Vision and Perception 2020
GAN for Dogs Image Generation

Irene Bondanza, Andrea Caciolai

Master Degree in
Artificial Intelligence and Robotics
Sapienza, University of Rome

A.Y. 2019 - 2020

Introduction

- **Task**. For our project, we decided to implement a GAN to perform ***image generation***.
- **Dataset**. We opted for the ***Stanford Dogs*** dataset, containing 20.580 pictures of 120 different breeds of dogs from around the world.
- **Model**. We implemented two different models:
  - a ***DCGAN***;
  - a state-of-the-art GAN model, called ***BigGAN***.

  Unfortunately, neither one yielded satisfactory results.

Background

GANs are neural network architectures that pair two models: a generator, whose aim is to generate fake samples considered as real, and a discriminator, which distinguish real images from generated ones.
These two models have different and opposite task to accomplish:

- Training the generator to maximise discriminator error so that the fake images are classified as real
- Training the discriminator to minimize the error so that fake images and real ones are correctly distinguished.

Dataset description

Stanford Dogs dataset [6] contains images of 120 breeds of dogs for a total amount of 20580 images.
This dataset contains images with different shapes, angles and in some pictures, dogs are not the principal subject (e.g. presence of dog's owner) leading to get more difficult for the GAN to learn from it.

Figure: Images from Stanford Dogs dataset

Dataset preparation

The dataset is built using images and annotations from ImageNet [3] , so it contains bounding box information on how a image can be cut out to focus only on the dog.

Preprocessing

1. **Normalization in [-1, 1]**: related to the precence of Tanh activation function in generator's output layer
2. **Random flip**: randomly flip images

Models

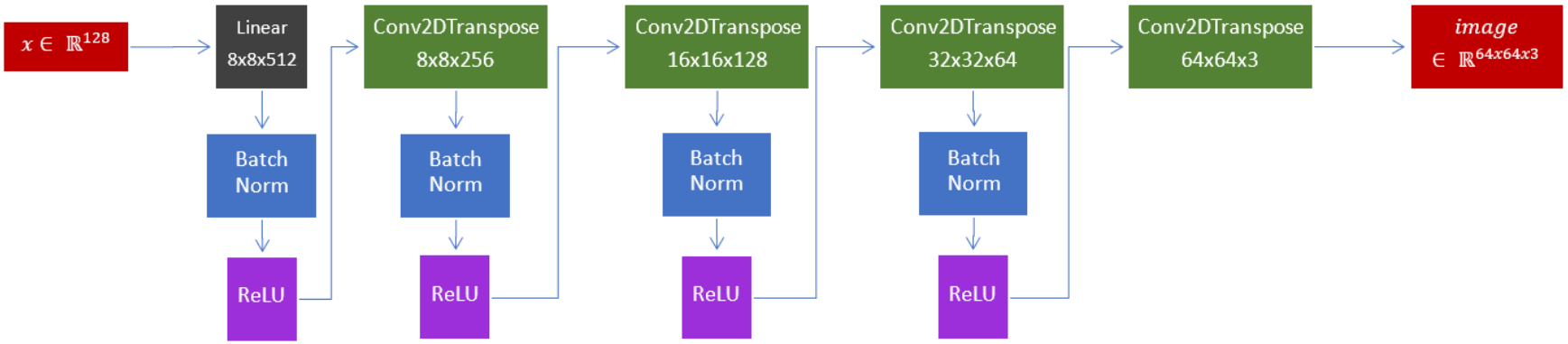
We implemented the first model, a **DCGAN** [8] but considering that we didn't achieve significant result we decided to implement a more complex model known as **BigGAN** [1].

DCGAN

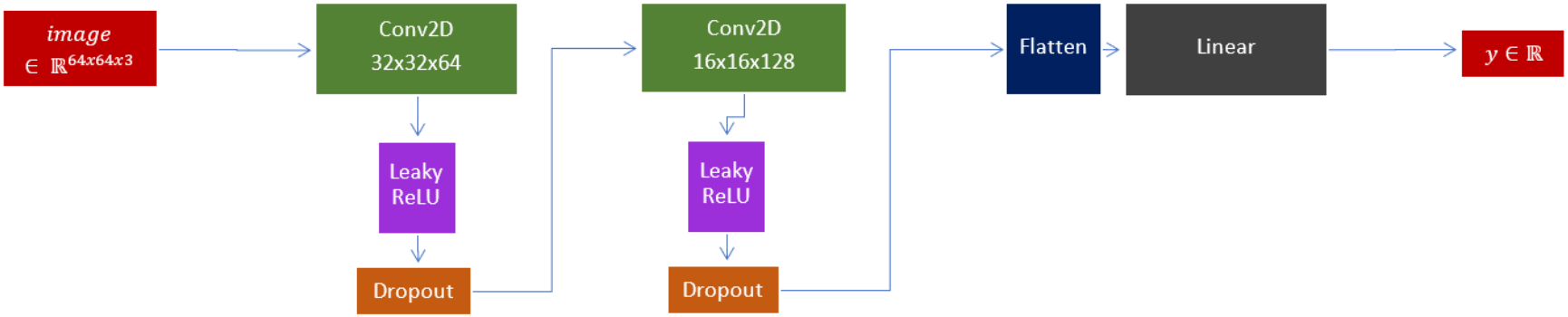
- Deep Convolutional GAN is a GAN architecture proposed in 2015 which uses basic principles of CNNs in order to build good image representations.
- According to [8] these are the guidelines for DCGANs:
  - Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
  - Remove fully connected hidden layers for deeper architectures.
  - Use batchnorm in both the generator and the discriminator.
  - Use ReLU activation in generator for all layers except for the output, which uses Tanh.
  - Use LeakyReLU activation in the discriminator for all layers.

DCGAN (Architecture)

Generator



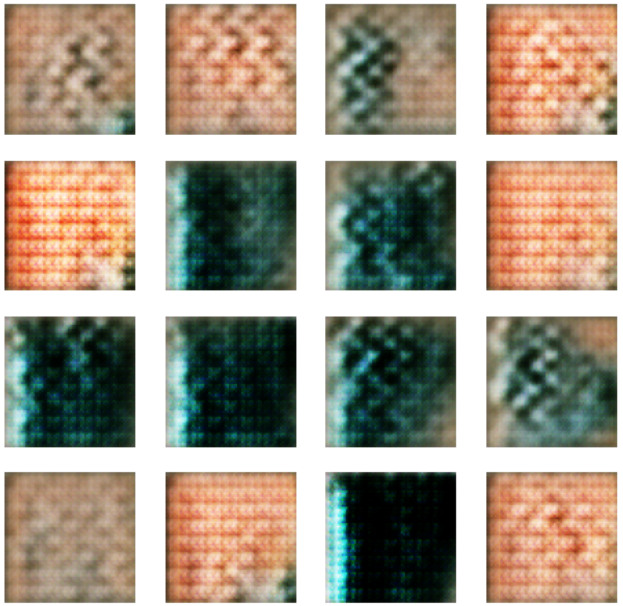
Discriminator



DCGAN (Implementation choices)

- Batch normalization applied only in generator.
- In discriminator we applied a dropout layer for each Convolutional layer.
- Normalization techniques may speed up a DCGAN's convergence:
  - Weight normalization.
  - Spectral normalization.

  Neither one nor the other have brought to any consistent improvement.

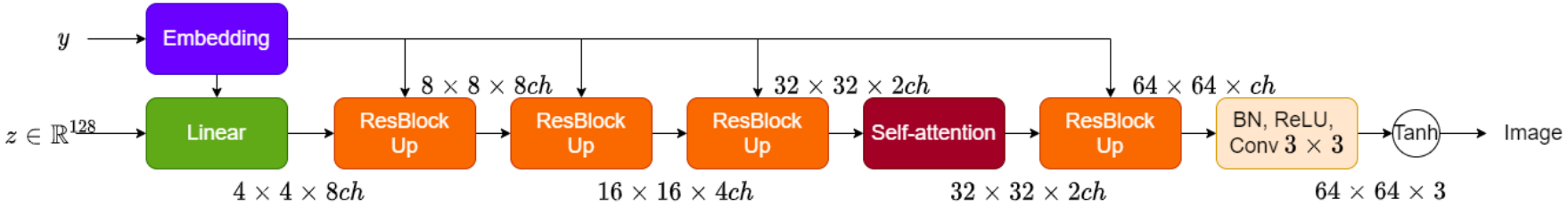


BigGAN

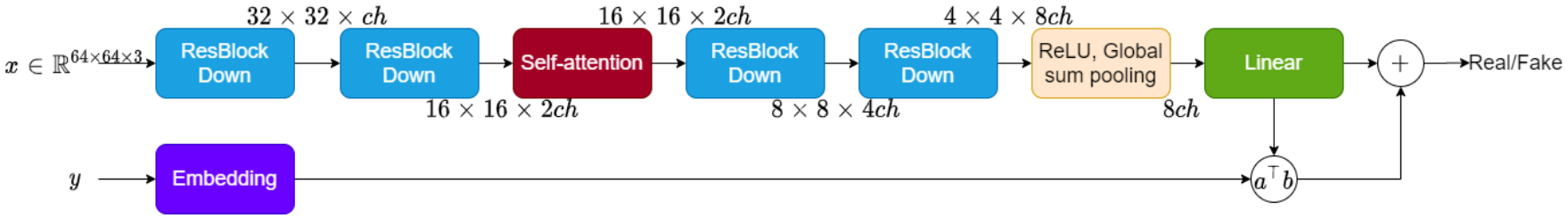
- We studied and implemented this much more complicated architecture to try to achieve better results.
- BigGAN is a GAN architecture proposed in 2019 in [1] in which the authors show its ability to generate high resolution images (up to 512×512) of impressive quality.
- It integrates several techniques proposed by earlier works:
  - Architecture with residual blocks [5] [4].
  - Conditional batch-normalization [9].
  - Self-attention [10].
  - Orthogonal regularization [2] and Spectral normalization [7].

BigGAN (Architecture)

Generator



Discriminator

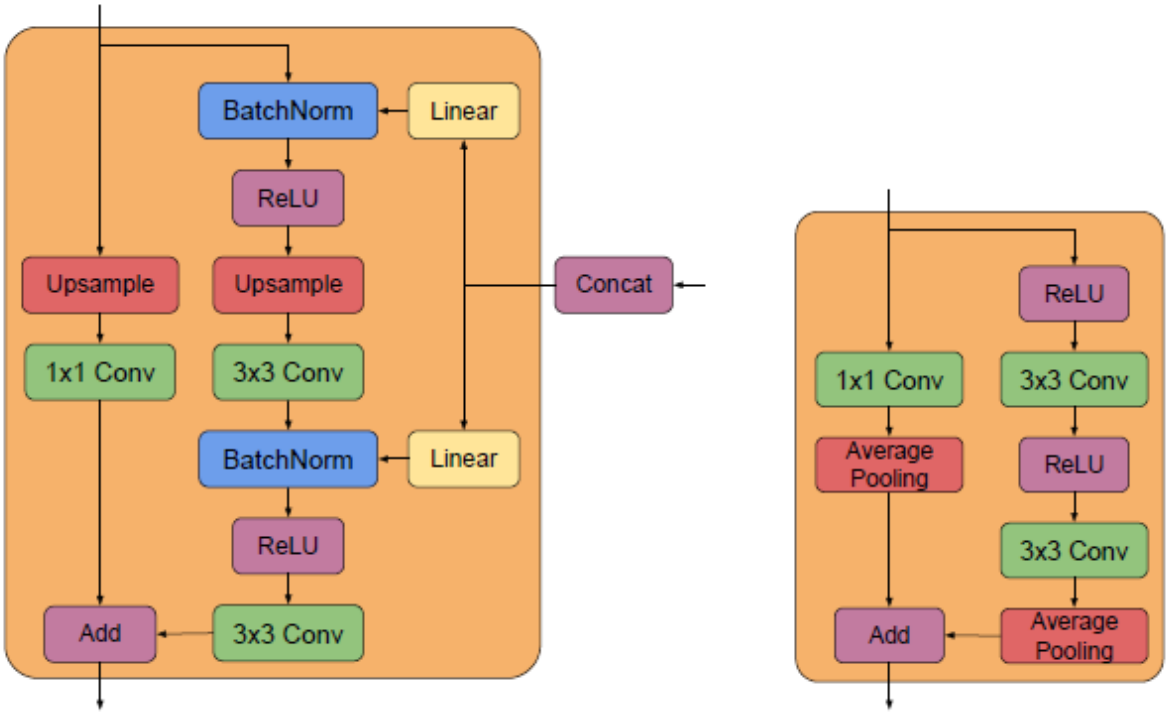


BigGAN (Residual Blocks)

- Residual blocks perform a (up, down)sampling with ***shortcut connections***.

$$y = \mathcal{F}(x) + x \tag{1}$$

  If the mapping the block is trying to learn is "close to the identity", shortcut connections lead to easier optimization.
- LeakyReLU and strided (de)convolution instead of pooling and upsampling.



BigGAN (Conditional BN)

- Conditional batch-normalization is used in the residual blocks to inject class information (dogs breed) into the generator (cGAN). It should be easier to generate a specific breed when requested than a generic dog.
- Conditional batchnorm performs the following transformation

$$x \mapsto \gamma \odot BN(x) + \beta \tag{2}$$

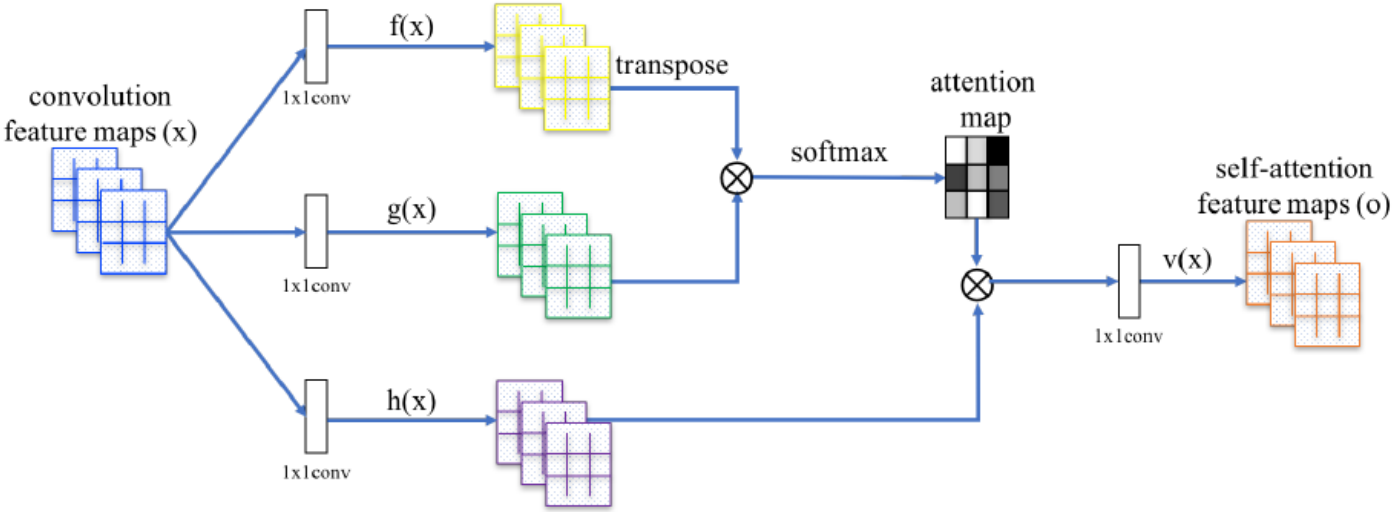
where

  - $BN(x)$ is regular batchnorm
  - γ, β are scaling and shifting coefficients computed by a linear layer

$$(\gamma, \beta)^\top \leftarrow \mathrm{Linear}(y) \tag{3}$$

BigGAN (Self-attention)

- Self-attention is a powerful mechanism to capture long-range dependencies.
- Since convolution processes information in a local neighborhood, this helps using cues from all feature maps locations.
- Linearly combines, at each location i, every location j according to its ***importance*** (attention from i to j), computed with an ***attention mechanism*** that produces an ***attention map***.



BigGAN (Self-attention) (2)

- Each location is encoded as a ***query***, ***key***, and ***value***.

$$x_i \mapsto f(x_i), \quad x \mapsto g(x_i), \quad x \mapsto h(x_i) \tag{4}$$

- Attention map tells us importance of key j for query i.

$$\beta_{ij} = \mathrm{softmax}(s_{ij}), \quad s_{ij} = \underbrace{f(x_i)^\top}_{\text{query}} \underbrace{g(x_j)}_{\text{key}} \tag{5}$$

- Attention map is used as linear coefficients for combining values j to compute the output value v_i.

$$v_i = \sum_j \overbrace{\beta_{ij}}^{\text{attention from } i \text{ to } j} \underbrace{h(x_j)}_{\text{value of } j} \tag{6}$$

- Final conv layer outputs new feature map

$$o = conv_{1\times1}(v). \tag{7}$$

Regularization

- Very powerful model, requires careful regularization: ***orthogonal regularization***, ***spectral normalization***.
- Orthogonal regularization, along with orthogonal ***initialization*** enforces ***orthogonality***.
- Orthogonality is a desirable quality in ConvNet filters: multiplication by an orthogonal matrix leaves norm unchanged, remedy to vanishing or exploding signals and gradients in deep networks.
- However, it is shown to be too limiting, so BigGAN uses a relaxed variant

$$R(W) = \beta \left\| W^\top W \odot (1 - I) \right\|_F^2. \tag{8}$$

Regularization (2)

- ***Spectral normalization*** is applied to every layer, not just (de)conv.
- Standard GANs formulation leads to unbounded gradients for the discriminator, need regularizing effect to enforce ***Lipschitz condition***.
- Spectral normalization enforces Lipschitz condition by constraining the spectral norm of each layer

$$W \mapsto W_{SN} = \frac{W}{\sigma(W)}, \quad \sigma(W) = \underbrace{max_{h \neq 0} \frac{\|Ah\|}{\|h\|} = \lambda_{max}(W)}_{\text{spectral norm}}. \tag{9}$$

- Eigendecomposition is expensive: approximated with ***power iteration method***.
  1. Estimate first left and right singular vectors u, v with iterative update rule.

$$\tilde{v} \leftarrow \frac{W^\top \tilde{u}}{\left\| W^\top \tilde{u} \right\|_2}, \quad \tilde{u} \leftarrow \frac{W\tilde{v}}{\|W\tilde{v}\|_2} \tag{10}$$

  2. Compute approximate spectral norm.

$$\sigma(W) \approx \tilde{u}^\top W \tilde{v} \tag{11}$$

Experimental setting

- The training was performed on Colab, exploiting its Tesla P100 GPUs.
- Training was done with the standard ***non saturating*** objective.

$$D^* = \arg\max_D \overbrace{\mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right]}^{-BCE(1,D(real))} + \overbrace{\mathbb{E}_{z\sim p_z(z)}\left[\log(1 - D(G(z)))\right]}^{-BCE(0,D(fake))} \tag{12}$$

$$G^* = \arg\max_G \underbrace{\mathbb{E}_{z\sim p_z(z)}\left[\log D(G(z))\right]}_{-BCE(1,D(fake))} \tag{13}$$

- We also tried WGAN-GP and RaLSGAN losses, without observing any benefit.

Hyperparameters

Hyperparam	DCGAN	BigGAN
Epochs	1000	500
Batch size	128	
Optimizer	Adam $(0.5, 0.999)$	Adam $(0, 0.999)$
Disc. steps	1	2
Lr	$(10^{-4}, 10^{-4})$	$(2 \cdot 10^{-4}, 5 \cdot 10^{-5})$
Ch	–	64
Label smoothing	0.3	
Label noise	0.05	

Evaluation: Metrics

- To evaluate our models, we used
  - ***visual inspection***, qualitative metric.
  - ***FID*** (Frechet Inception Distance), quantitative metric.
- FID: metric proposed as an improvement over the existing IS metric (Inception Score).
- IS estimates the quality of a collection of synthetic images based on how well ***Inception v3*** classifies them as one of 1,000 known objects.
- Problem: statistics of real world samples are not used.
- FID score: evaluate synthetic images based on the statistics of a collection of synthetic images compared to the statistics of a collection of real images from the target domain.

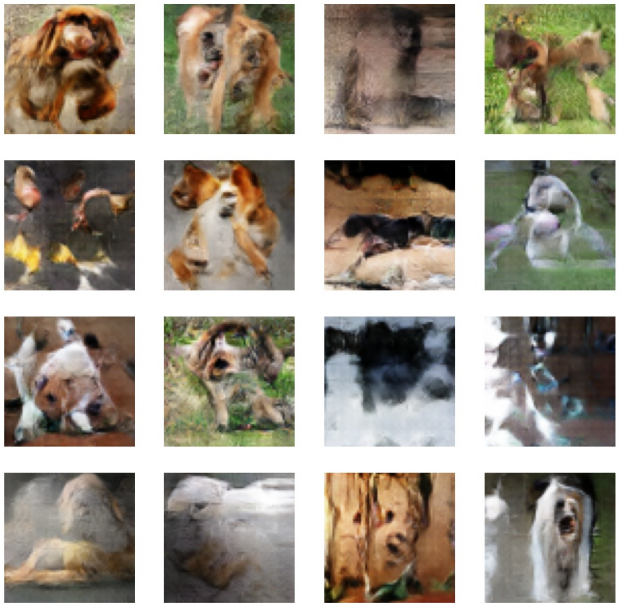
$$d^2 = \|\mu_{real} - \mu_{fake}\|^2 + \mathrm{Tr}\left(\Sigma_{real} + \Sigma_{fake} - 2\sqrt{\Sigma_{real}\Sigma_{fake}}\right) \tag{14}$$

- Statistics extracted from activations (output layer removed) computed by Inception v3.

Evaluation: Results

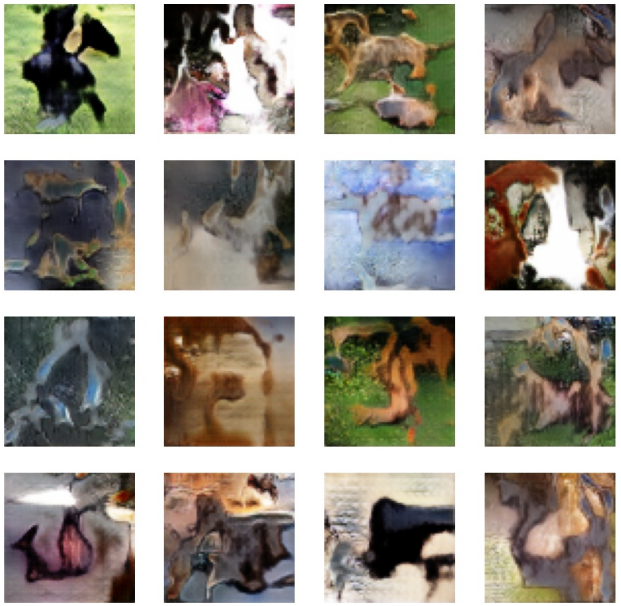
DCGAN

FID score: 341



BigGAN

FID score: 353



Results: Considerations

- Results were underwhelming, despite our best efforts.
- We believe this is a very difficult learning problem, due to the dataset:
  - Images are very ***diverse***: 120 different dog breeds, both puppy and adult dogs, very broad example pool of what a "dog" is.
  - Images taken in ***different conditions***: the pictures are taken from various different angles, with different illumination, dogs represented frontally, sideways, outdoor, indoor, etc.
  - ***Partial*** images: the bounding boxes provided to retain only the dogs in the picture usually capture only portions of the dogs, and also portions of other objects enter the image.
- Would be interesting to see how our models perform on "easier" datasets.

References (1)

- Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large scale gan training for high fidelity natural image synthesis". In: ***arXiv preprint arXiv:1809.11096*** (2018).
- Andrew Brock et al. "Neural Photo Editing with Introspective Adversarial Networks". In: (2016). arXiv: 1609.07093 [cs.LG].
- J. Deng et al. "ImageNet: A large-scale hierarchical image database". In: ***2009 IEEE Conference on Computer Vision and Pattern Recognition***. 2009, pp. 248–255.
- Ishaan Gulrajani et al. "Improved Training of Wasserstein GANs". In: (2017). arXiv: 1704.00028 [cs.LG].
- Kaiming He et al. "Deep Residual Learning for Image Recognition". In: (2015). arXiv: 1512.03385 [cs.CV].

References (2)

- Aditya Khosla et al. "Novel Dataset for Fine-Grained Image Categorization". In: ***First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition***. Colorado Springs, CO, 2011.
- Takeru Miyato et al. "Spectral Normalization for Generative Adversarial Networks". In: (2018). arXiv: 1802.05957 [cs.LG].
- Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: ***arXiv preprint arXiv:1511.06434*** (2015).
- Harm de Vries et al. "Modulating early visual processing by language". In: (2017). arXiv: 1707.00683 [cs.CV].
- Han Zhang et al. "Self-Attention Generative Adversarial Networks". In: (2018). arXiv: 1805.08318 [stat.ML].

Thank you for your attention