Synthetic Image Generation from Textual Description

B.Tech Computer Science Major Project[1,✉]

[1]NIT Hamirpur
[1]Aditi Singh, Alekh Gupta, Kashish Srivastva, Arshad Ali, Aman Matta

Abstract

Automatically generating realistic visuals from text would be fascinating and valuable, but existing AI systems are still a long way from achieving this aim. In recent years, however, general and powerful recurrent neural network designs for learning discriminative text feature representations have been constructed. Meanwhile, deep convolutional generative adversarial networks (GANs) have started to produce highly engaging images of certain categories like faces, record covers, and room interiors. In this paper, we use a novel deep architecture and GAN formulation to bridge the gap between text and picture modelling, efficiently transferring visual notions from characters to pixels. Our model is shown to be capable of generating credible images of birds from comprehensive text descriptions.

1. Introduction

We want to translate text in the form of single-sentence human-written descriptions straight into image pixels in this project. "This small bird has a long, sharp orange beak and a yellow belly," for example, or "bird with a black beak, white feathers, and a white underbelly." The topic of producing images from visual descriptions has piqued researchers' interest, yet it remains unsolved.

Fig. 1. Examples of generated images from text descriptions.

2. Related Work

Here, we develop a deep architecture and GAN formulation combining GAN-CLS by Reed et al.[4] and MS-GAN regulation term by Mao et al.[2] to translate text, in the form of single-sentence human-written descriptions directly into image pixels. We learn a mapping directly from words and characters to image pixels. We will train our model to generate plausible images of birds from detailed text descriptions.

3. Background

In this section we briefly describe several prerequisites that our work is built upon.

3.1 Generative adversarial networks

In a two-player minimax game, generative adversarial networks (GANs) are made up of a generator G and a discriminator D. The discriminator tries to tell the difference between real training data and fake images, while the generator tries to deceive the discriminator. D and G, for example, play the following game on V(D,G):

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Goodfellow et al. (2014) shows that this minimax game has a global optimium when pg = pdata, and that pg converges to pdata under mild conditions (e.g. G and D have ample capacity). In practise, D's initial training samples are exceedingly bad, and D rejects them with a high degree of confidence. It has been discovered that having the generator maximise l works better in practise.

3.2 Word2Vec

One of the most common ways to express document vocabulary is by word embedding. They're a type of vector representation of a word. Word2Vec is a widely used shallow neural network technique for learning word embeddings. We used Google's pre-trained word2vec model in our model.

4. Model Architecture

In this section we explain the model architecture and loss functions we used for our model training.

4.1 Generative Adversial Network - Conditional Latent Space (GAN-CLS)

Viewing (text, picture) pairs as joint observations and training the discriminator to judge pairs as authentic or fake is the most straightforward way to train a conditional GAN.

The discriminator has no clear idea if real training images match the text embedding context, resulting in the *Genera-*

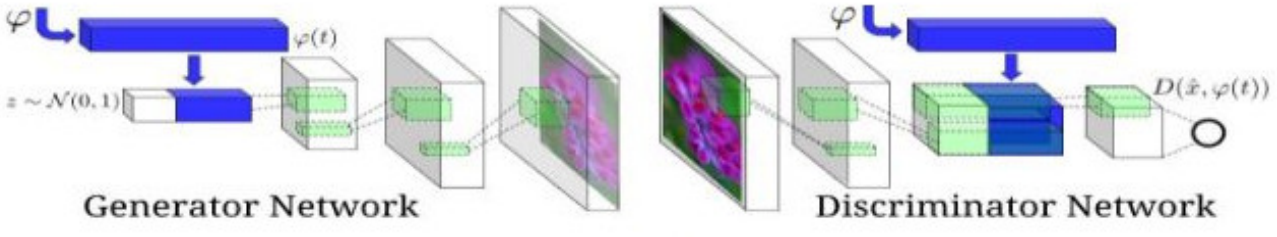

Fig. 2. Convolutional GAN architecture that is text-conditional. Both the generator and the discriminator employ text encoding (t). Reed et al. are cited as a source.

tive Adversarial Networks Conditional Latent Space.

4.2 GAN-CLS Generator

The Generator in this model has the following architecture. The discriminator ignores the conditioning information in the beginning of training and easily rejects samples from G since they do not appear reasonable. G must learn to align plausible images with the conditioning information once it has learned to generate them, and D must learn to evaluate whether samples from G fit this conditioning requirement. Instead of generating random images, this formulation will allow Generator to generate images that are aligned with the supplied description.

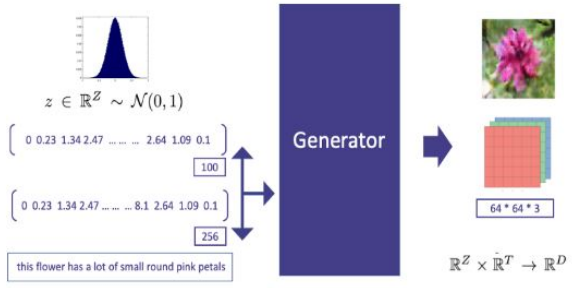

Fig. 3. The textual description was converted to a 256-dimensional embedding and then combined with a 100-dimensional noise vector sampled from a Normal distribution. Instead of generating random images, this formulation will allow Generator to generate images that are aligned with the supplied description.

4.3 GAN-CLS Discriminator

The discriminator in a naive GAN looks for two types of inputs: actual photos with matching text and synthetic images with random text. As a result, it must implicitly distinguish between two types of errors: unrealistic visuals (for any text) and realistic images of the incorrect class that contradict the conditioning information. We changed the GAN training method to distinguish these mistake sources and added a third type of input: actual images with mismatched text that the discriminator must learn to assess as fake. The discriminator can provide an extra signal to the generator by learning to maximise image/text matching in addition to image realism.

During the training period, a sequence of diverse (picture, text) pairings are given as input for the Discriminator's two-fold responsibility:

- The input variables are (Real Image, Real Caption) and the target variable is set to 1
- As input, there is a pair of (Wrong Image, Real Caption) and the target variable is set to 0.
- The input variables are (Fake Image, Real Caption) and the goal variable is set to 0.

When the Discriminator's target variable for the (Fake Image, Real Caption) pair is 0. Generator loss is set to 1 because Generator wants Discriminator to identify it as a true image.

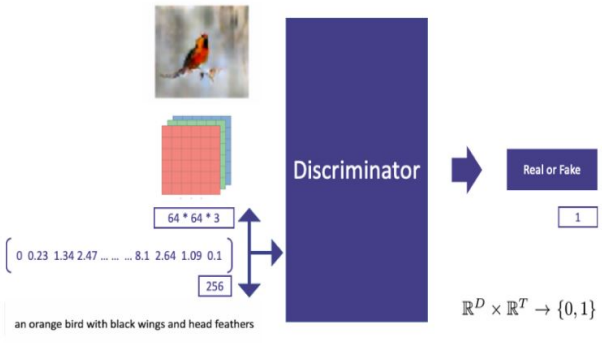

Fig. 4. It also predicts if an image is real or a fake, as well as whether the image and text are aligned. This formula forces Generator to make images that are not only realistic in appearance, but also match the supplied description.

4.4 Algorithm

We use the Caltech-UCSD Birds (CUB-200) [5] dataset to experiment with the combination of GAN-CLS method by Reed et al. [4] and regulatory term of the MS-GAN by Mao et al. [2].

The GAN architecture provided by Ledig et al [3] was also tested.

Supplementary Note 1: 5. Implementation

In this section we present the project implementation and workflow.

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

$$+ \quad \max_G \left(\frac{d_i(G(c, z_1), G(c, z_2))}{d_z(z_1, z_2)} \right)$$

Fig. 5. GAN-CLS algorithm by Reed et al. [4] and the MS-GAN regulation term by Mao et al. [2].

5.1 Proposed Workflow

We have discussed the workflow steps in Fig. 6.

5.2 Dataset

Caltech-UCSD Birds (CUB-200) [5] is a photo dataset with linguistic descriptions of 200 bird species (mainly from North America). Each bird species has a total of 100 images.

5.3 Learning word embeddings

The text was initially preprocessed to remove any noise, punctuations, white spaces and spelling errors (Normalization). The stopwords (a, an, the) were removed and the captions were converted to lowercase. The words were stemmed.
All the captions were then fed to Google's pre-trained word2vec model. Then the output was saved as a vector file for further use.

5.4 Generator

The generator model is a 39-layer model with four layers for reducing text dimensionality and six convolution layers for picture reduction.

We start by sampling from the noise prior. We use word2vec to encode the text query. The word(caption) embedding is first compressed to a tiny dimension using a fully-connected layer, then leaky-ReLU, and finally concatenated to the noise vector z. Following that, a standard deconvolutional network is used for inference. We send it into the generator G, which generates a synthetic image.

The generator loss is the difference between the generated and real image's binary cross entropy.

5.4 Discriminator

The discriminator model has 25 layers, with many layers of stride2 convolution followed by spatial batch normalisation and leaky ReLU.

After rectification, we reduce the dimensionality of the word embedding in a (separate) fully-connected layer. To compute the final score from D, we do a 1 1 convolution followed by rectification and a 4 4 convolution. On all convolutional layers, batch normalisation is applied.

The discriminator loss is the total of the binary cross entropies between the wrong image and the real caption, and the fake image and the real caption.

6. Results

With a learning rate of 0.000035 and beta 1=0.5, we trained the GAN model for the discriminator and generator for 150 epochs. The majority of the synthesised images look to reflect genuine bird colours and shapes, and there appears to be a lot of diversity and variety; nonetheless, the GAN had some minor mode collapse difficulties when synthesising images based on made-up descriptions. Figures 7, 8, and 9 show the results of the model training.

7. Conclusions

We created a simple and successful methodology for generating graphics from extensive visual descriptions in this paper.

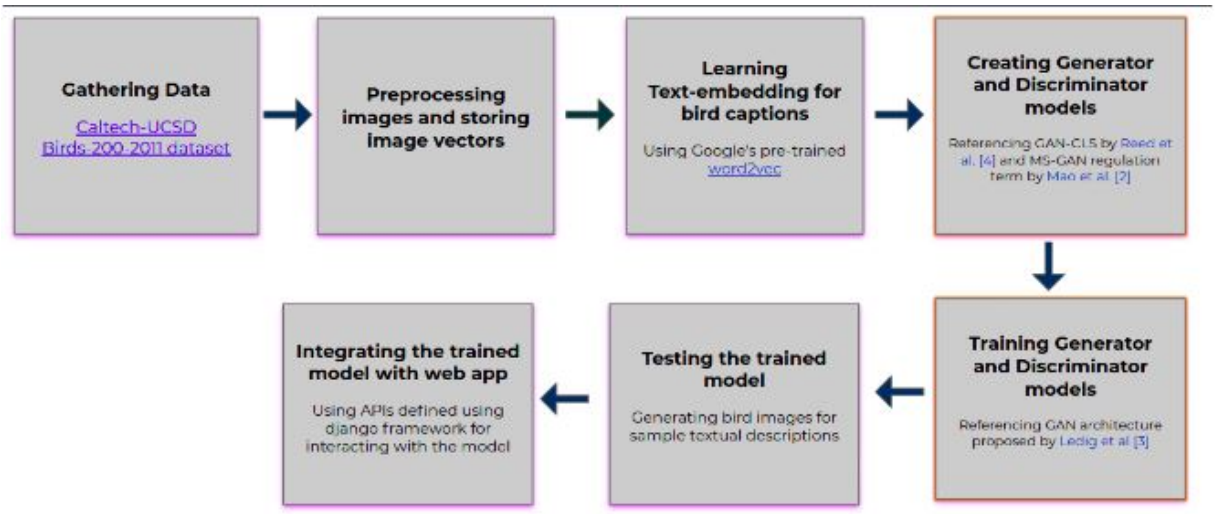

Fig. 6. Workflow steps for implementation.

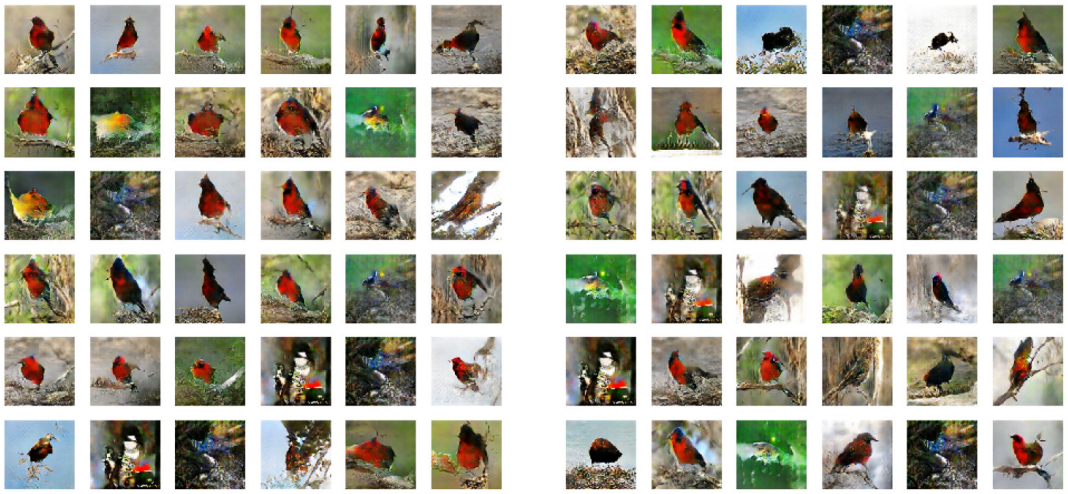

Fig. 7. Images generated for caption "This bird is completely red and black."

Fig. 8. Interpolation between sentence vectors

We showed that the model can generate a large number of plausible visual interpretations of a single text caption. The text to image synthesis on CUB was significantly enhanced by our manifold interpolation regularizer. We demonstrated the separation of style and information, as well as the transfer of bird stance and background from query photos to text descriptions.
We plan to scale up the model to higher resolution photos in the future and incorporate more forms of text.

Acknowledgements

This work has been created as part of B.Tech. four-year major project in Computer Science Engineering at NIT-Hamirpur, under the direction of Dr. Pradeep Singh, who provided regular assistance and support.

References

1. I. J. Goodfellow et al., “Generative Adversarial Networks”, arXiv [stat.ML]. 2014. Available: https://arxiv.org/abs/1406.2661v1

2. Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, en M.-H. Yang, “Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis”, arXiv [cs.CV]. 2019. Available: https://arxiv.org/abs/1903.05628

3. C. Ledig et al., “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”, arXiv [cs.CV]. 2017. Available: https://arxiv.org/abs/1609.04802

4. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, en H. Lee, “Generative Adversarial Text to Image Synthesis”, arXiv [cs.NE]. 2016. Available: https://arxiv.org/abs/1605.05396

5. C. Wah, S. Branson, P. Welinder, P. Perona, en S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset”, California Institute of Technology, 2011. Available: http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

6. . Karras, S. Laine, en T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks”, arXiv [cs.NE]. 2019. Available: https://arxiv.org/abs/1812.04948

7. A. Radford, L. Metz, en S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, arXiv [cs.LG]. 2016. Available: https://arxiv.org/abs/1511.06434