# CS725 - Final Project Evaluation

Project Title - Exploring Generative Adversarial Networks for Doodling

Ashita Saxena - 21Q050009
Pritish Chakraborty - 21Q050011
Ashutosh Sathe - 21Q050012
Swaroop Nath - 21Q050014

November 22, 2021

# Task

# Task Description

## Task

- We aim to explore use of GANs for creative sketch generation
- Specifically, we plan to reproduce the results of DoodlerGAN [Ge et al., 2020]
- We provide a comparison between DoodlerGAN and StyleGAN2 [Karras et al., 2020] which is a state-of-the-art in image generation on following qualities:
  1. Generation quality
  2. Generation diversity
  3. Complexity of the model

# Dataset

# Dataset Used

- Creative Birds dataset was presented by [Ge et al., 2020]
- It contains 8067 sketches of birds
- Sketches are available as a set of individual parts where each individual part is an SVG stroke
- Sketches also include "initial stroke" which doesn't belong to any part by default. This is also available as an SVG stroke

**Figure 1:** Sample sketches from Creative Birds dataset

# Features of the dataset

Each bird sketch is divided into following parts:

1. eye
2. head
3. body
4. beak
5. legs
6. wings
7. mouth
8. tail

Not all birds will have the same set of parts !

# Models

# Generative Adversarial Nets

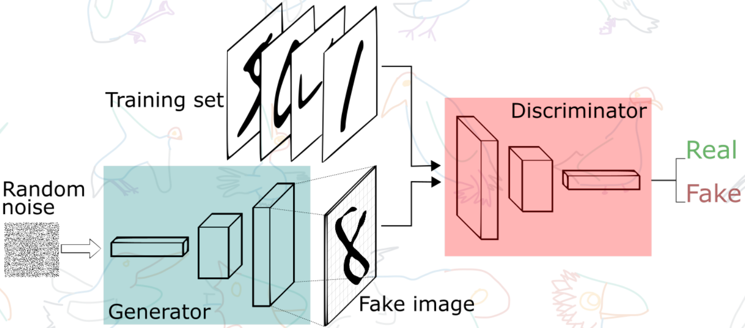



**Figure 2:** GAN framework presented by [Goodfellow et al., 2014], image from [Silva, 2017]

$$\min_{\theta_G} \max_{\theta_D} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_D}(x) + \mathbb{E}_{z \sim p(x)} \log(1 - D_{\theta_D}(G_{\theta_G}(z))) \right] \tag{1}$$

## Main techniques used

- DoodlerGAN
  - A part-based GAN that sequentially generates one part at a time
  - DoodlerGAN automatically determines the order of parts conditioned on the partial sketch generated so far
- Unconditional StyleGAN2
  - Vanilla StyleGAN2 generator trained on the dataset
- Progressive GAN [Karras et al., 2018] like approach to generate sketches at higher resolutions
  - StyleGAN architecture trained in a progressive fashion with the resolution of generated images gradually increasing from 4 x 4 to 16 x 16.

# Approach 1 - DoodlerGAN

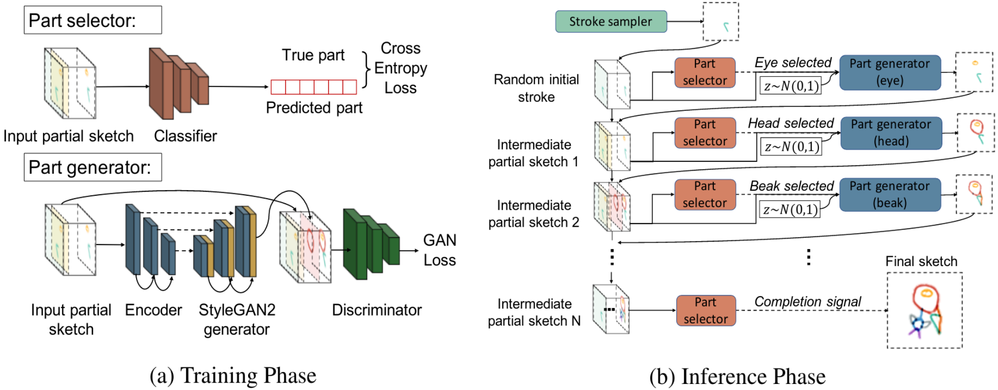



**Figure 3:** Architecture of DoodlerGAN

# Approach 1 - DoodlerGAN

## Generating training data

- For every input sketch, we have (1) initial stroke (2) 1 SVG stroke for each part
- Split the set of parts randomly into 2 sets (say $A$ and $B$) [1]
- Now $(\{\text{Stroke}_{\text{init}}\} \cup \{\text{Stroke}_k | k \in A\}, \text{Stroke}_{B[0]})$ can be used as a training example for the first part of $B$. Moreover, $(\{\text{Stroke}_{\text{init}}\} \cup \{\text{Stroke}_k | k \in A\}, B[0])$ can be used as a training example for part selector
- Also, $(\{\text{Stroke}_{\text{init}}\} \cup \{\text{Stroke}_k | k \in A\} \cup \{\text{Stroke}_{B[0]}\}, \text{Stroke}_{B[1]})$ can be used as a training example for the second part of $B$ and so on
- This way, we have split the original sketch dataset into $n =$ number of parts datasets and 1 dataset for part selector. Now each part generator as well as the part selector can be trained independently

[1] Authors always keep the "eye" as the first part for consistency and training stability reasons i.e. "eye" $\notin B \forall B$ and "eye" $= A[0]$

## Approach 2 - Unconditional StyleGAN2

- StyleGAN2 [Karras et al., 2020] is the current state-of-the-art in image generation
- This approach involves training a vanilla StyleGAN2 architecture on the given dataset and analyse the quality of generation
- The training is done by rasterizing each sketch at resolution of 32x32
- Note that despite being known for high quality generation, this approach is limited by how much data it's trained on
- DoodlerGAN can modify sketch on-the-go using human supervision, StyleGAN2 approach does not allow for this
- Despite this shortcoming, can a vanilla StyleGAN2 generate interesting sketches ? Or will the diversity of the generated sketches will be poor ?

# Approach 3 - Higher resolution sketch generation using StyleGAN (progressively)

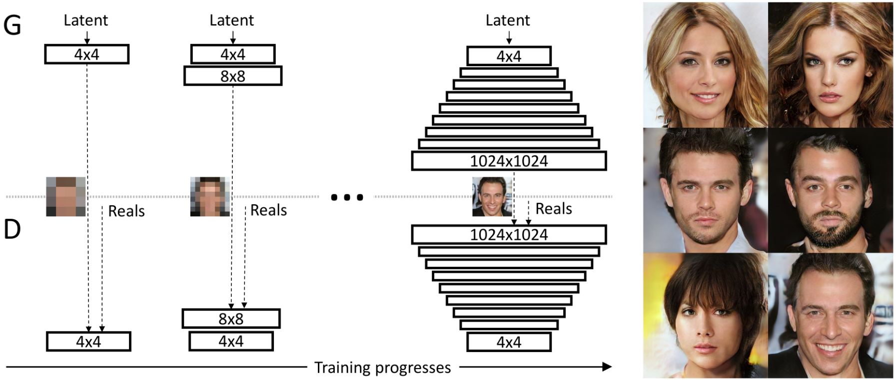



**Figure 4:** ProgressiveGAN [Karras et al., 2018] architecture. Can we generate a higher resolution sketch using the same latent space ? Can we generate more diverse sketches if we condition at multiple resolutions ?

# Hyperparameters for both approaches

- Image size 32 x 32 with stroke size of 2 pixels
- 55000 training steps (batch size 160)
- Learning rate $1 \times 10^{-4}$
- Data Augmentation :
  - Random affine transformations :
    - Random rotation $\pm 15^{\circ}$
    - Random scaling $\pm 10\%$ of the width or height
    - Random translation $\pm 1\%$ of the width or height
  - Random horizontal flip with $p = 0.5$

# Results

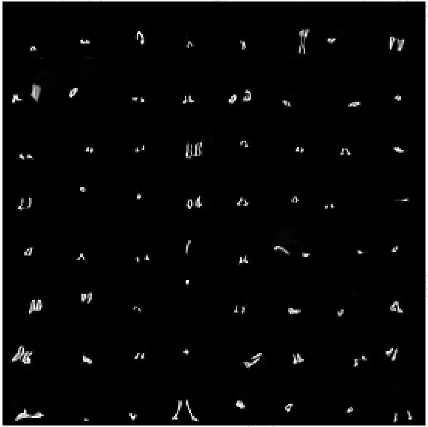
## Generation quality

**(a)** Sketches generated by DoodlerGAN (stroke width increased for better visualization)

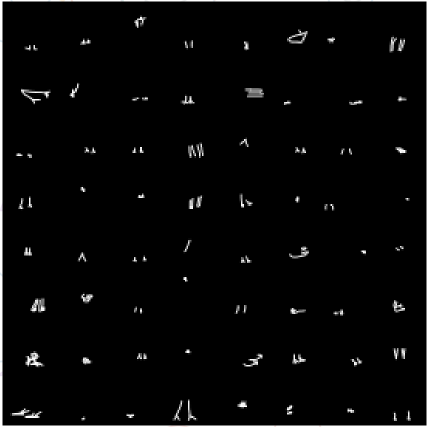
**(b)** Sketches generated by StyleGAN2 (single-step, unconditional generation)

**Figure 5:** Comparing generation quality of both the models

# Training DoodlerGAN



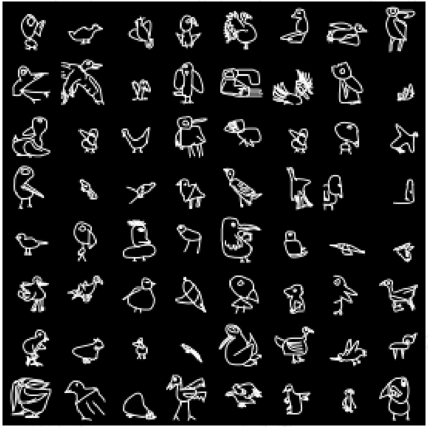
**(a)** Legs part generated by the generator



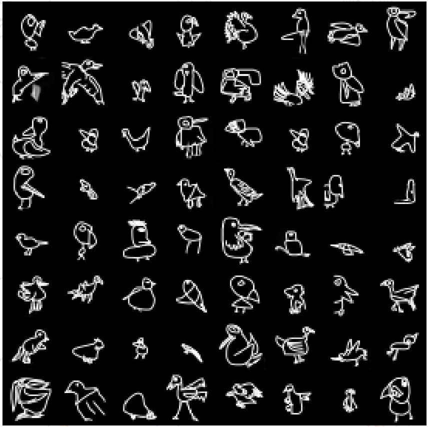
**(b)** Legs images from the real dataset

**Figure 6:** Legs Images from the Creative Birds Dataset - Part I

# Training DoodlerGAN



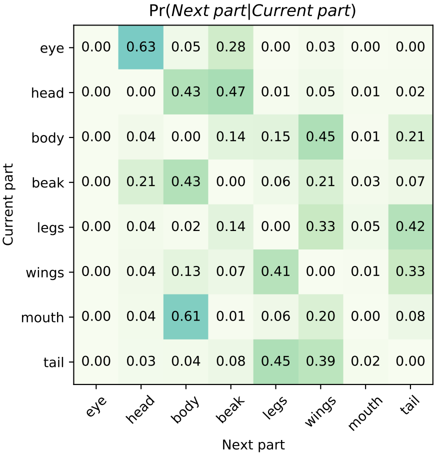
**(a)** Combination of generated parts with rest of sketch



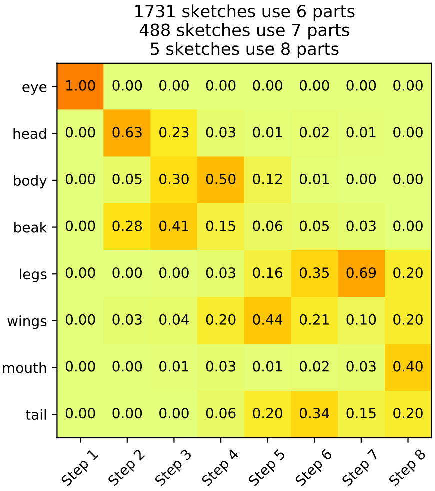
**(b)** Full sketch (with legs)

**Figure 7:** Legs Images from the Creative Birds Dataset - Part II

# DoodlerGAN part selector statistics




**(a)** Part Conditional Probability Matrix




**(b)** Stepwise Part Distribution Matrix

# Parts generated at every stage using DoodlerGAN

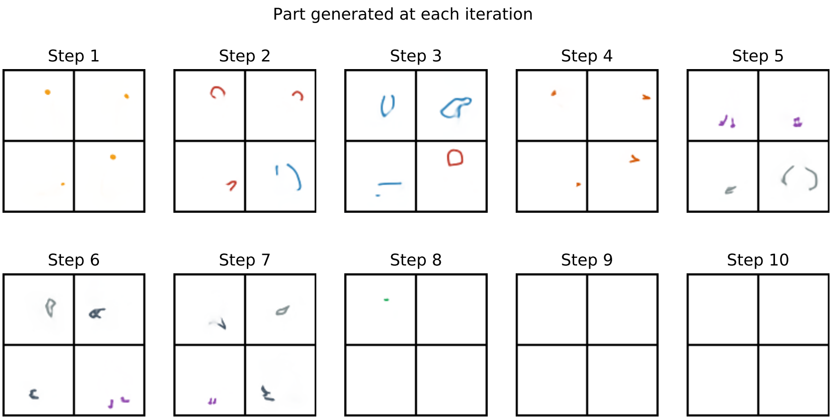



**Figure 9:** Parts generated at every stage using DoodlerGAN. Notice that only one of the sketch needed to generate a part at stage 8 while no sketch generated parts in stages 9 and 10

# Evolution of sketch at every stage using DoodlerGAN

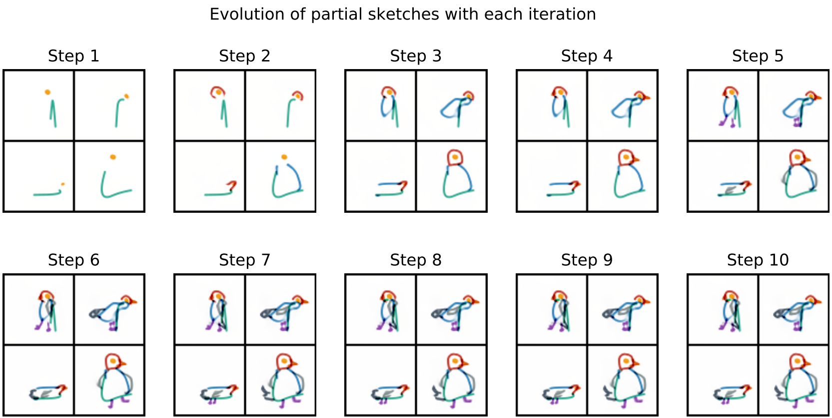



**Figure 10:** Evolution of sketches at every stage using DoodlerGAN. Notice that no sketch changed during stages 9 and 10 since no new part was generated

# Metrics

- Frechet Inception Distance [Heusel et al., 2017] : (Lower is better)
  1. DoodlerGAN - 9.044
  2. Vanilla StyleGAN2 - 17.804
- Generation Diversity (Average distance between generated images) : (Higher is better)
  1. DoodlerGAN - 174.35
  2. Vanilla StyleGAN2 - 98.92

# Source code + Demo

- Link to source code: `https://github.com/ashutoshbsathe/cs725_project`
- Link to demo: `https://github.com/ashutoshbsathe/cs725_project/blob/main/demo/cs725_project_demo.mp4`[2]
- Acknowledgements:
  - DoodlerGAN authors for providing DoodlerGAN code as well as pre-processed dataset - `https://github.com/facebookresearch/DoodlerGAN`
  - Phil Wang for StyleGAN2 implementation in PyTorch - `https://github.com/lucidrains/stylegan2-pytorch`
  - GetsEclectic for PyTorch implementation of FID - `https://github.com/GetsEclectic/pytorch-fid`

[2] Google drive mirror can be found here

# Work Distribution

1. **Literature Review**: Ashita and Pritish
2. **Training of DoodlerGAN Parts**: Team effort, with various bird parts divided between team members.
3. **ProgressiveGAN approach**: Swaroop
4. **Vanilla StyleGAN2 approach**: Ashutosh
5. **Analysis of DoodlerGAN and vanilla StyleGAN2**: Ashutosh
6. **Report Preparation**: Ashita and Pritish

**Thank You**

Ge, S., Goswami, V., Zitnick, C. L., and Parikh, D. (2020).
**Creative sketch generation.**

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
**Generative adversarial nets.**
In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017).
**Gans trained by a two time-scale update rule converge to a local nash equilibrium.**
*Advances in neural information processing systems*, 30.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018).
**Progressive growing of gans for improved quality, stability, and variation.**
In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). **Analyzing and improving the image quality of stylegan.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119.
- Silva, T. (2017). **A Short Introduction to Generative Adversarial Networks.** https://sthalles.github.io/intro-to-gans/.