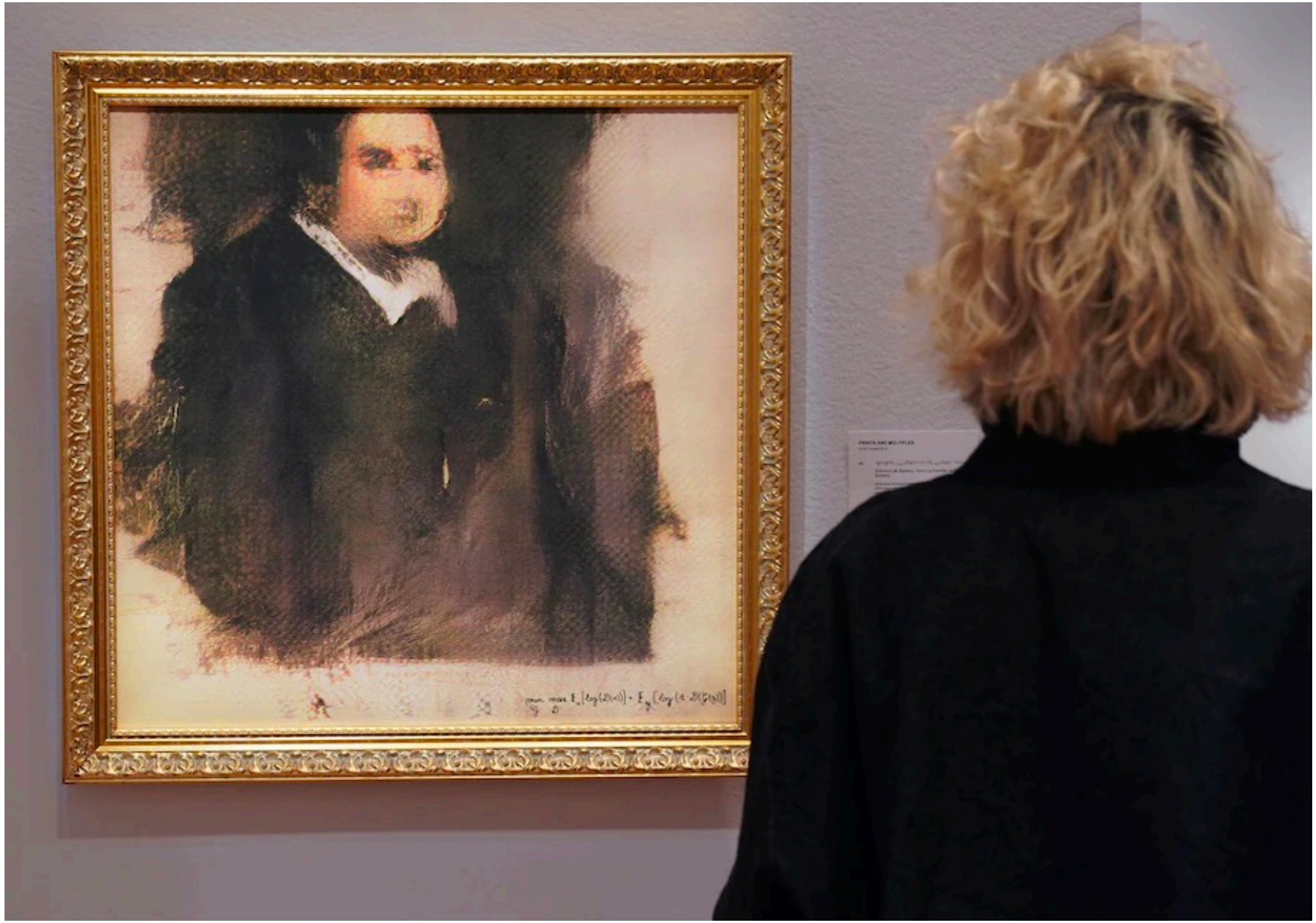
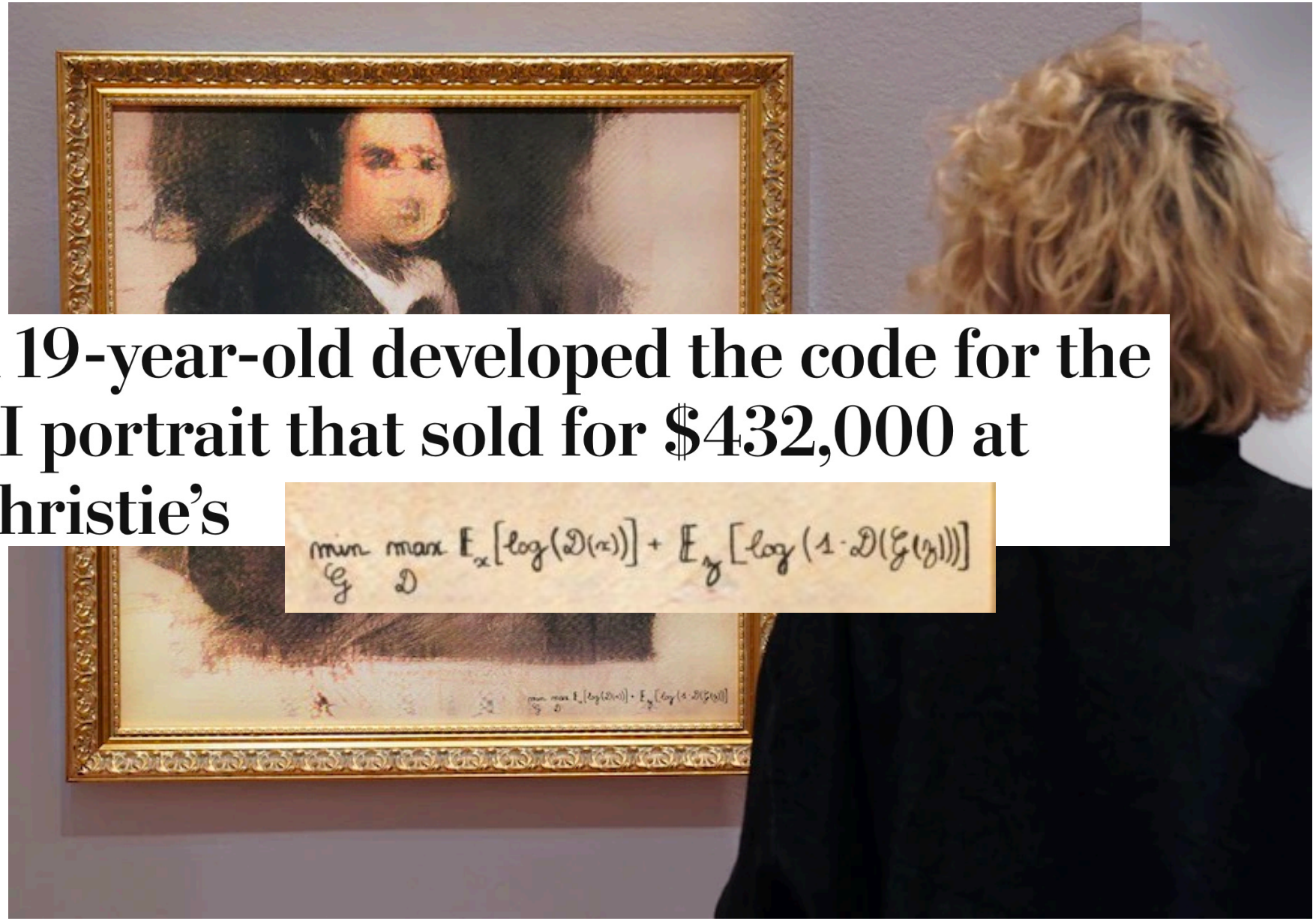


Image Source:

Edmond de Belamy, from La Famille de Belamy (link) https://www.washingtonpost.com/nation/2018/10/26/year-old-developed-code-ai-portrait-that-sold-christies/



A 19-year-old developed the code for the AI portrait that sold for $432,000 at Christie's

$$\min_G \max_D \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1 - D(G(z)))]$$

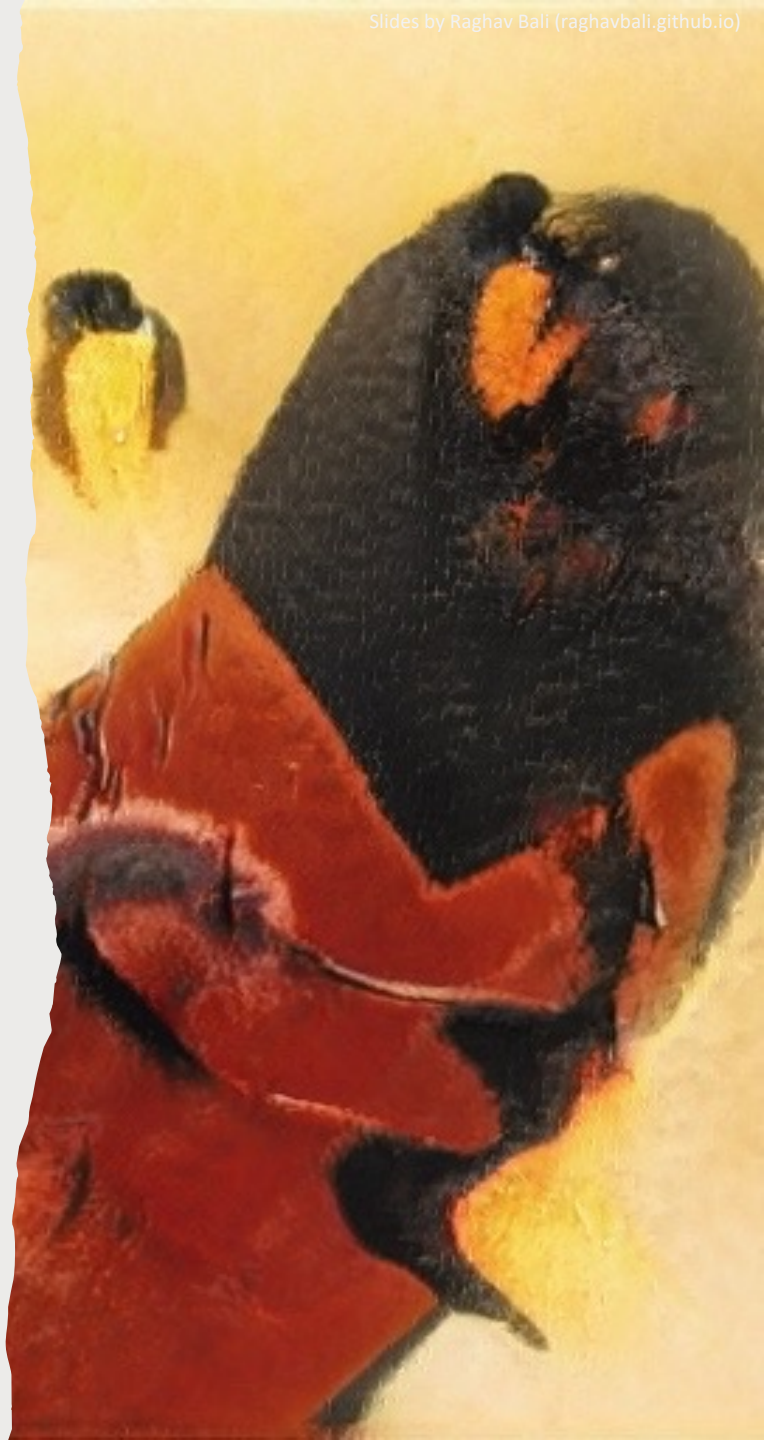
Image Source:

Edmond de Belamy, from La Famille de Belamy (link) https://www.washingtonpost.com/nation/2018/10/26/year-old-developed-code-ai-portrait-that-sold-christies/

Generating Images with GANs

Raghav Bali

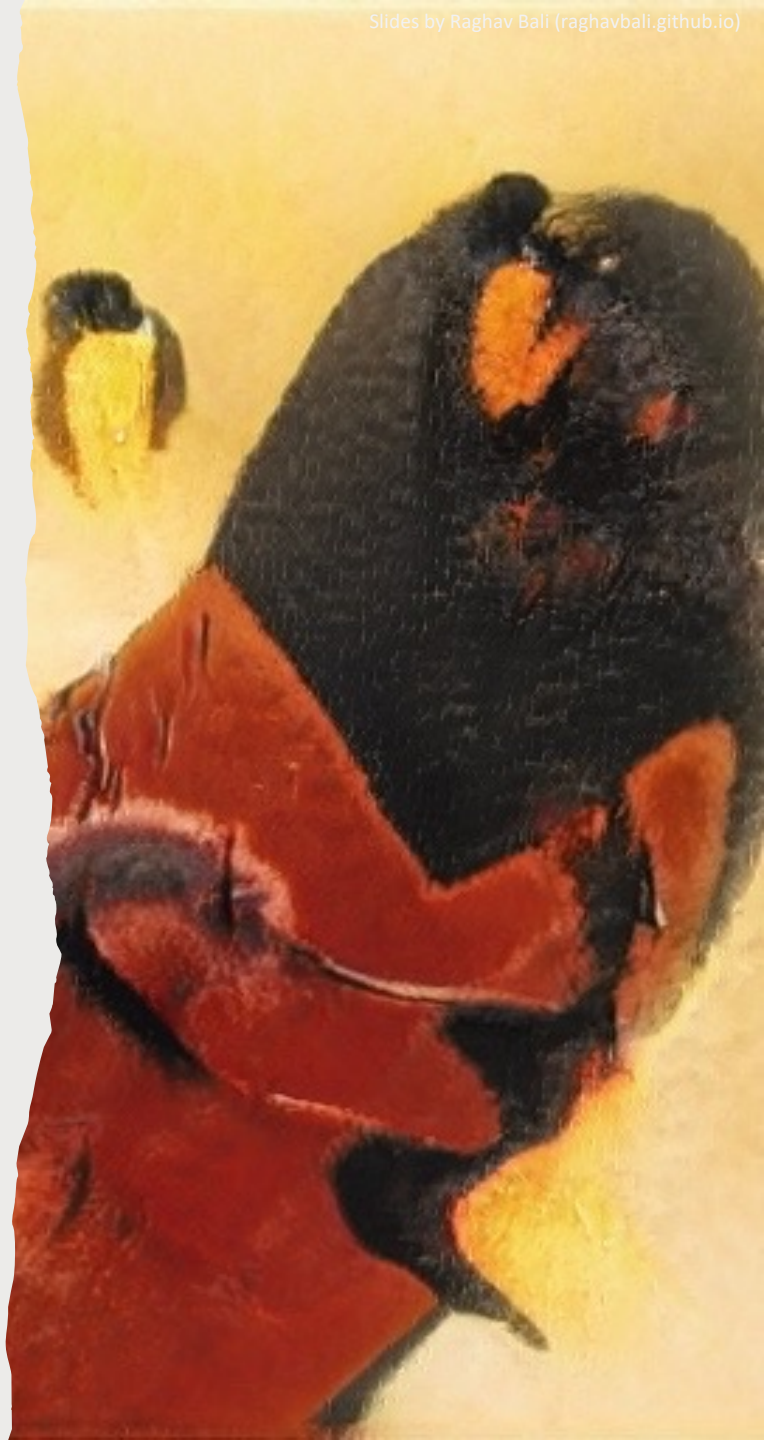
Image Sources: https://thisartworkdoesnotexist .com/



Agenda

- About Me
- Generative Modeling
- Generative Adversarial Networks
- Generate Images : Hands-on
- Challenges
- Q/A

Image Sources: https://thisartworkdoesnotexist .com/



Raghav Bali

- Staff Data Scientist at DeliveryHero (Berlin)
- A decade's experience involving research & development of enterprise level solutions based on Machine Learning, Deep Learning, Computer Vision, NLP, Generative Models and Augmented Reality for real world use-cases.

Delivery Hero
2022 - Present

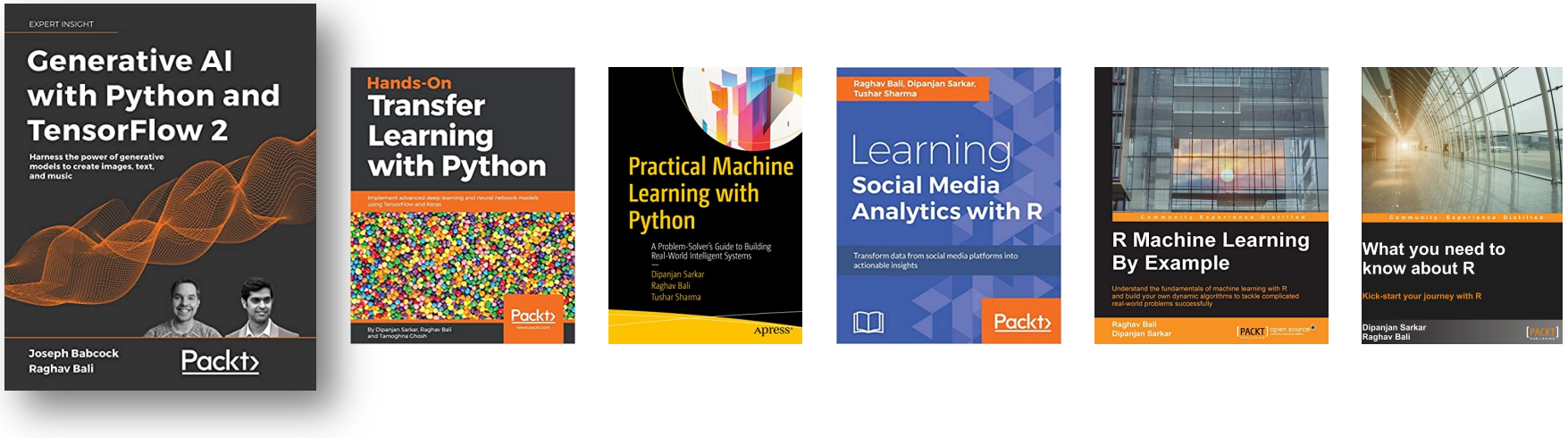
Optum
2017 - 2022

Intel
2015 - 2017

American Express
2014 - 2015

Raghav Bali : Publications

- Talks/Workshops at leading conferences such as Analytics Vidhya, SpringBoard, ODSC, etc.
- Patents: 7 patents in the field of healthcare, deep learning, machine learning and NLP
- Papers
  - Preprint 2022: Easter2.0: Improving Convolutional Models for Handwritten Text Recognition
  - IEEE SSCI 2021, An Interpretable Deep Learning System for Automatically Scoring Request for Proposals
  - IEEE ICTAI 2021, Exclusion and Inclusion--A model agnostic approach to feature importance in DNNs
  - CAIAC 2021, EASTER: Simplifying Text Recognition using only 1D Convolutions
  - CAIAC 2021, A Simple and Interpretable Predictive Model for Healthcare
  - IEEE SmartData 2016, Real Time Failure Prediction of Load Balancers and Firewalls
- Books



Slides by Raghav Bali (raghavbali.github.io)

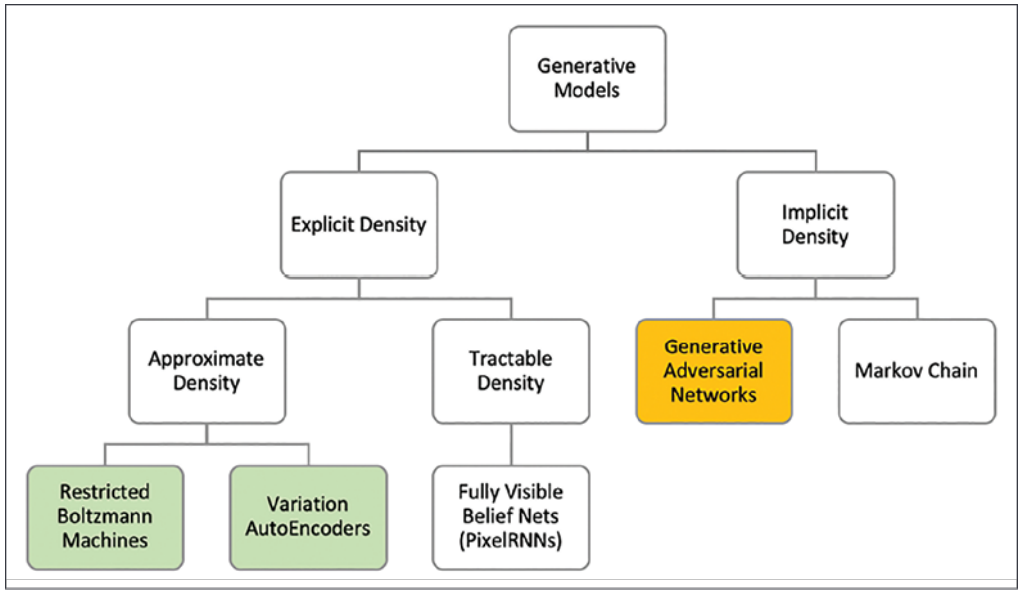
Generative Modeling

- Generative models are a class of models in the unsupervised machine learning space
- These help us model the underlying distributions responsible for generating the dataset under consideration

> “There is tremendous amount of data easily accessible in most cases. Generative models analyze and understand such datasets”

Slides by Raghav Bali (raghavbali.github.io)

Generative Modeling: Topology

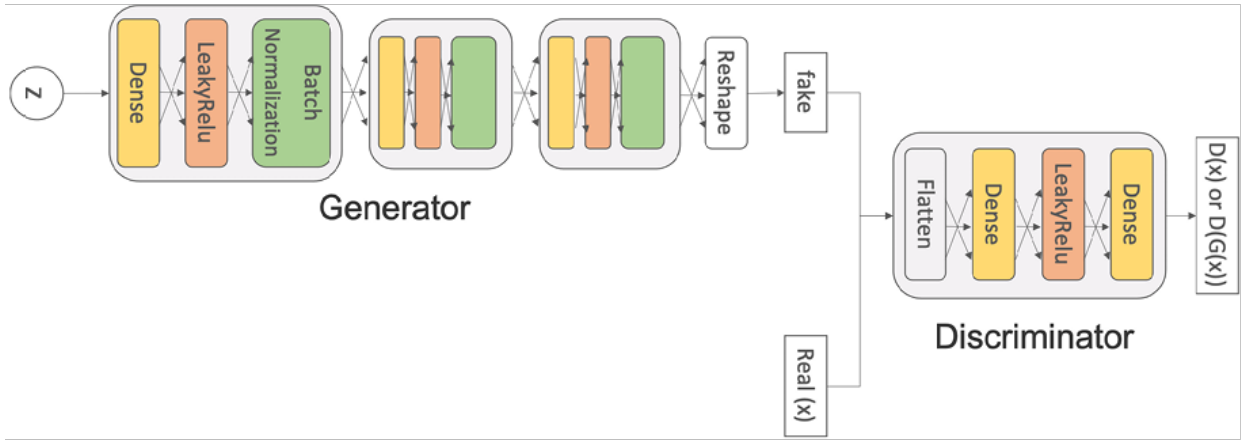


- **Explicit Density**
  - Define an underlying probability distribution function, $p\theta$
  - Objective is to increase the maximum likelihood of sampling from this distribution
- **Implicit Density**
  - Define methods to draw samples from underlying probability distribution function (without defining it explicitly)

Slides by Raghav Bali (raghavbali.github.io)

Generative Adversarial Networks

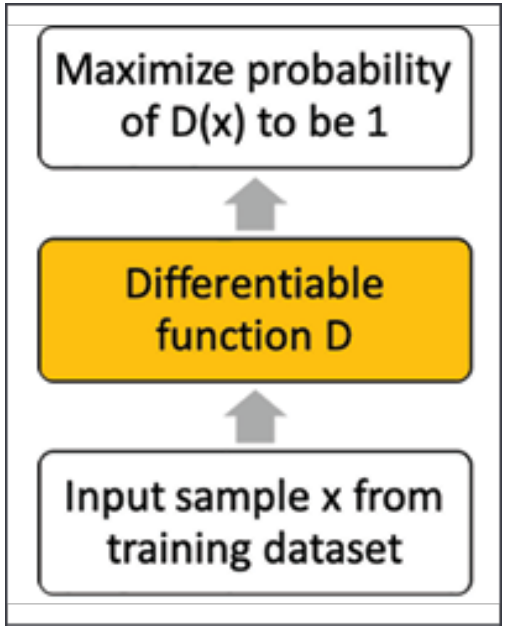
- GANs are implicit density functions which sample directly from the underlying distribution.
- The adversaries compete against each other under well-defined reward functions and each player tries to maximize its rewards



Discriminator **Vs** Generator

Slides by Raghav Bali (raghavbali.github.io)

GAN: Discriminator

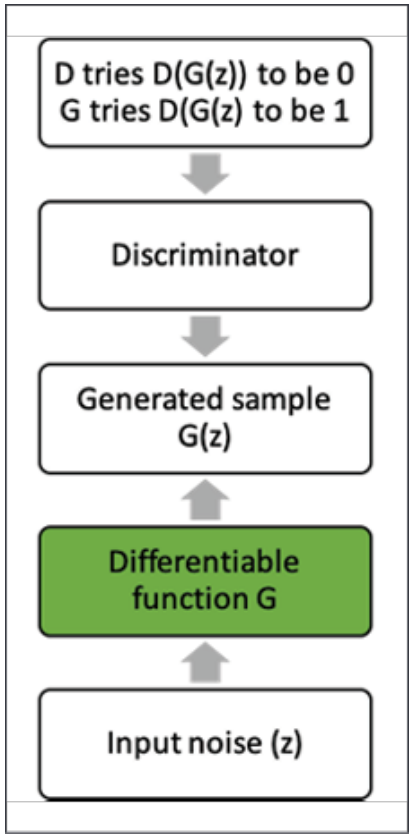


- A differentiable function that tries to **maximize a probability of 1** for samples drawn
from the training distribution.
- The discriminator is also used to classify whether the output from the generator is **real or fake**.
- We denote the discriminator model as D and its output as $D(x)$.

Slides by Raghav Bali (raghavbali.github.io)

GAN: Generator

- This model generates samples that are intended to resemble the samples from our training set.
- The model takes random unstructured noise as input (typically denoted as *z*) and tries to create a varied set of outputs.
- We denote the generator as *G* and its output as *G*(*z*).
- We typically use a lower-dimensional *z* as compared to the dimension of the original data, *x*, that is, $z_{dim} \leq x_{dim}$



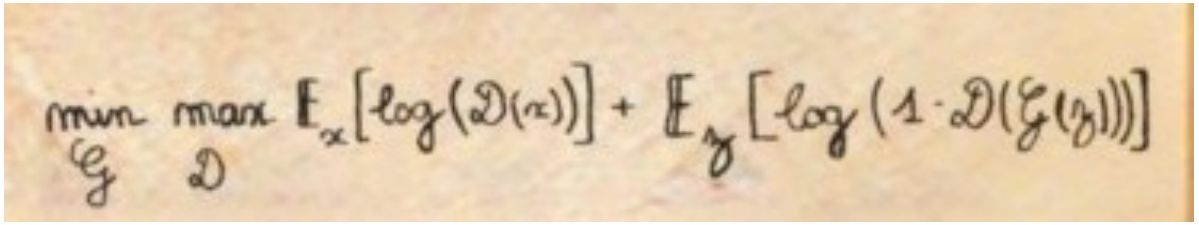
Slides by Raghav Bali (raghavbali.github.io)

GAN: Training

- Training a GAN is like playing this game of two adversaries.
- More formally, this is termed as the minimax game, where the value function *V*(*G*, *D*) is described as follows:

$$min_G max_D V(G,D) = E_{x \sim p_{data}} log\ log\ D(x) + E_{z \sim p_z} log\ log\ (1 - D(G(z)))$$

Discriminator Loss **Generator Loss**



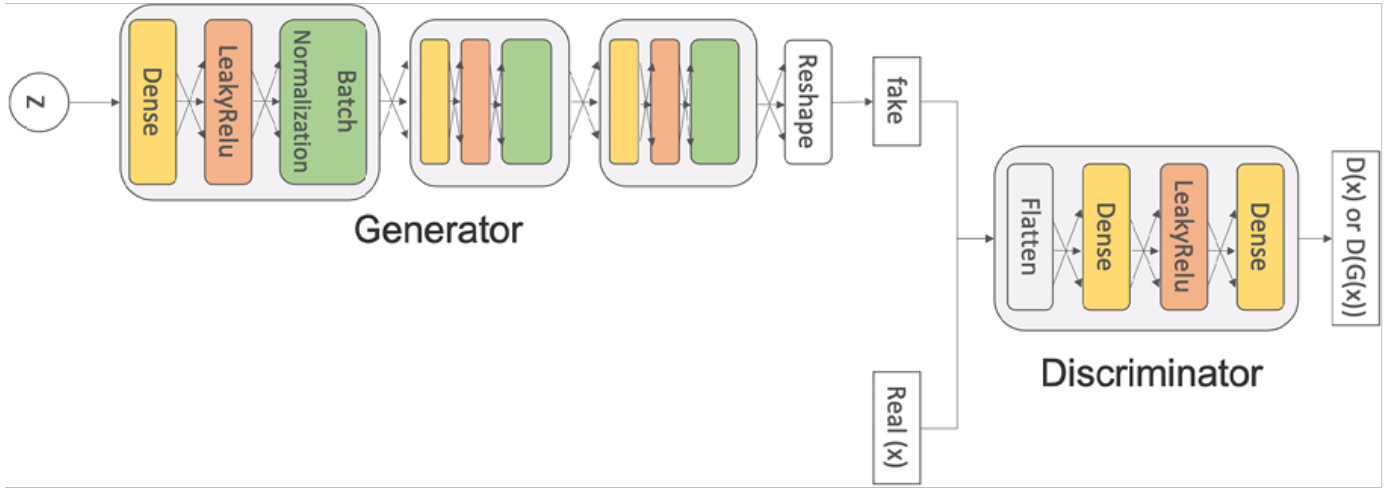
Slides by Raghav Bali (raghavbali.github.io)

Hands-on

- Generate Images

GitHub Repo

Scan Me



Generative Modeling: Applications

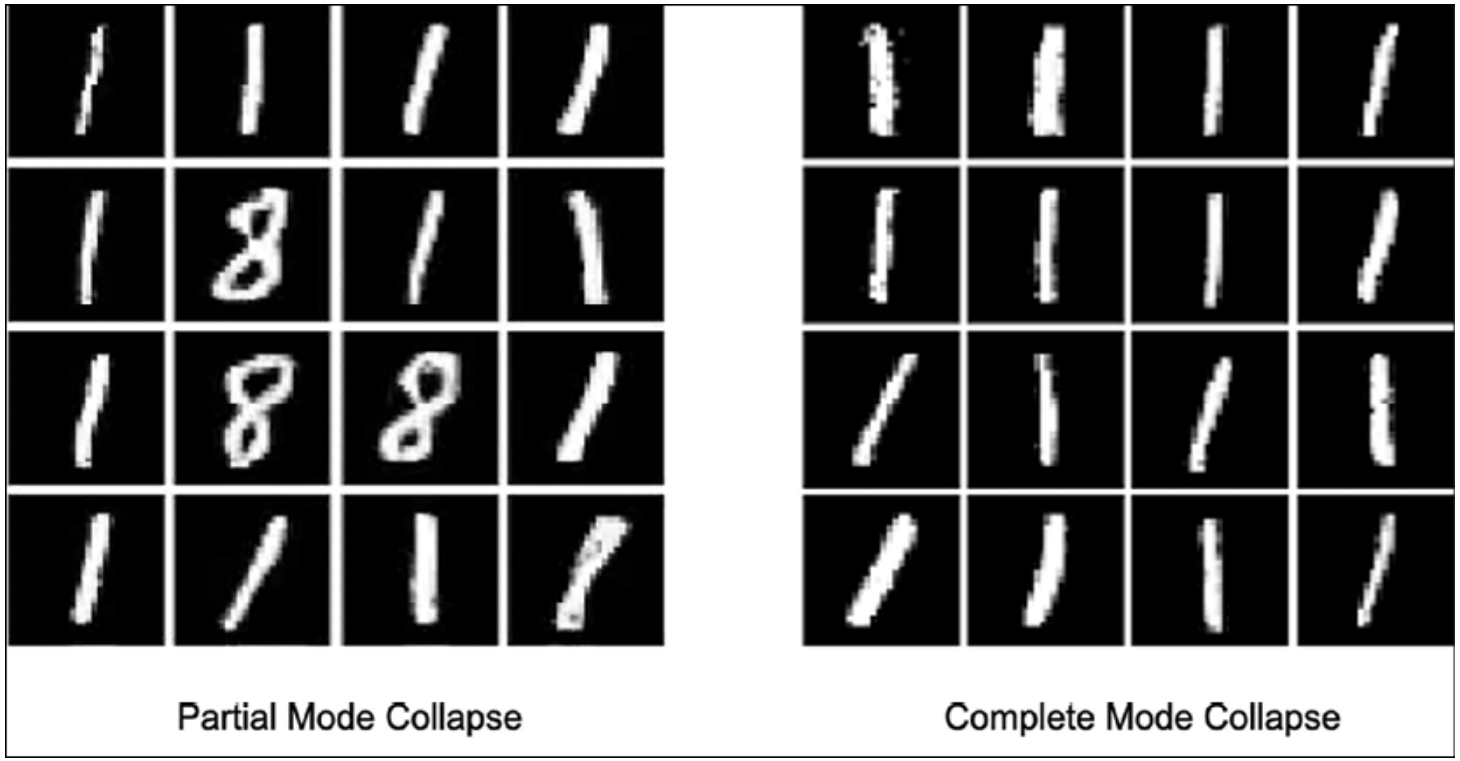
- Productive Applications
  - Image Denoising
  - In-painting
  - Super-resolution
  - Music generation
  - Text generation
  - Style transfers… and many more

- Malicious Applications
  - Pornography & Identity theft
  - Fake-News/Misinformation spread

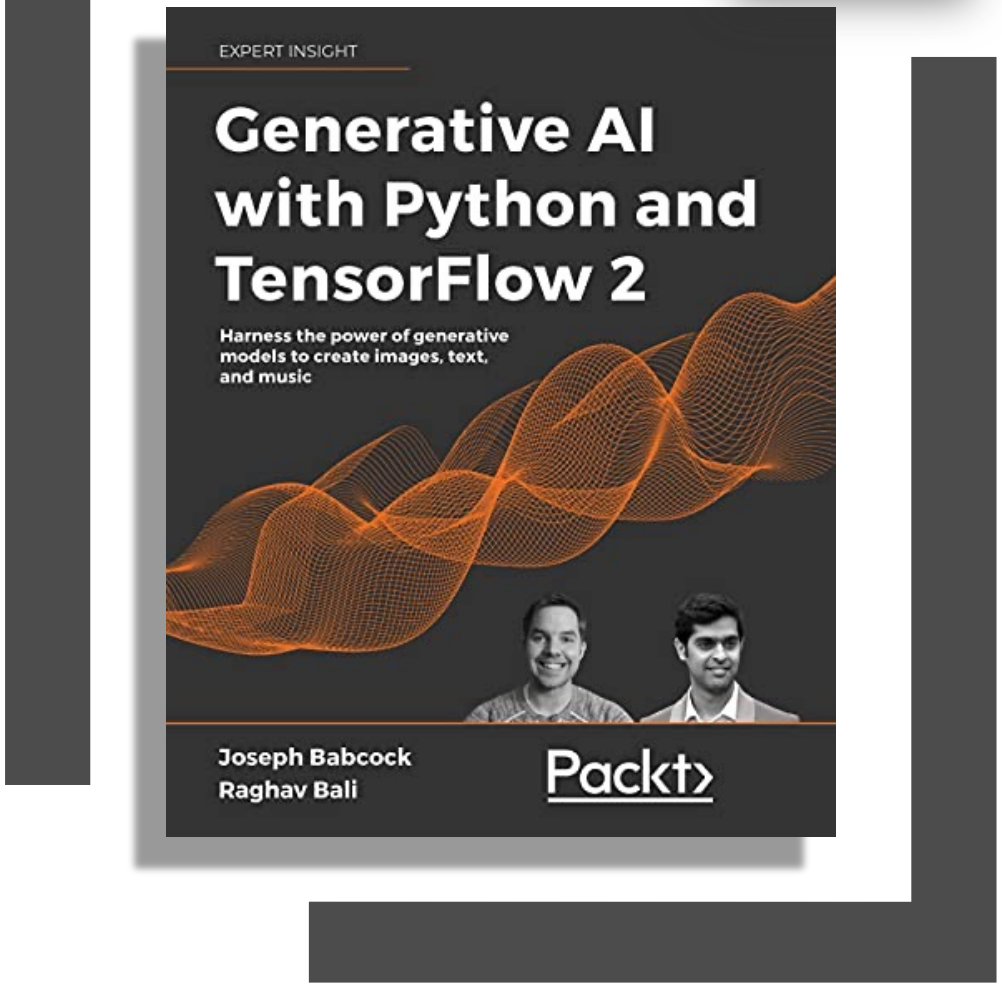
Slides by Raghav Bali (raghavbali.github.io)

Challenges

- Training Instability
- Mode Collapse
- Uninformative Loss



Book Link



Q/A

For more such interesting and usable generative models, refer to
"**Generative AI with python and TensorFlow 2**"

Connect with me On LinkedIn