Sketch to Image Translation Using GANs

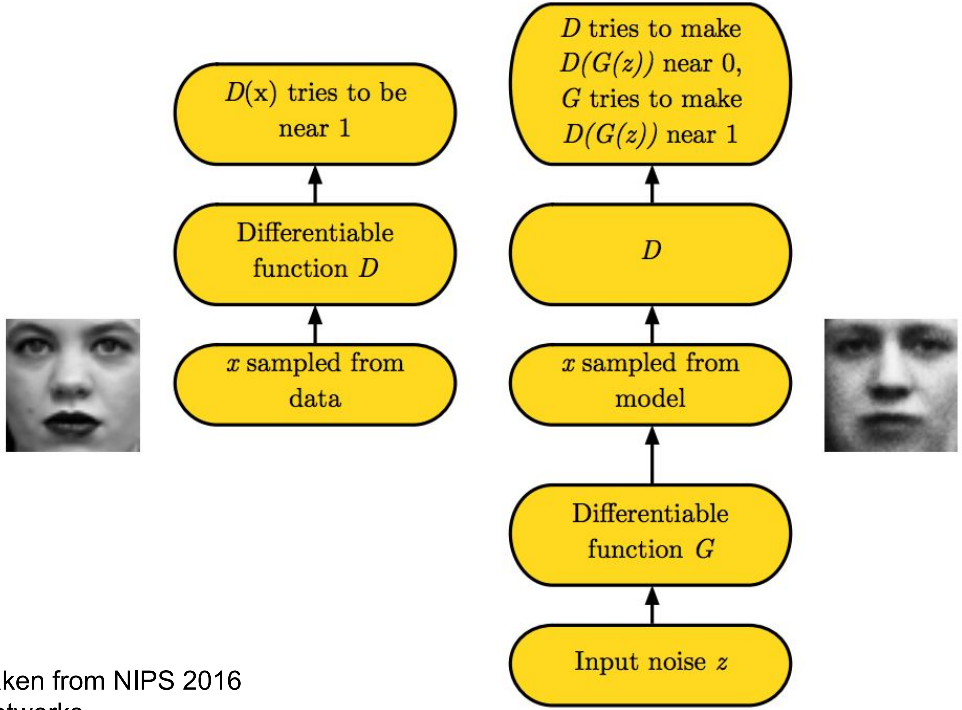
Featuring: 2N Softmax Discriminator

Lisa Fan, Jason Krone, Sam Woolf
COMP150DL-Tufts
Spring 2017

Overview

1. Brief Review
2. Related Work
3. Problem Statement
4. Architecture & Experiments
5. Evaluation
6. Future Work

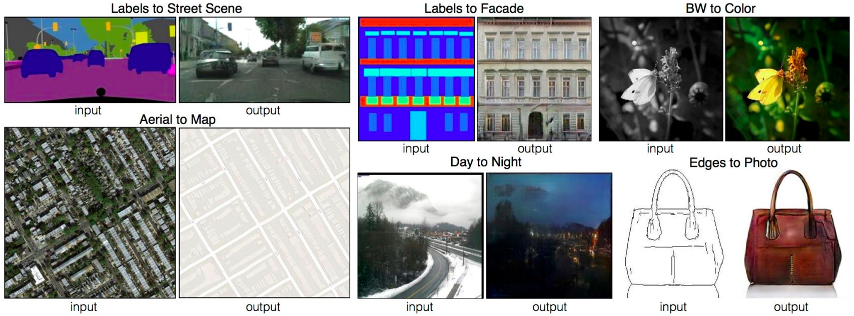
Brief Review



Credit: Figure by Ian Goodfellow taken from NIPS 2016
Tutorial: Generative Adversarial Networks

Related Work

Image-to-Image Translation
with Conditional Adversarial Networks
Isola, et al, 2017



Improved Techniques for Training GANs
Salimans, et al, 2016

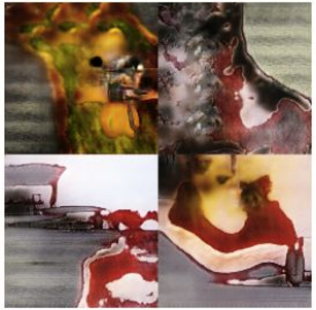
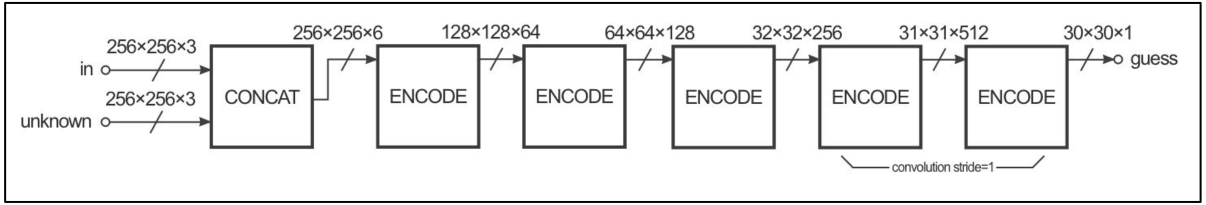
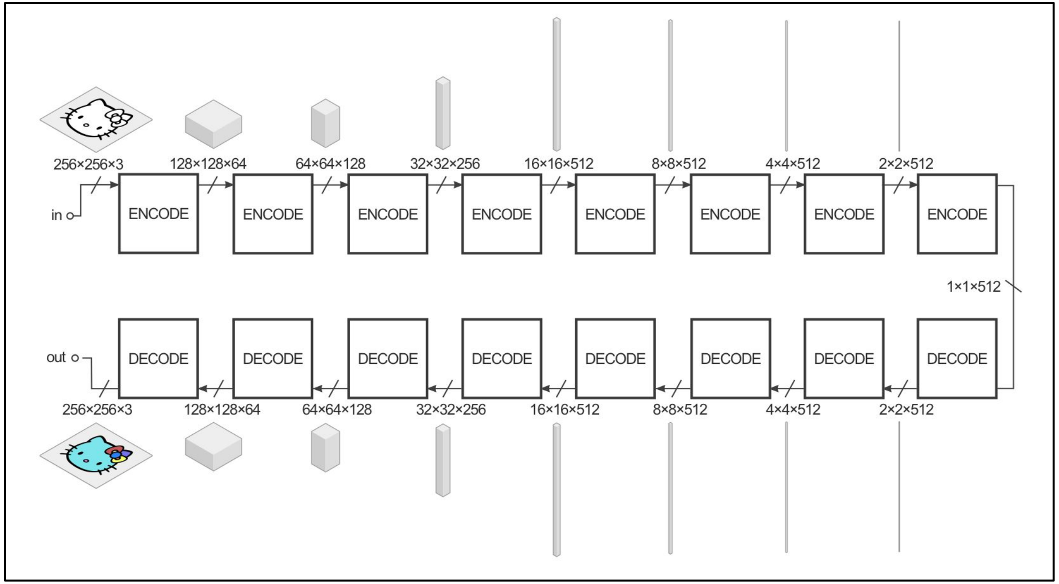


Image-to-Image Translation

Discriminator



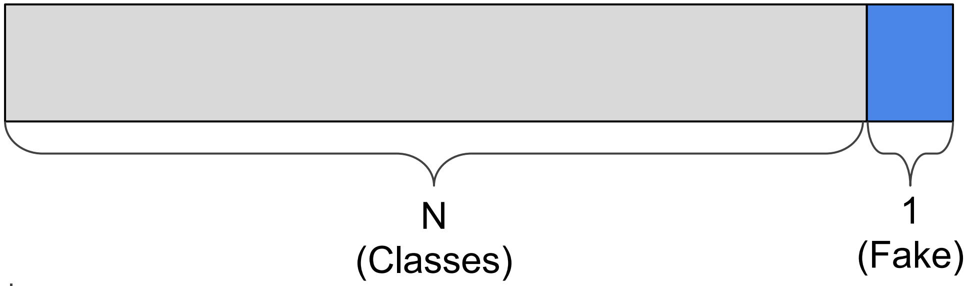
Generator



Credit: Figures by Christopher Hesse
taken from https://affinelayer.com/pix2pix/

Improved Techniques for Training GANs

$$
\begin{aligned}
L &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y=K+1|\boldsymbol{x})] \\
&= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where} \\
L_{\text{supervised}} &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)} \log p_{\text{model}}(y|\boldsymbol{x}, y < K+1) \\
L_{\text{unsupervised}} &= -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log[1 - p_{\text{model}}(y=K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G} \log[p_{\text{model}}(y=K+1|\boldsymbol{x})]\},
\end{aligned}
$$



Credit: Equations by Salimans, et al

Problem Statement

Quora

Can Generative Adversarial networks use multi-class labels?

Ian Goodfellow, Lead author of the Deep Learning textbook: http://www.deeplearningbook.org

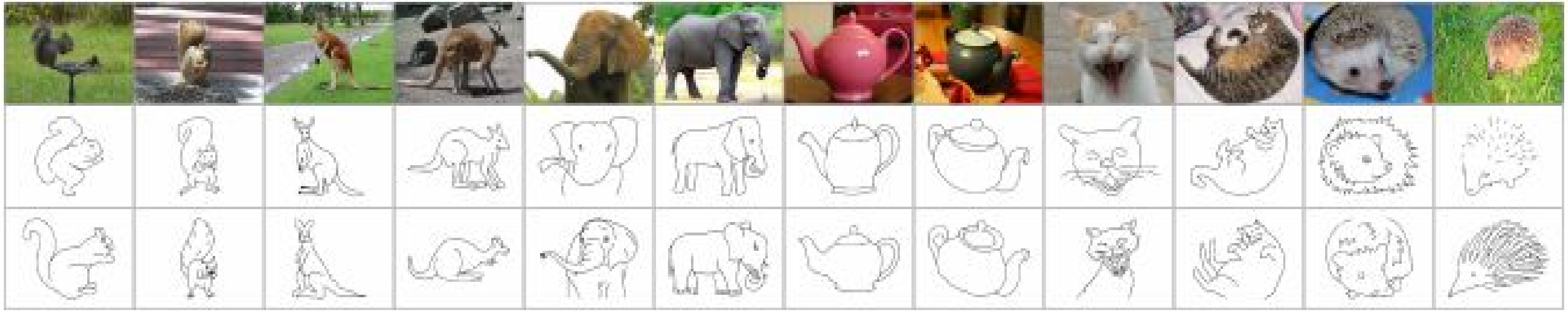
You could also imagine using 2N output classes, with a real and a fake version of each class, e.g. real dog, real cat, fake dog, fake cat. I don't know of anyone who has experimented with that approach yet.

Inspiration

- What, if any, improvement can be attained by using a discriminator with 2N output classes (real and fake scores for each class) for image generation?
- Can cGANs generate photo-realistic images from rough sketches?

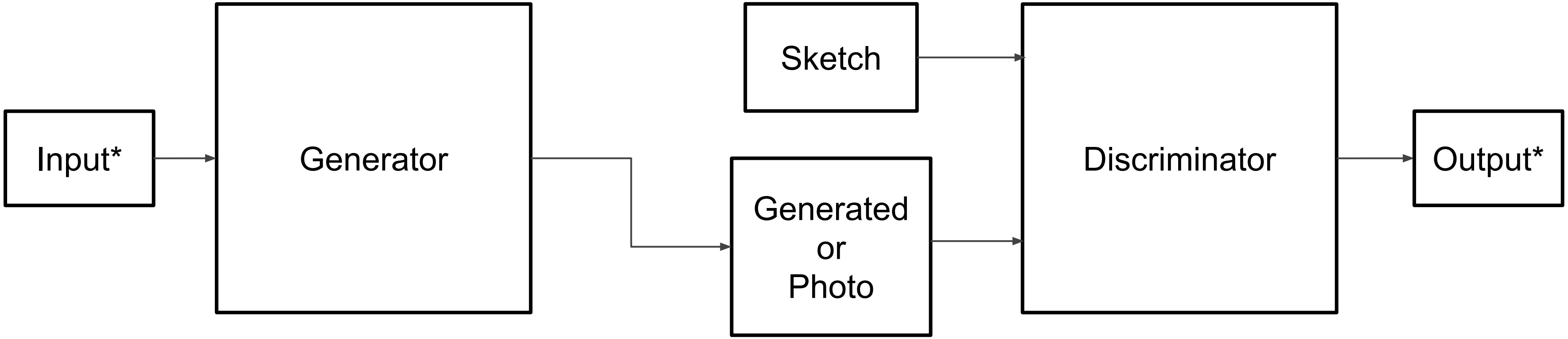
Dataset - Sketchy Database

- Developed by Georgia Tech
- 125 Object Categories (subset of ImageNet)
- 12,500 Photographs
- 75,471 Sketches of Photos



http://sketchy.eye.gatech.edu/

Network Architecture



*Input:
Sketch
or
Sketch + Class

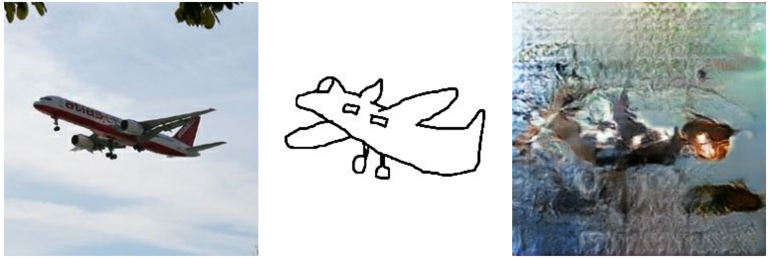
*Output:
PatchGAN
or
2N Softmax

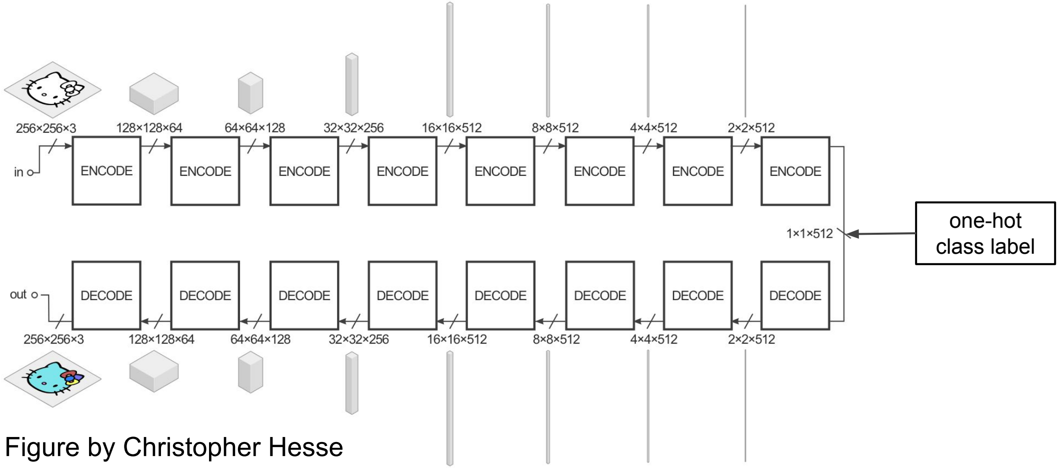
Experiments

1. Baseline: Out of the box *Image-to-Image Translation* model
2. Class conditional Generator
3. 2N output discriminator with cross entropy
4. 2N output discriminator with penalized cross entropy
5. Cropped training images using semantic segmentation

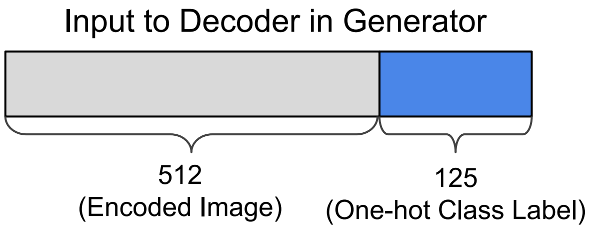
Class Conditional Generator

- Input to Generator is sketch and class label
- One-hot class label is appended to encoded image vector
- Coupled with baseline Discriminator





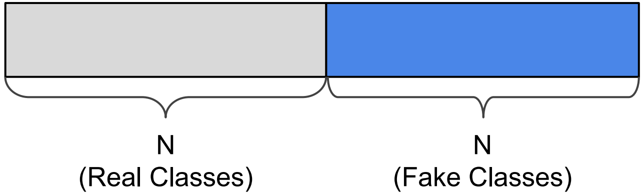
Credit: Figure by Christopher Hesse
taken from https://affinelayer.com/pix2pix/



2N Discriminator with Cross Entropy

- Added FC layer to Discriminator
  to produce 2N-dimensional vector of logits

$$p_{model}(y = j|x) = \frac{\exp(l_j)}{\sum_{i=1}^{2N} \exp(l_i)}$$



$$L_D = -(\mathbb{E}_{x,s,y \sim p_{data}(x,s,y)}[\log p_{model}(y|x, s, y \leq N)] + \mathbb{E}_{s,y \sim p_{data}(s,y)}[\log p_{model}(y|G(s), s, N < y \leq 2N)])$$

$$L_G = -\mathbb{E}_{s,y \sim p_{data}(s,y)}[\log p_{model}(y - N|G(s), s, N < y \leq 2N)] + \lambda\ L1\ (G)$$

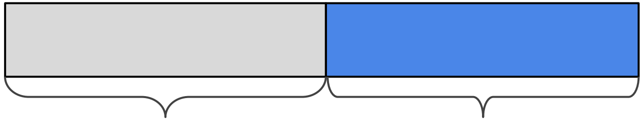
2N Discriminator with Penalty Cross Entropy

- Penalizes errors by type
- Utilizes more information than standard cross entropy

$$L_D = -(\mathbb{E}_{x,s,y\sim p_{data}(x,s,y)}[\log p_{model}(y|x,s,y \leq N)] \times pen(y,\hat{y})$$
$$+ \mathbb{E}_{s,y\sim p_{data}(s,y)}[\log p_{model}(y|G(s),s,N < y \leq 2N)] \times pen(y,\hat{y}))$$

$$pen(y,\hat{y}) = \begin{cases} a; c(y) = c(\hat{y}), \text{is-fake}(y) \neq \text{is-fake}(\hat{y}) \\ b; c(y) \neq c(\hat{y}), \text{is-fake}(y) = \text{is-fake}(\hat{y}) \\ c; c(y) \neq c(\hat{y}), \text{is-fake}(y) \neq \text{is-fake}(\hat{y}) \end{cases}$$

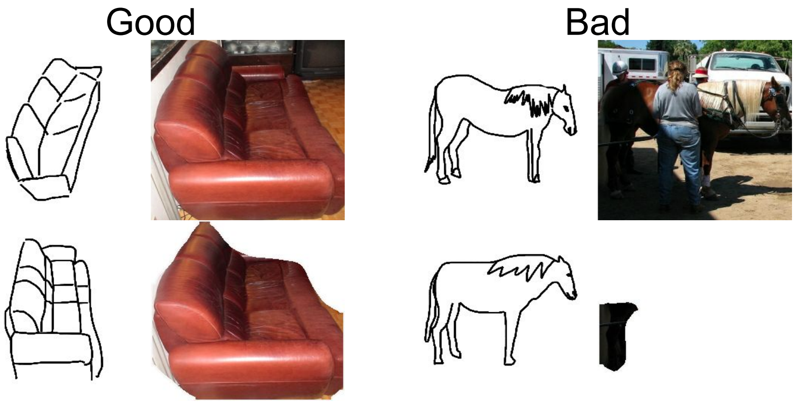
where $a < b < c$



$$L_G = -\mathbb{E}_{s,y\sim p_{data}(s,y)}[\log p_{model}(y-N|G(s),s,N < y \leq 2N)] \times pen(y-N,\hat{y}) + \lambda L1\ (G)$$

Preprocessing: Cropped Training Images

- Applied semantic segmentation mask to remove background in training data
- Used network from "Fully Convolutional Networks for Semantic Segmentation" by Long, et al
- Network pre-trained on PASCAL VOC Segmentation data
- 15 object categories
- Cropped ~9,000 photos
- Generated images using baseline model



https://github.com/warmspringwinds/tf-image-segmentation

Evaluation

Qualitative:

1. Our favorite photos

Quantitative:

1. Stand Alone Discriminator for Classification
2. Inception Score
3. Generated images classified using Inception network

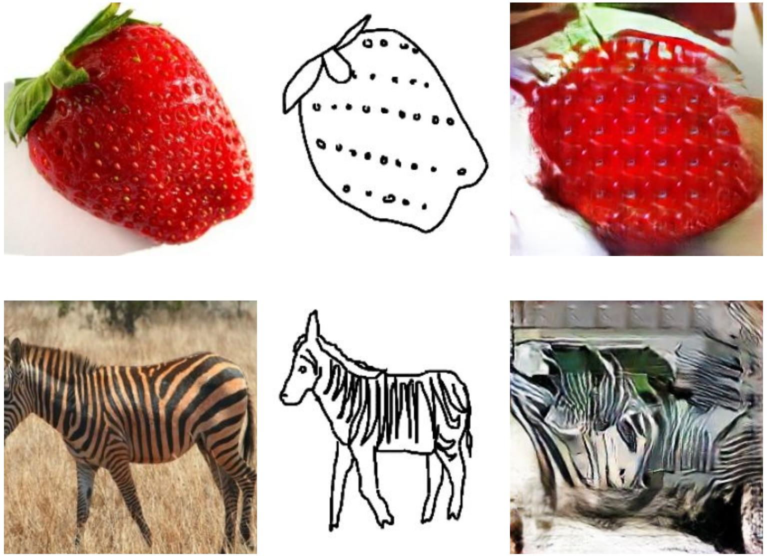
* All evaluated models ran for 50,000 iterations unless otherwise specified

1. Sketches
2. Photos
3. Segmented Photos
4. Baseline
5. Conditional Generator
6. 2N Cross Entropy
7. 2N Penalized Cross Entropy
8. Trained on Segmented Photos

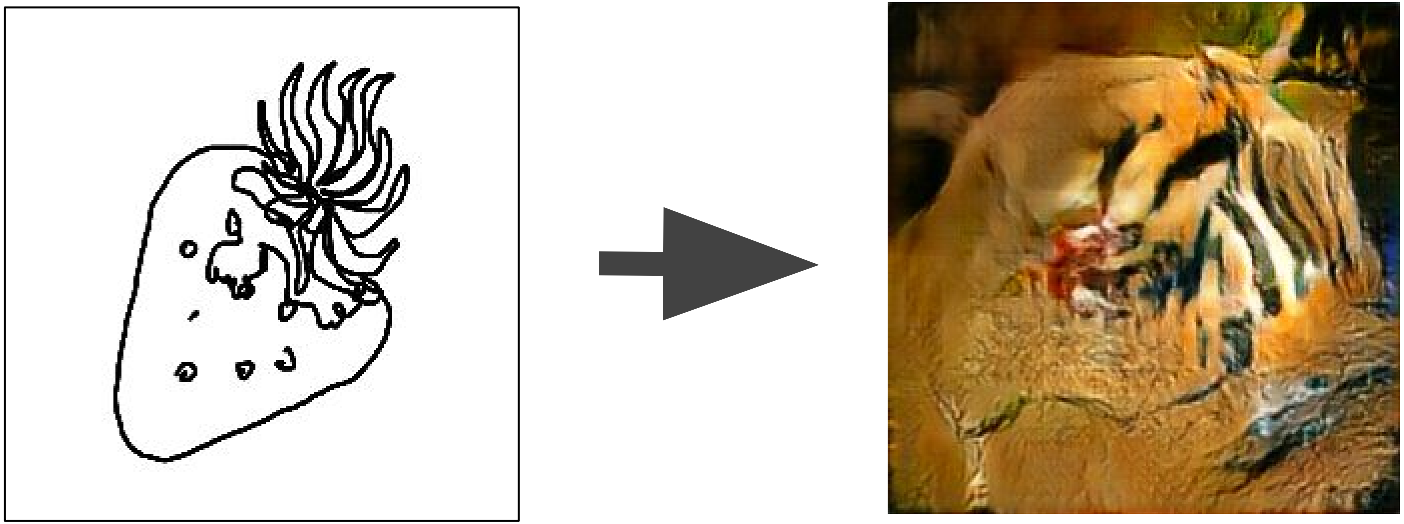
Awesome Generated Images

Original real/fake (83,700 iterations)

Penalty (134,500 iterations)



The Formidable Tiger Strawberry



Stand Alone Discriminator for Classification

- Decoupled Discriminator from GAN framework, used to classify images
- Typically a use case for semi-supervised classification
- For us it is an additional evaluation metric

Model	Accuracy
2N (50,000 iterations)	26.98%
Penalty (50,000 iterations)	29.09%
Penalty (134,500 iterations)	10.26%

Inception Score

- Inception score metric proposed in *Improved Techniques for Training GANs*
- Measure of how "real" a generated set of images look
- Uses pre-trained Inception network to calculate p(y) and p(y|$\boldsymbol{x}$)
- Expects low entropy for conditional class distribution p(y|$\boldsymbol{x}$)
- Expects high entropy for marginal class distribution across all images p(y)
- $\exp(\mathbb{E}_{\boldsymbol{x}} \mathrm{KL}(p(y|\boldsymbol{x}) || p(y)))$
- Previous works:
    - Real images: 11.0 ~ 26.0
    - Generated images: 8.0 ~ 9.0

Pretrained Inception model from:
http://download.tensorflow.org/models/ image/imagenet/inception-2015-12-05.tgz

Inception Score

Model	Mean	Std Dev
Ground Truth Photos	74.81	1.40
Baseline (real/fake)	5.26	0.16
Class Conditional Generator	4.25	0.07
2N Cross Entropy	6.11	0.11
Penalized 2N Cross Entropy	6.20	0.10
Segmented Ground Truth Photos	13.33	0.90
Images Trained on Segmented Photos	5.96	0.25

Classifying Generated Images by Inception

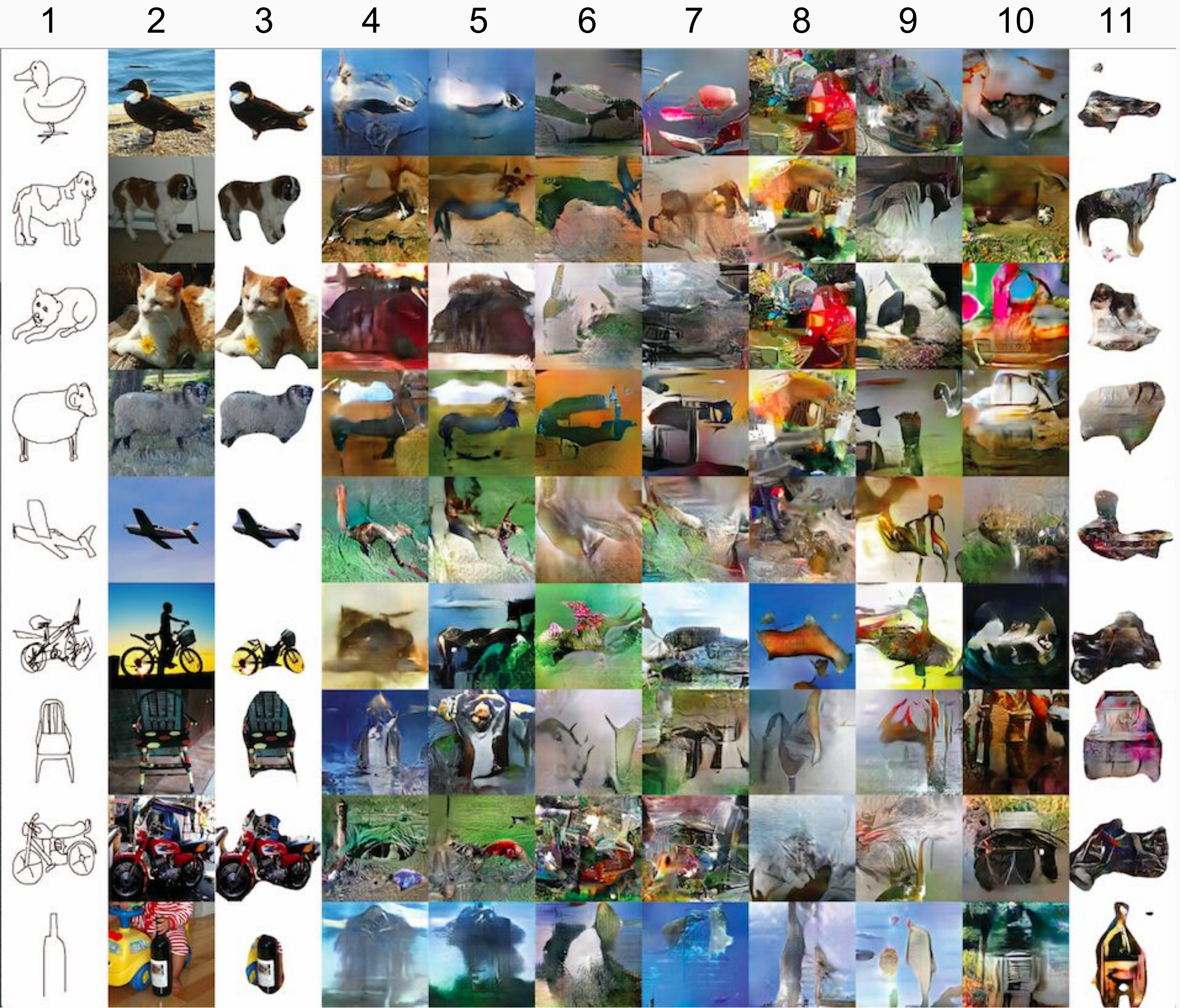
- Measure of how well the classes are generated
- Ran generated images through pre-trained Inception network
- Looked at top 1 and top 5 accuracy

Classifying Generated Images by Inception

Model	Top 1 Accuracy	Top 5 Accuracy
Ground Truth Photos	71.90%	79.04%
Baseline (real/fake)	0.83%	2.36%
Conditional Generator	1.05%	3.13%
2N Cross Entropy	0.48%	1.90%
Penalty	0.85%	2.44%
Segmented Ground Truth Photos	40.58%	60.51%
Images Trained on Segmented Photos	1.99%	04.42%

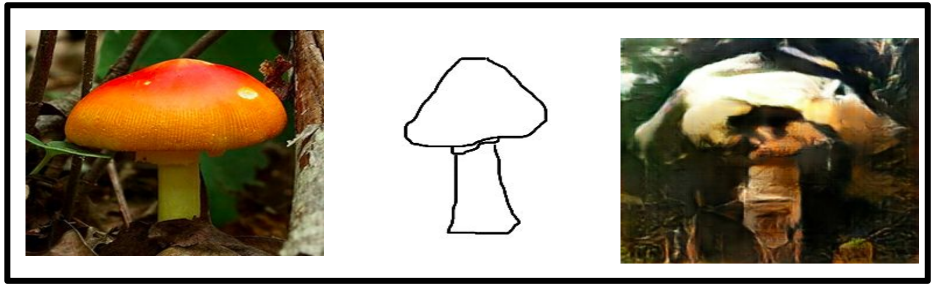
Future Work

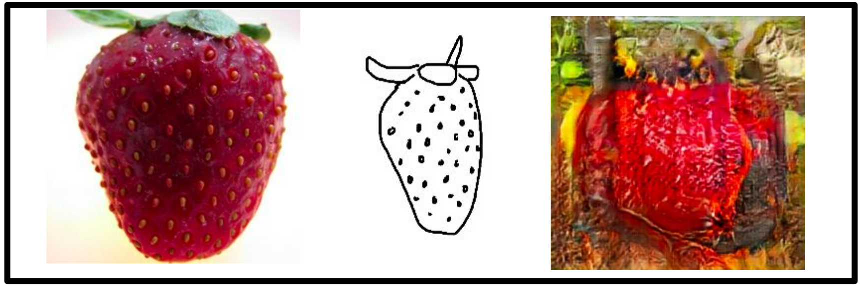
- Include a N+1 model as alternative baseline
- Evaluate performance on models trained for 200k steps instead of 50k steps
- Learn penalties values (a, b, c) using cross validation
- Test if people think our images are real using using Mechanical Turk

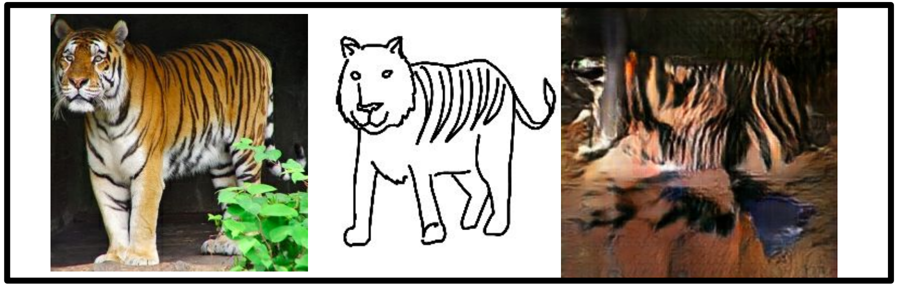


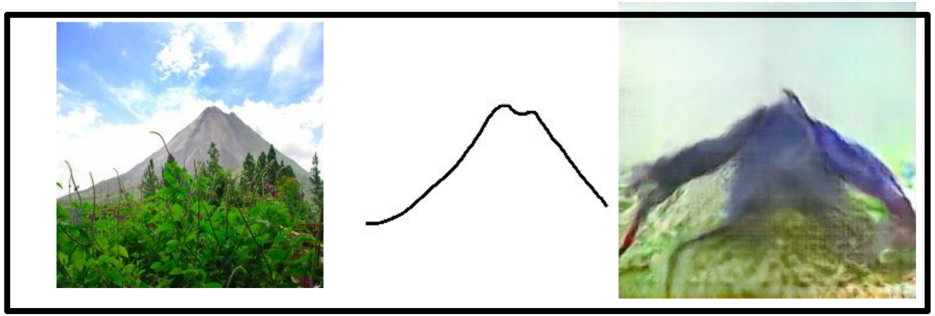
1. Sketches
2. Photos
3. Cropped Photos
4. X Orig
5. Orig2
6. 2N2N
7. X 2N
8. X Pen
9. Pen2
10. CondN
11. Image Seg

More Images: Awesome Generated Images

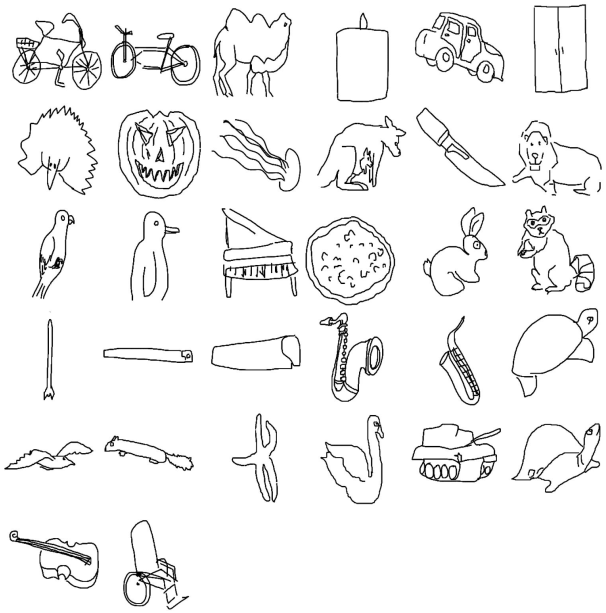




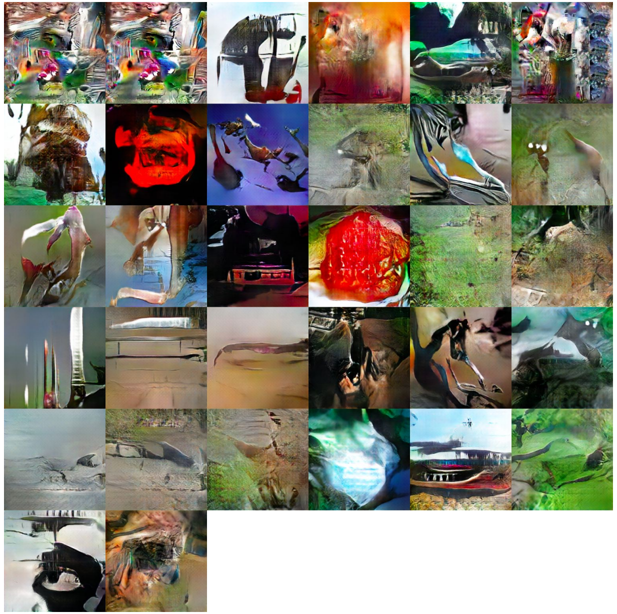




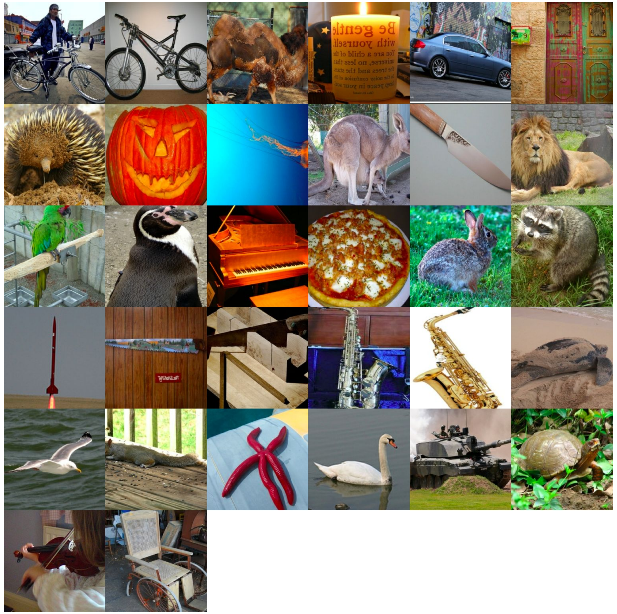
More Images: 2N Class Discriminator



Input

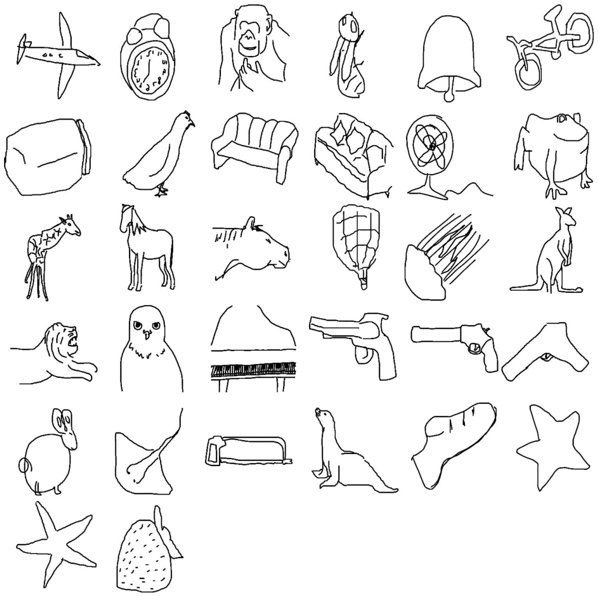


Output



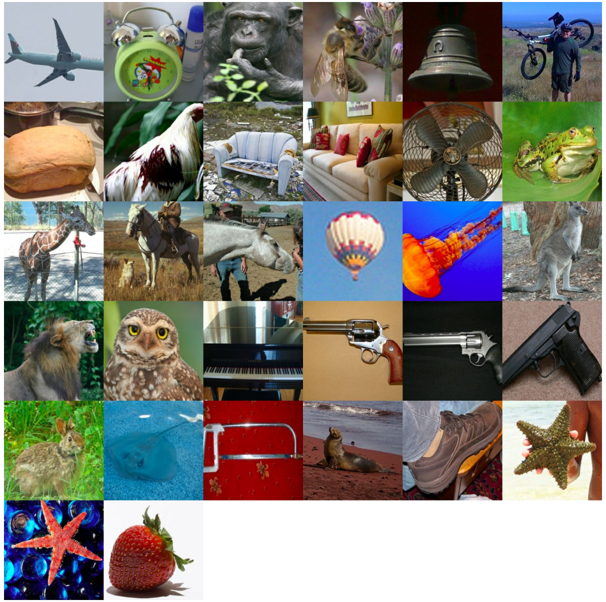
Targets

More Images: Class Conditional Generator



Input

Output



Targets