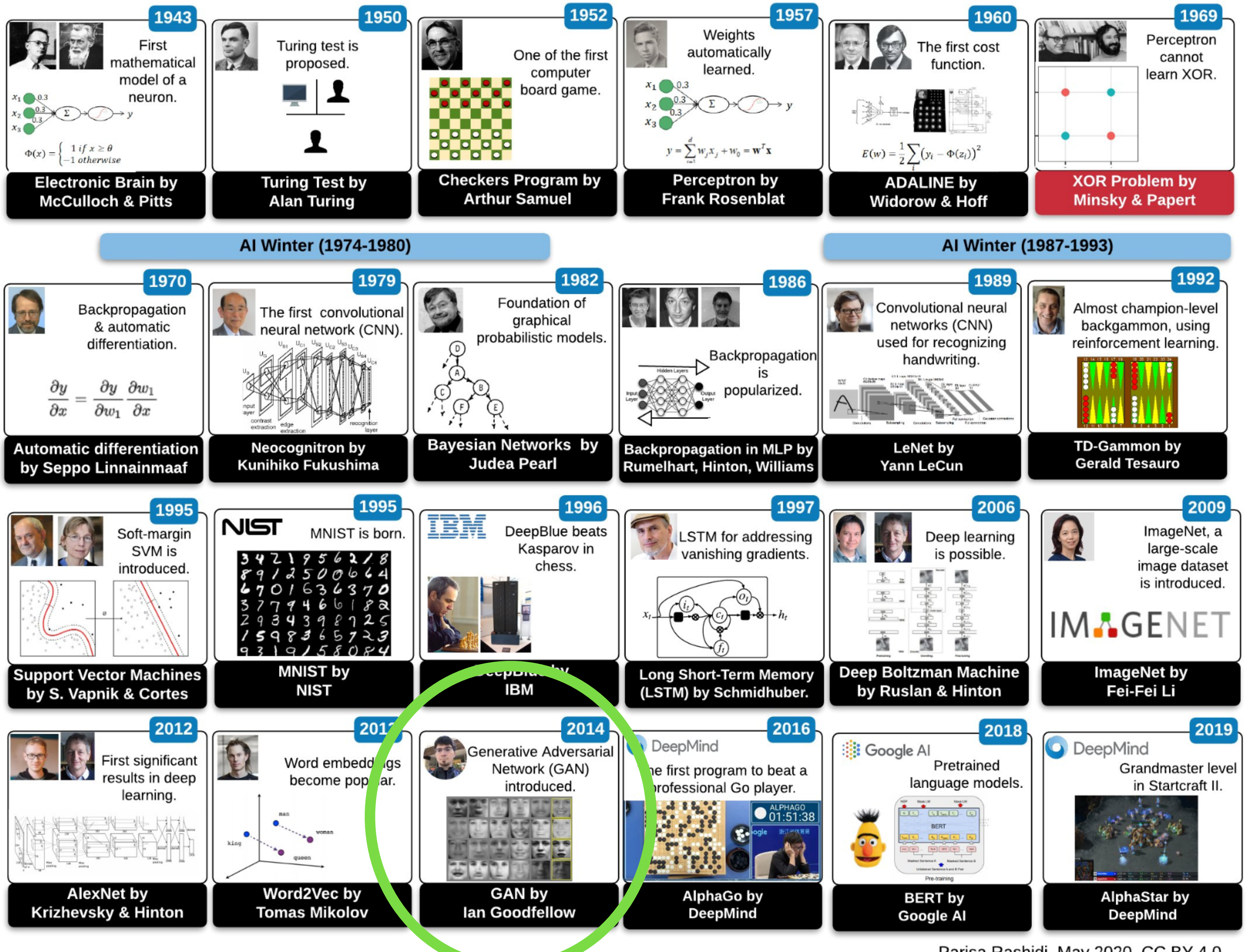
  - **Generative Adversarial Network (GAN)** is an example of implicit generative model. It implicitly represents a distribution over all objects that can be produced by the generator network

  - Limitations
    - requires adversarial training which is very unstable (eg, due to mode collapse)

Today’s Topics

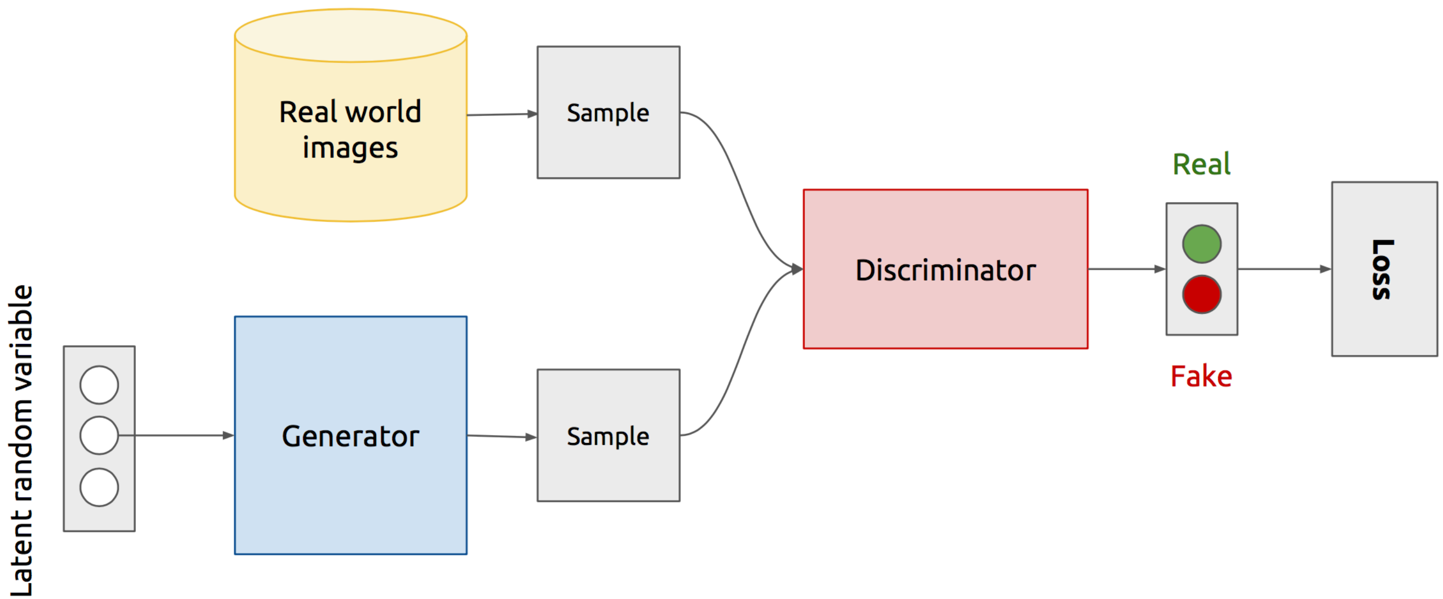
- **Generative Modeling**
- **Introduction to generative adversarial network (GAN)**
- GAN variations
- Computer vision applications with GAN
- GAN model in my research

Where is GAN in AI's historical milestones?



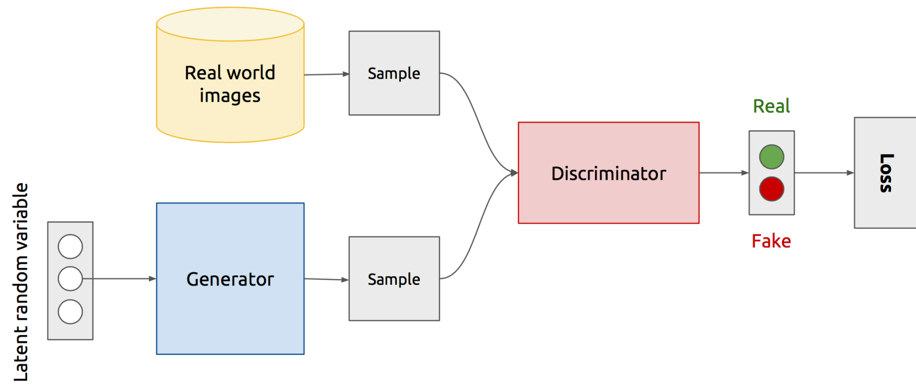
What is GAN — intuitive introduction?

- Adversarial play between a two entities: a generator and a discriminator
  - Generator
  - Discriminator

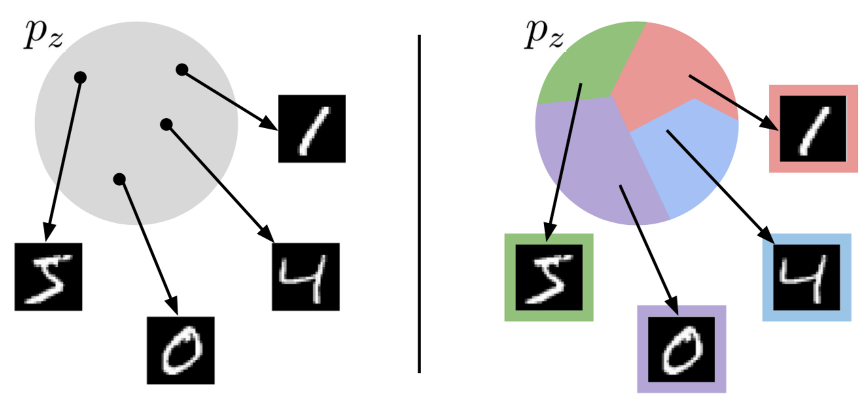


Random variable as input to Generator

- Random variable: something that's easy to sample from, like a <u>uniform distribution</u>
- Generator then transforms this noise into a meaningful output
- GAN tries produce a wide variety of data, sampling from different places in the target distribution

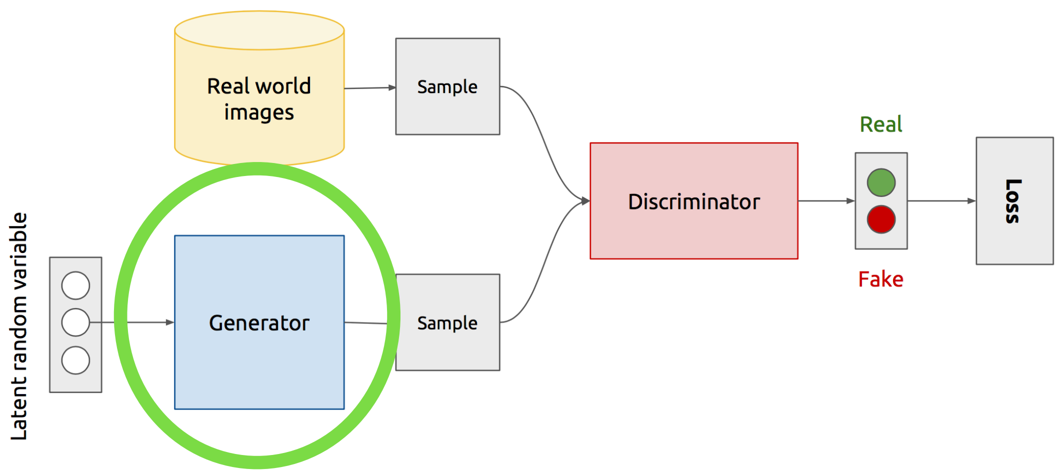


$$q_\theta(x) = G(z), z \sim p_z$$



Generator training

- Sample random noise
- Produce generator output from sampled random noise
- Let discriminator determine "Real" (label=1) or "Fake" (label=0) classification for generator output
- Calculate loss from discriminator classification
- Backpropagate through both the discriminator and generator to obtain gradients
- Use gradients to change only the generator weights

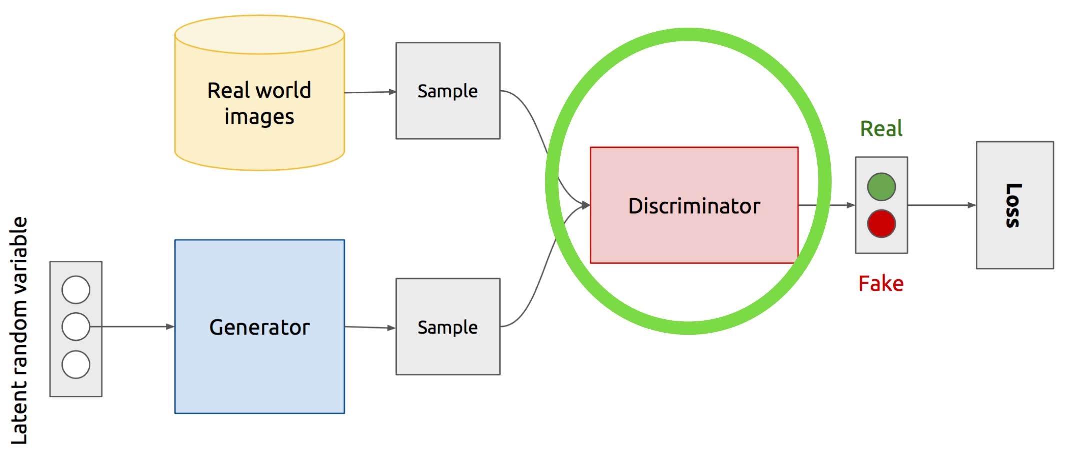


Discriminator training

- The discriminator classifies both "Real" (label=1) and "Fake" (label=0) from the generator
  - Two sets of inputs: ”Real" (label=1) and "Fake" (label=0)

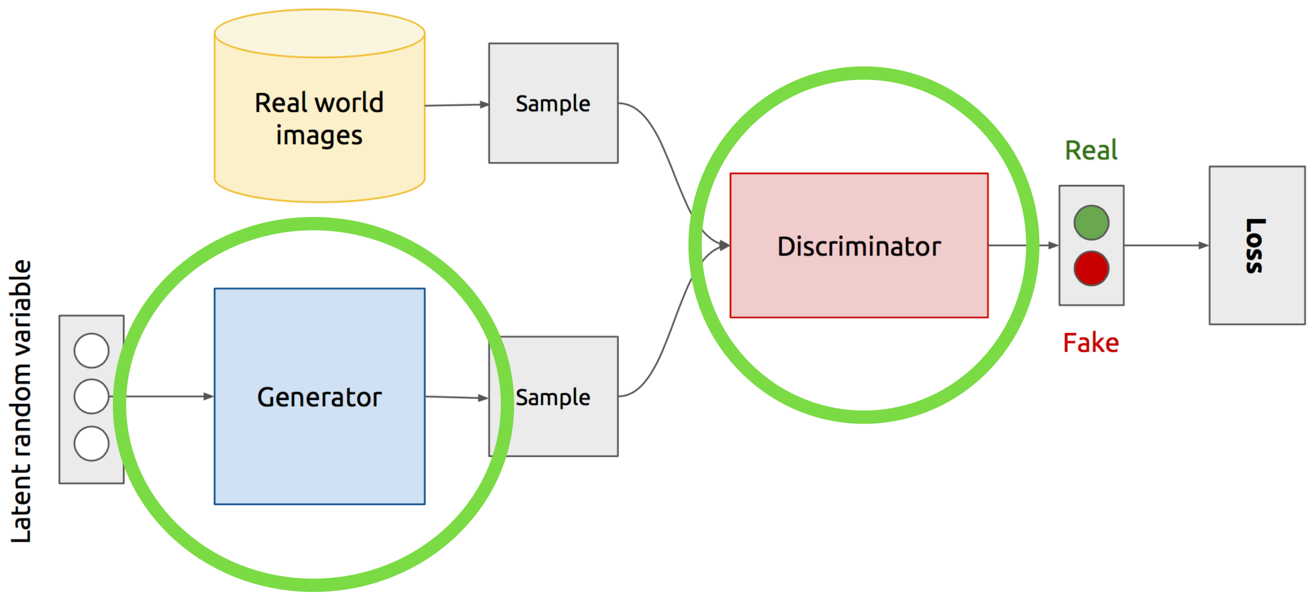
- The discriminator loss penalizes the discriminator for misclassifying a real instance as fake or a fake instance as real

- The discriminator updates its weights through backpropatation from the discriminator loss



Overall GAN training

- GAN training proceeds in alternating periods (for several times until convergence):
  - The discriminator trains for one or more epochs
  - The generator trains for one or more epochs



How does GAN work — mathematical model?

- Generator tries to minimize the loss function while the discriminator tries to maximize

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

- Discriminator tries <u>gradient ascent</u> to <u>maximize</u> using the following cost

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

- Generator tries <u>gradient descent</u> to <u>minimize</u> using the following cost

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

Today’s Topics

- **Generative Modeling**
- Introduction to generative adversarial network (GAN)
- **GAN variations**
- Computer vision applications with GAN
- GAN model in my research

GAN variations

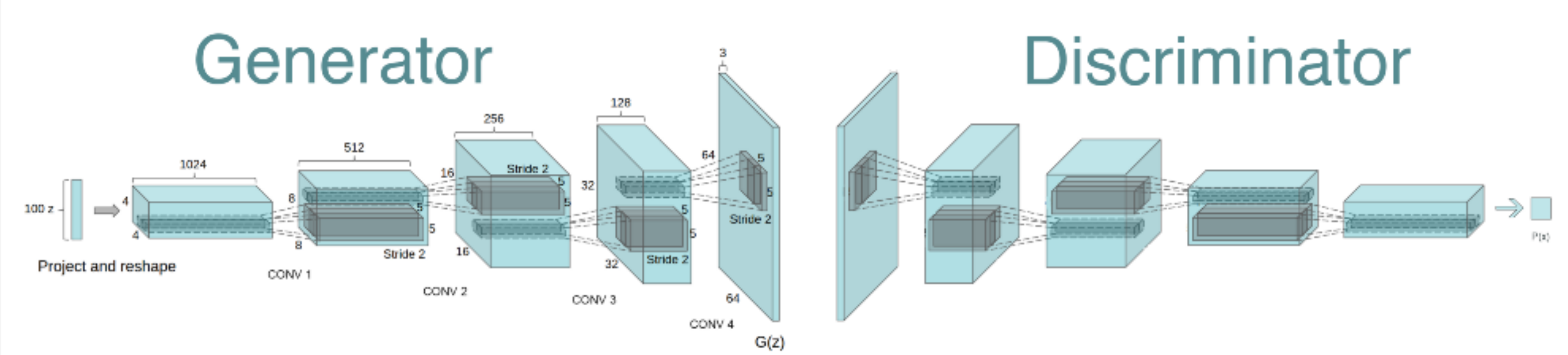
- Deep Convolutional GAN (DCGAN)

- Conditional GAN (cGAN)

- Wasserstein GAN (WGAN)

GAN Variations

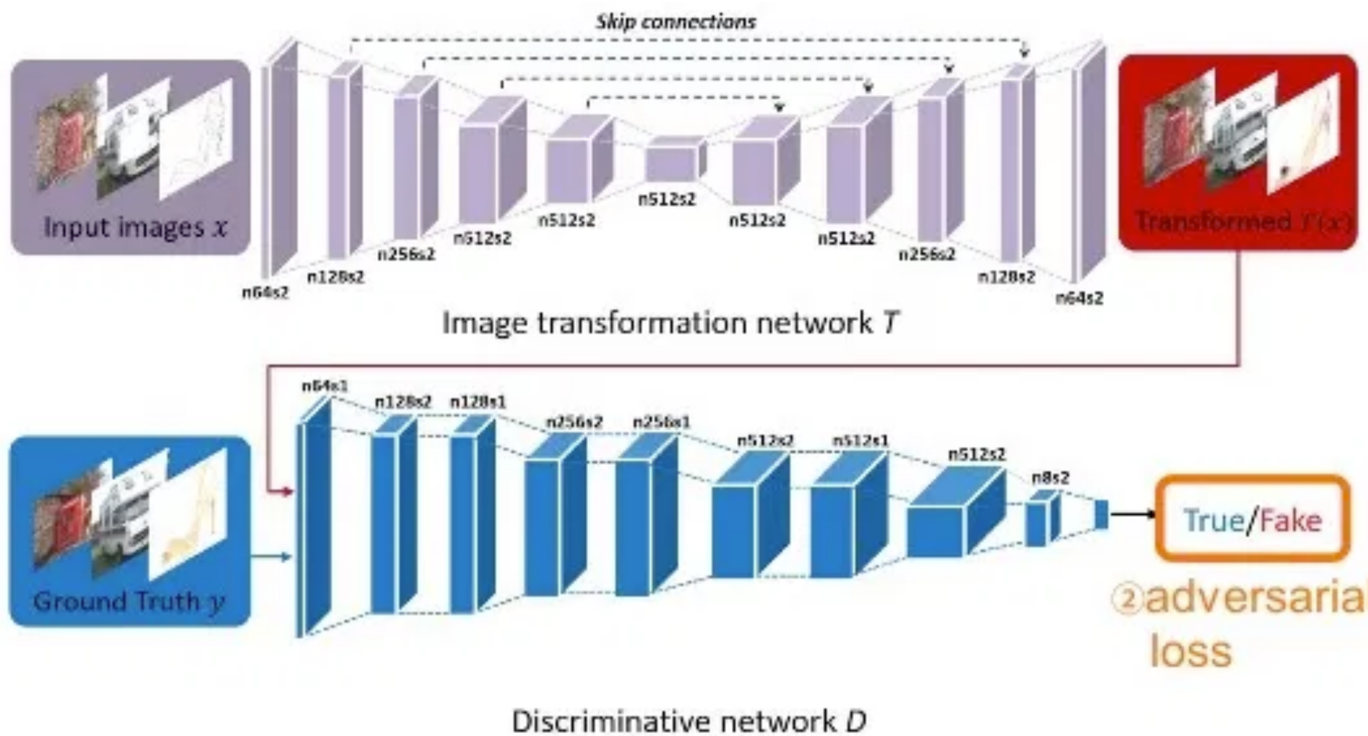
- Deep Convolutional GAN (DCGAN)
  - Convolutional layers to generate image
  - Can be used to generate photorealistic images



GAN Variations

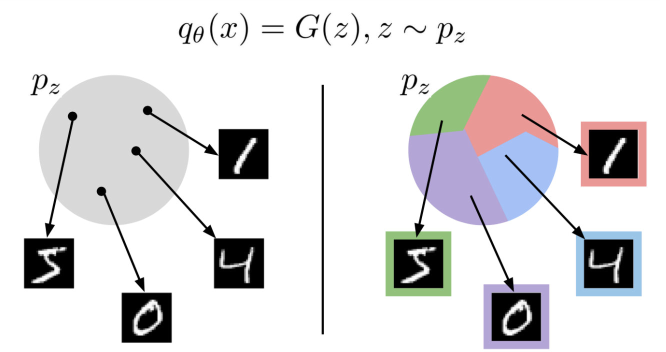
- Conditional GAN (cGAN)

  - Can be used to translate from one domain (source) into another (target)
  - In addition to the random noise, sample from source domain is fed into generator



GAN Variations

- Wasserstein GAN (WGAN)
  - Different loss function than minimax (the one I showed so far)
  - Help address the issue of mode collapse issue during GAN training
    - Generator may collapse to a setting where it always produces same outputs (gets stuck in a small space with low variety), this is a common failure case for GANs
    - fails to learn to represent the complex real-world data distribution



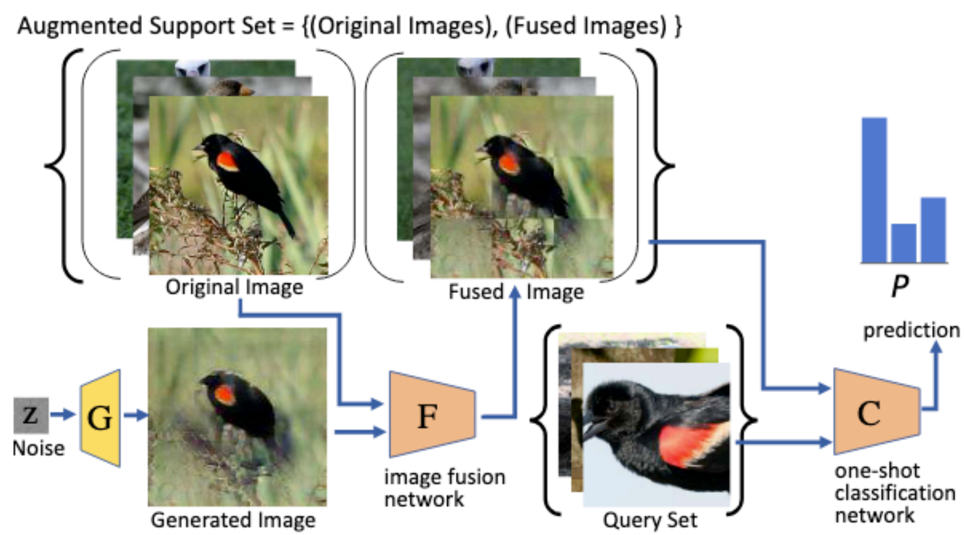
Today's Topics

- **Generative Modeling**
- Introduction to generative adversarial network (GAN)
- GAN variations
- **Computer vision applications with GAN**
- GAN model in my research

Why GAN is useful?

- Image generation eg, StyleGAN, change the content/style of the image

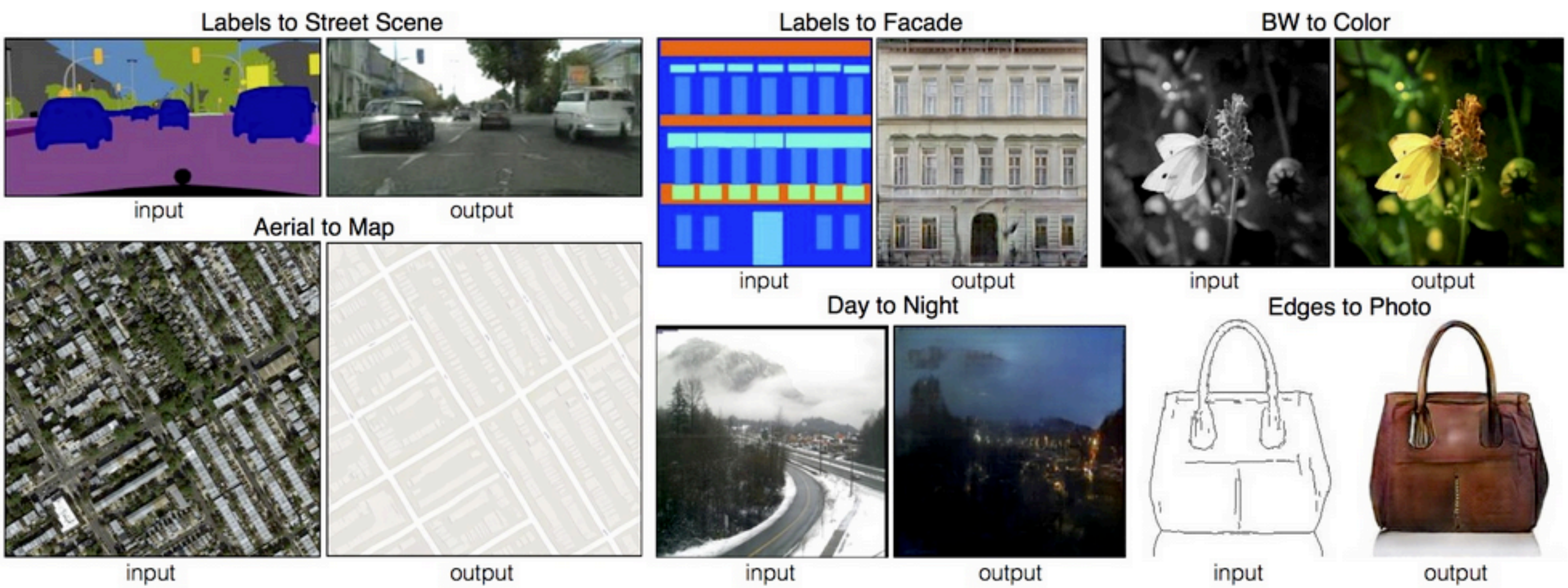
- Data augmentation
  - Generate synthetic data
  - improve your downstream models with GAN-generated data



Computer Vision Applications

- Paired Image to image translation (pix2pix)
  - Add extra loss function in addition to minimax
  - U-Net generator
  - PatchGAN discriminator

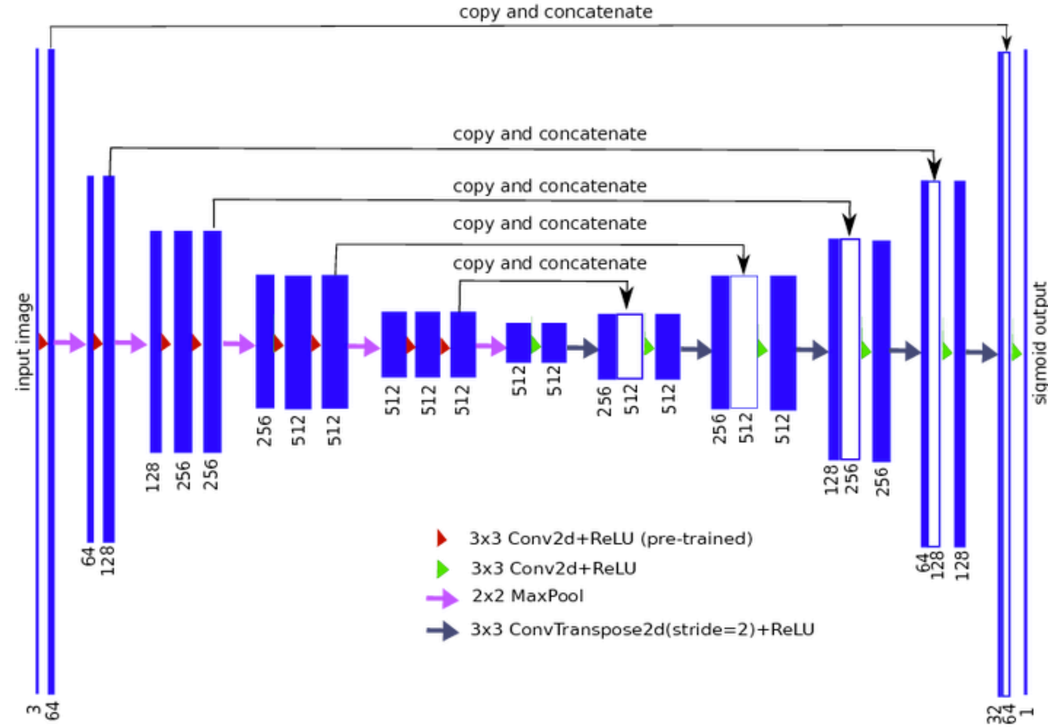
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$



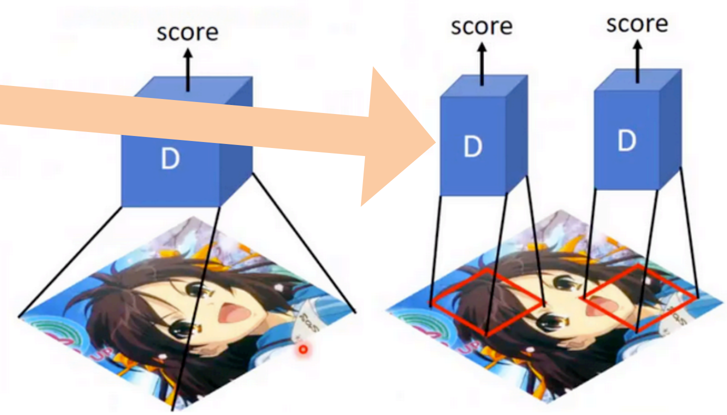
Example results on several image-to-image translation problems. In each case we use the same architecture and objective, simply training on different data.

Computer Vision Applications

- Pix2pix
  - U-Net **generator**

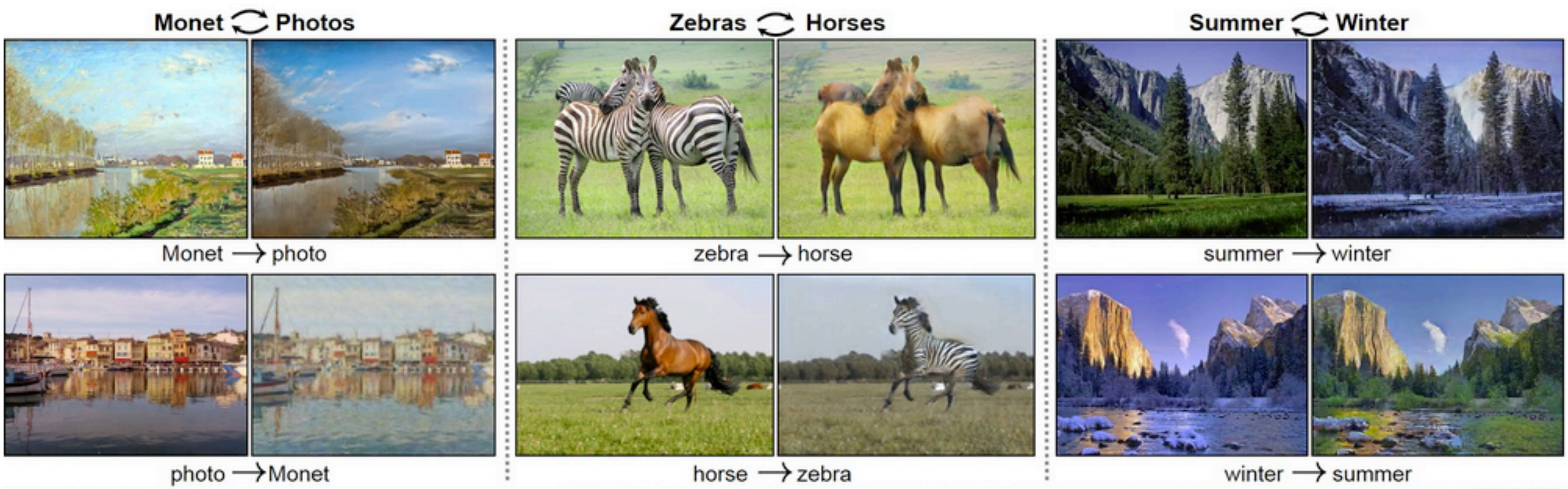


  - PatchGAN **discriminator**



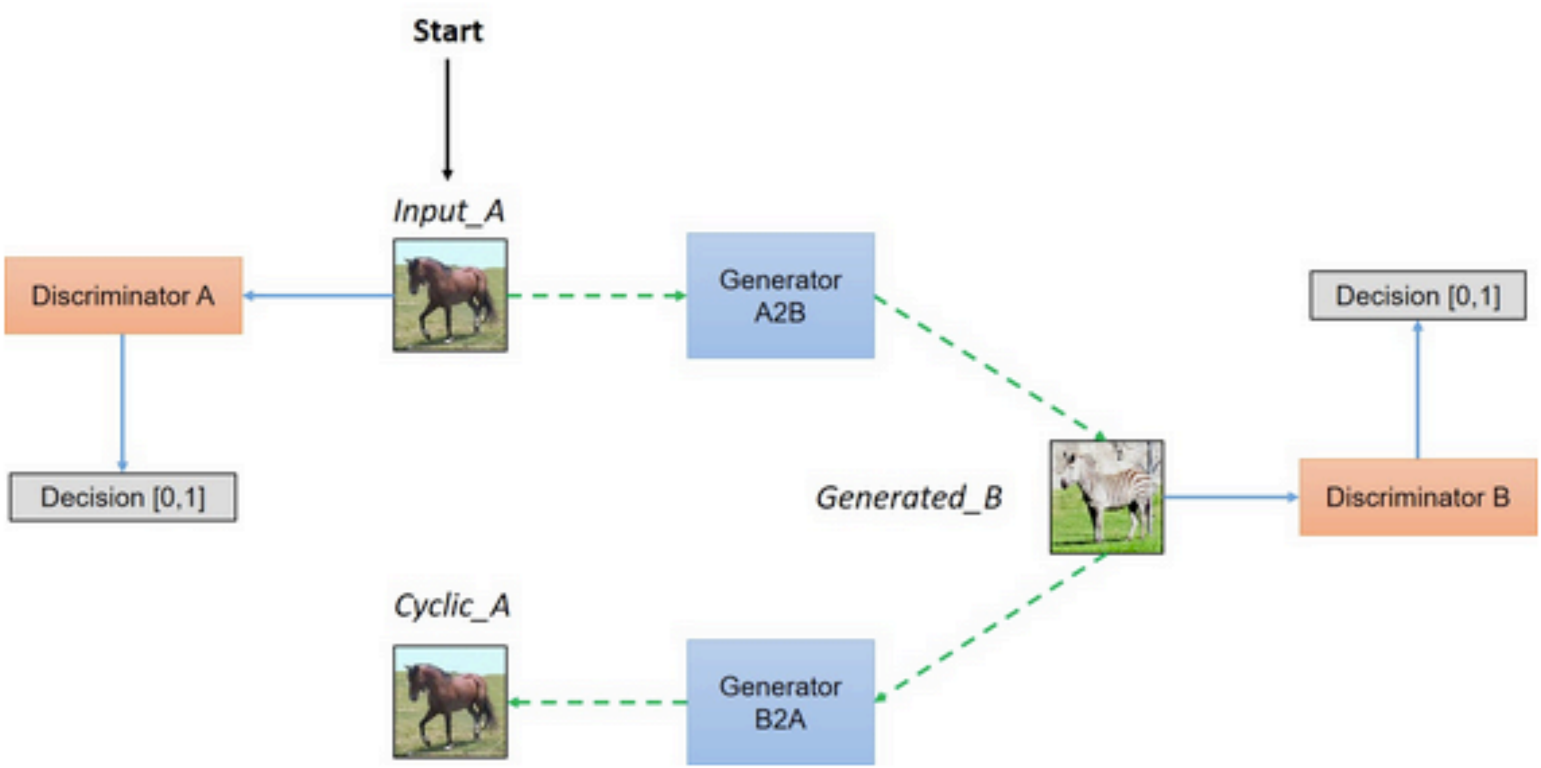
Computer Vision Applications

- <u>Unpaired image-to-image translation (CycleGAN)</u>
  - an unpaired image-to-image translation model, to adapt horses to zebras (and vice versa) with two GANs in one



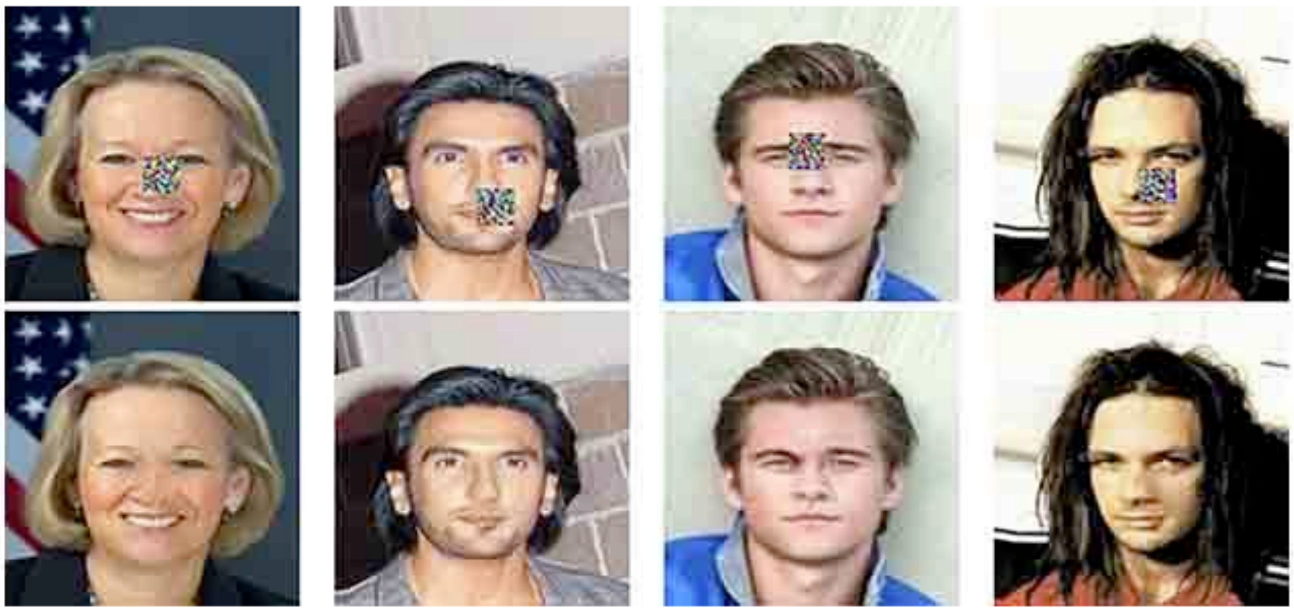
Computer Vision Applications

- <u>Unpaired image-to-image translation (CycleGAN)</u>
  - an unpaired image-to-image translation model, to adapt horses to zebras (and vice versa) with two GANs (A2B and B2A)



Computer Vision Applications

- <u>Face Inpainting</u>
  - Filling the masked region of a real image



Computer Vision Applications

- Text to Image (StackedGAN)
  - Generating an image from a textual description

Today's Topics

- **Generative Modeling**
- Introduction to generative adversarial network (GAN)
- GAN variations
- Computer vision applications with GAN
- **GAN model in my research**

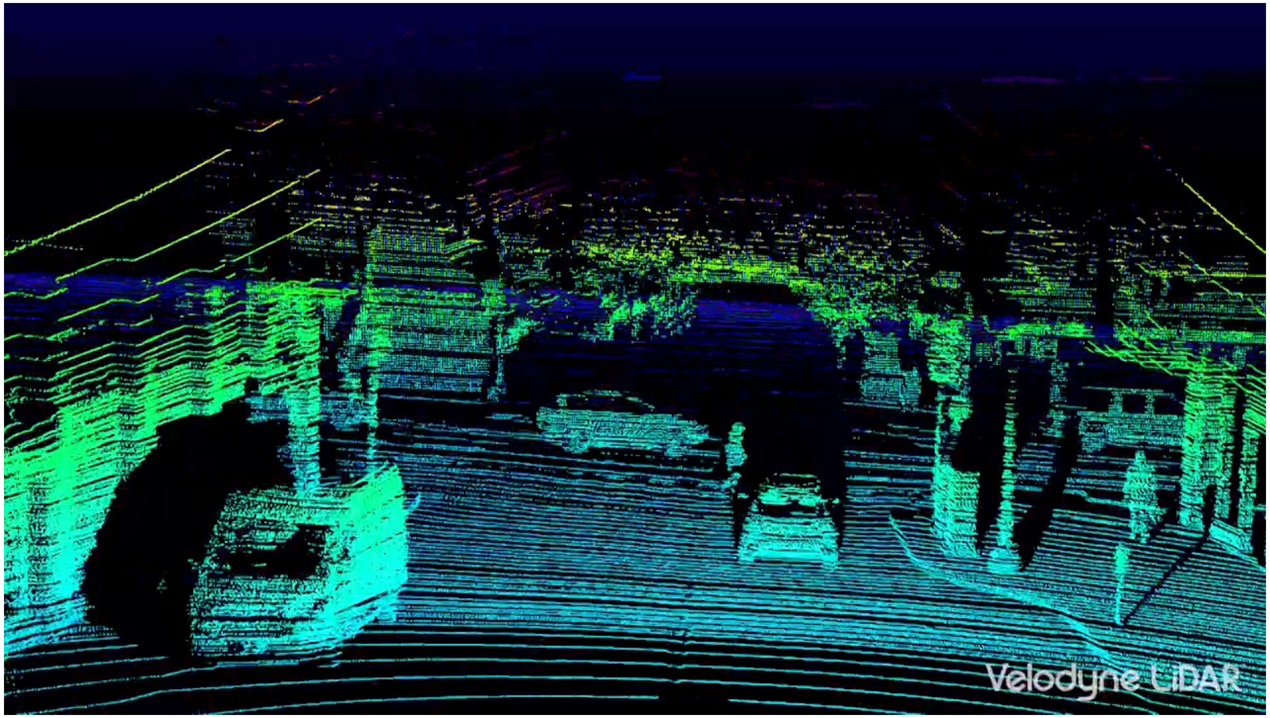
Depth Estimation from an Image

- **Motivation**
  - Outdoor image may contain object at a far-away distance
  - Useful for an autonomous robot to find depth of a distant object
  - object detection in military mission
  - surveillance
  - autonomous driving etc

Figure: Long-range surveillance

Depth Estimation from an Image

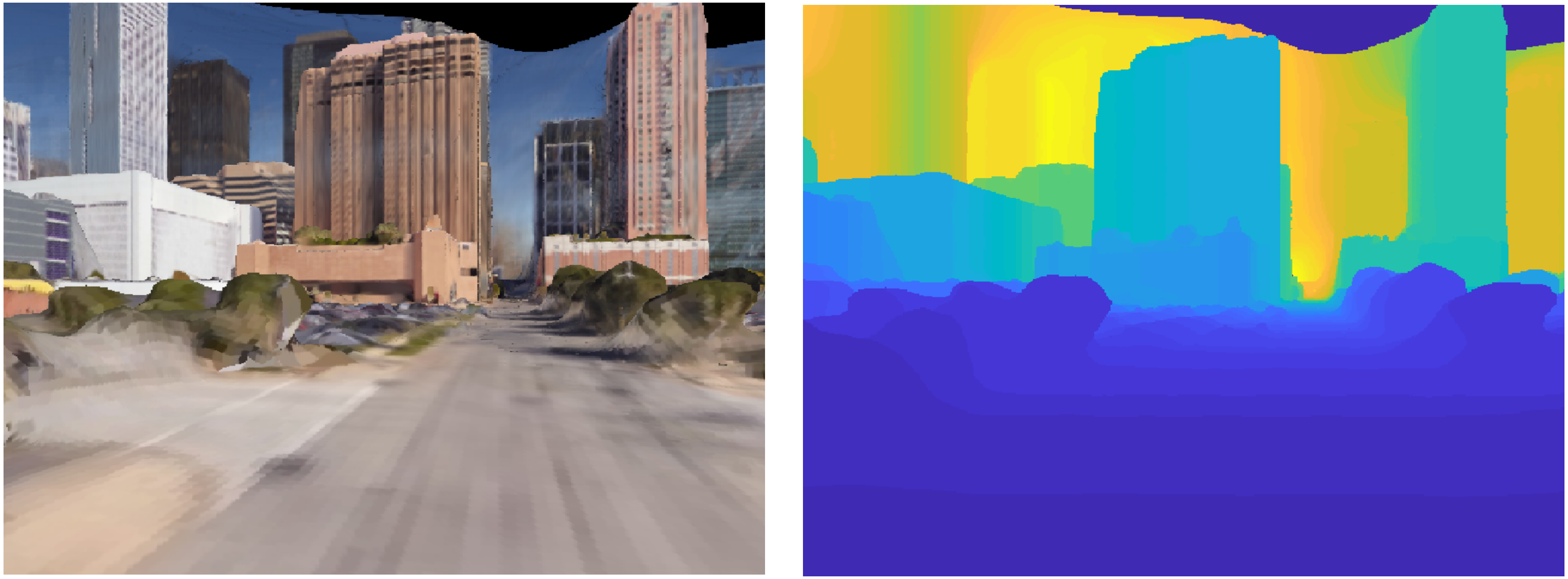
- **Difficulties with existing approaches**
  - Most of monocular depth estimation algorithms provide depth estimates less than 100 meters
  - Depth sensors (eg, Laser, LiDAR) have limited range:



Depth Estimation from an Image

- **My solution:** GAN based algorithms that can predict the depth from an image

Figure: RGB and generated Depth images in the city of Houston, Texas



- Developed an RGB and Depth image pair generation algorithm that goes up to 1000 meters
  - utilizing 3D reconstruction of the scene from multi-view images

Depth Estimation from an Image

- **Conditional Generative Adversarial Network (cGAN*):**
  - Depth prediction as an image-to-image translation problem using cGAN, where the goal is to translate an RGB image (source domain) to a depth image (target domain)

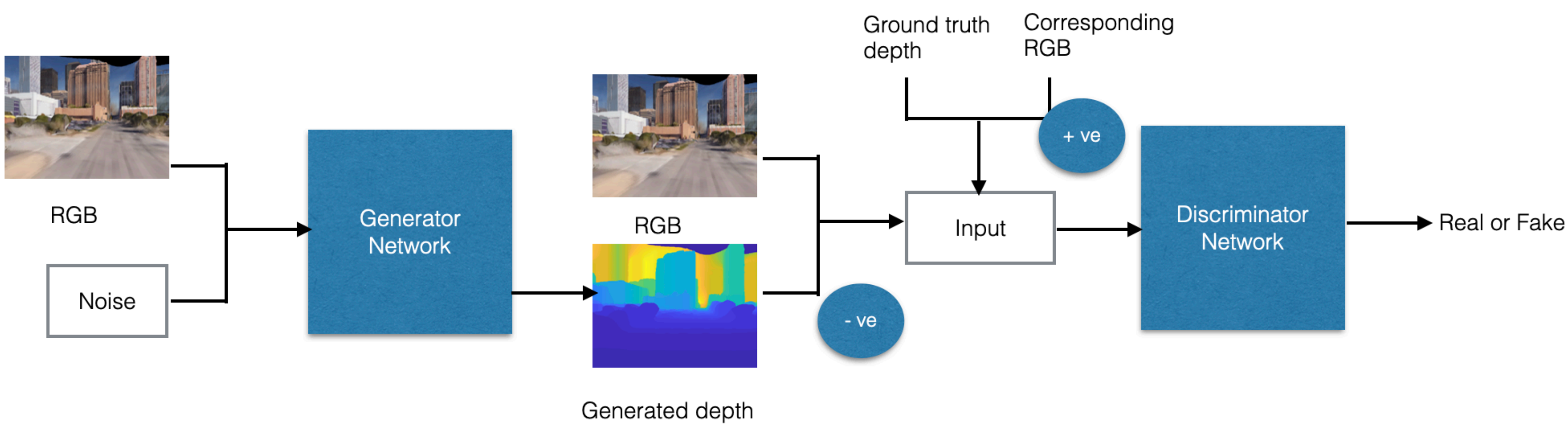


Figure: cGAN architecture for depth prediction

*P. Isola, J. Zhu, T. Zhou, A. Efros, Image-to-Image Translation with Conditional Adversarial Networks, (CVPR), 2017

Synthetic Data Generation

- Collect urban images of a geolocation (eg, New York City, Chicago, Houston) using Google Earth software
- Reconstruct the 3D model from the images
- Annotate key points for the trajectory along the model
- Render the RGB+ Depth image pairs along the trajectory

Image Collection for 3D Reconstruction

- Extract images (street view) of a city eg, NYC from multiple viewpoints
- Google Earth allows to record a video-clip of a city as user hover around from different viewpoints
- Extracted images are sampled for 3D reconstruction using *Adobe Remake*

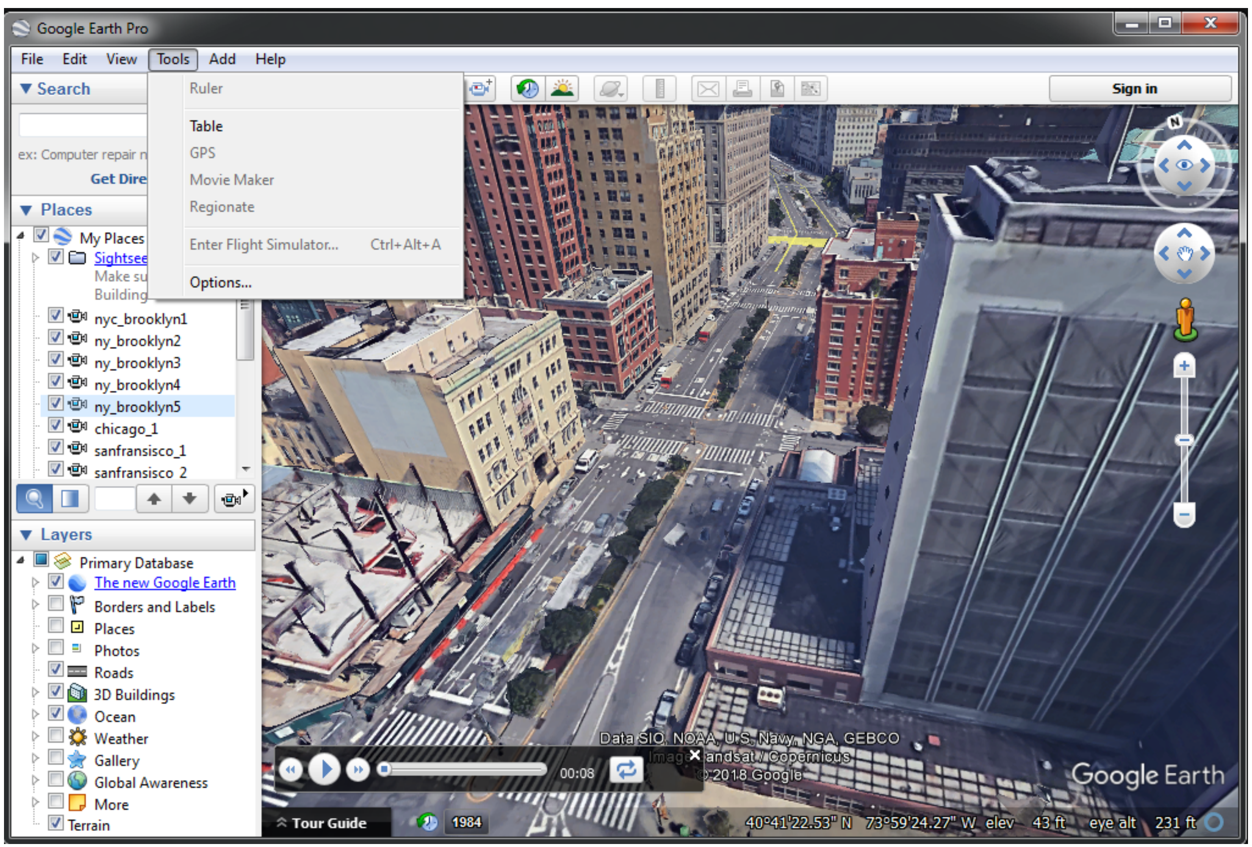


Figure: Collecting a video clip from Google Earth Pro in New York City. The *Movie Maker* feature enables this option. Images are exported from the recorded video clip at 30 fps.

3D Reconstruction using Adobe Remake

- Model of the city is reconstructed from a subset of manually selected images (120~250)

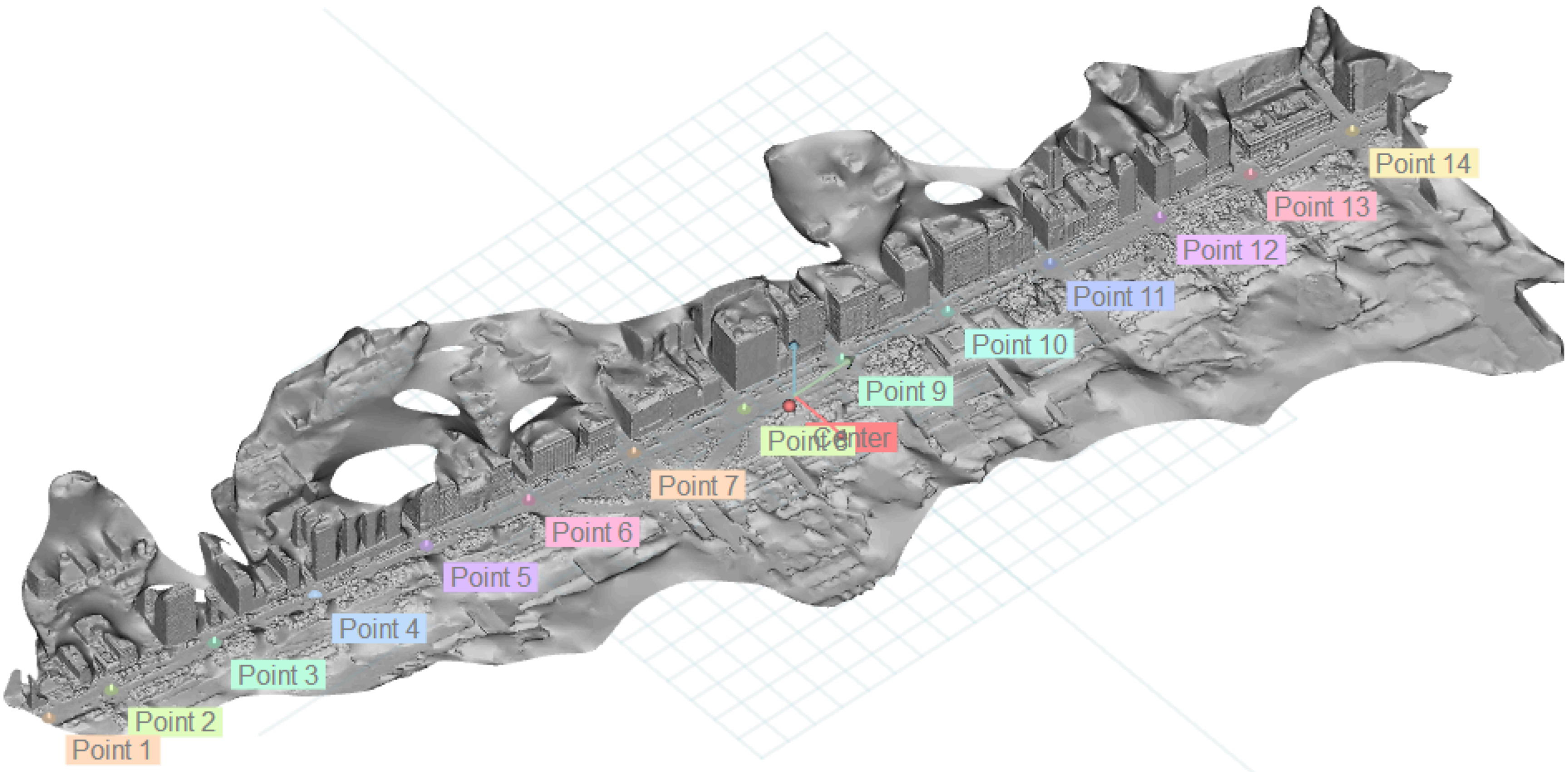


Figure: An example 3D reconstructed model from our dataset. The keypoints (*Point 1*, *Point 2*, *Point 3*, etc.) are manually selected in the 3D model.

Render an RGB and Depth pairs from 3D Model

- Generate a trajectory of a moving virtual camera between two keypoints *A* and *B* in the 3D model
- Render the scene from the viewpoint of the virtual camera using OpenGL from each point along the trajectory

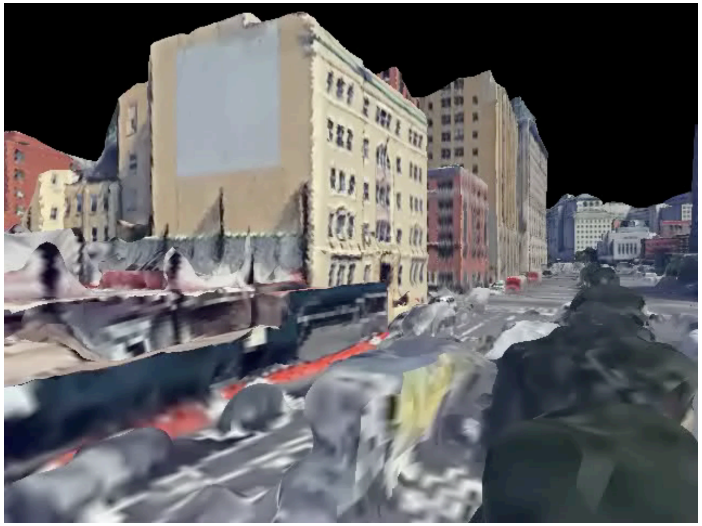
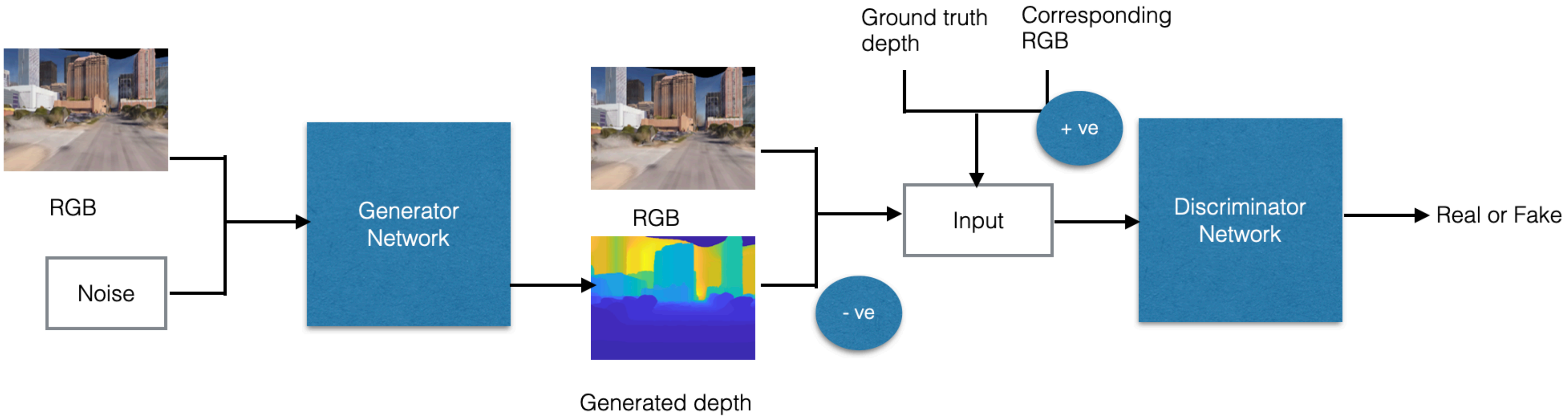


Figure: Rendered RGB and Depth images from different virtual camera trajectory locations inside the *New York City* reconstructed 3D model.

cGAN trained using rendered depth images

- Trained cGAN using the quantized depth values as ground truths



- Evaluated depth prediction on 3 metrics:
  - Absolute Relative Error
  - Linear RMSE
  - Scale Invariant RMSE

Scene	# Frames
New York City	300
Miami	250
Houston	250
Salt Lake City	100
Total	900

Visualization of the Depth Prediction

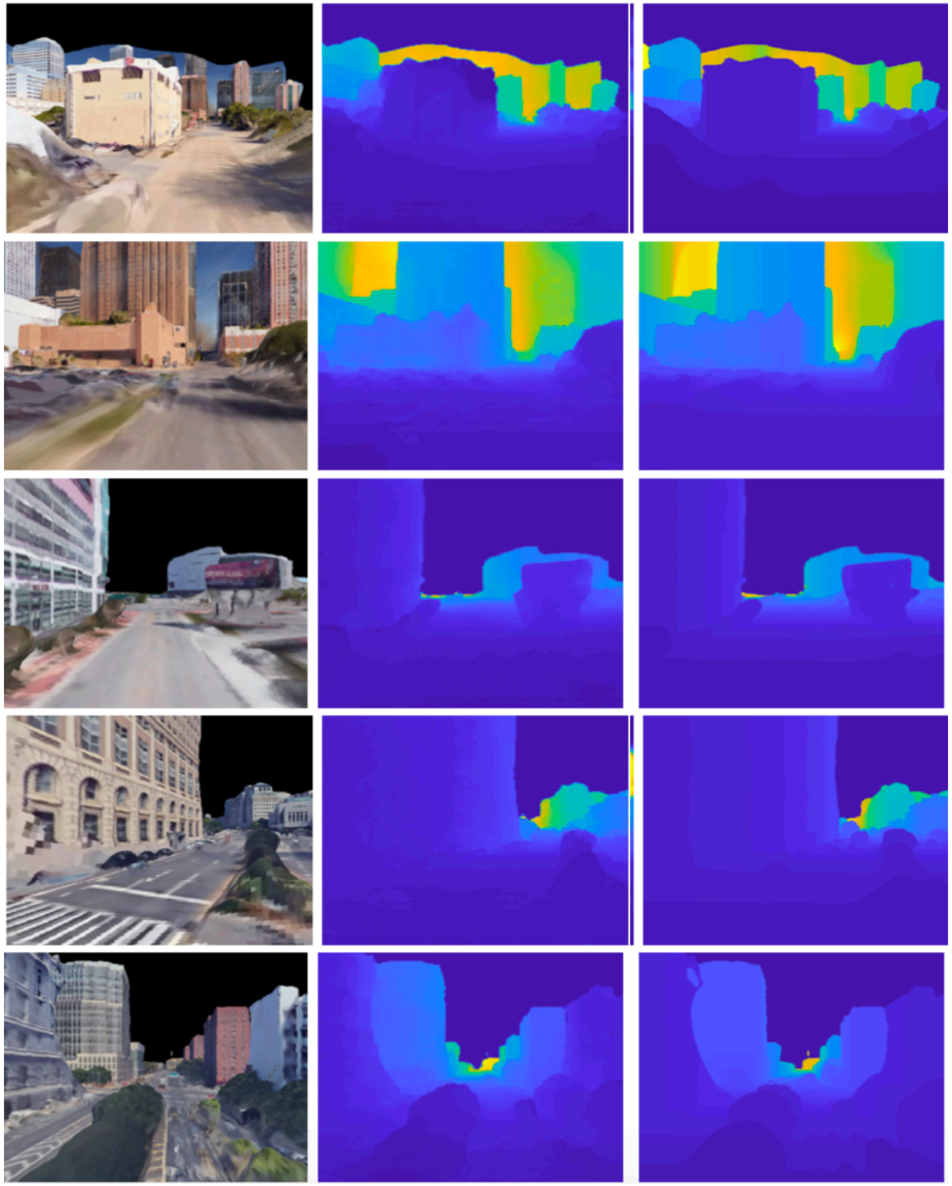


Figure: (From left to right) the input RGB image, the prediction from GAN model, and the ground truth depth.

Next Lecture
Generative Model: Likelihood-Based Model

- Generative modeling is based on representation of probability distribution

- Likelihood-based models:
    - directly learns the distribution's probability density function (PDF) via maximum likelihood estimation method

    - Variational Autoencoder (VAE) is an example of likelihood-based generative model

    - Limitations
        - rely on surrogate objectives to approximate maximum likelihood training
        - require strong restriction on the model architecture for tractable normalization