Movie script to animation with a GAN Algoorithm(text to images synthesis) and inbetweeing between the images using motion estimation method along with interpolation

Amit H Patel
The University of Adelaide
a1697700@student.adelaide.edu.au

Abstract

Generating animation from natural language is one of the latest topic of recent conditional generative models. Besides testing the ability to model conditional, highly dimensional distributions, text to animation has many exciting and practical applications such as animation creation or computer-aided content creation. Using Generative Adversarial Networks(GANs), more applications are in the progress with higher success rate. This paper explains about the current applications for generating animation from text such as VideoGAN algorithms and then it will propose a slight different approach to make animation from natural language that can provide good results. Although, generating animation from text using the proposed method might be more slower then available VideoGAN algorithms,it can provide vast opportunities to get results as expected. The proposed method uses a GAN algorithm(DCgan) to generate images from text by using MS-COCO generic data-set. The generated images are transformed in an animation using motion estimation methods using interpolation that provides inter between image and generate a final animation. The paper shows results achieved through the experiment of the proposed algorithm and already implemented videoGan algorithm with briefing them.

1. Introduction

In computer vision, deep learning has advanced itself into full fledged algorithms. One of the most challenging problems in Natural Language Processing to generate videos: given an videos for training and a text description of the video must be produced. Text to video is the reverse problem: given a text description, an frames in the video which matches that description must be generated. It is having some similarity with a language translation problems from higher view. The overall problem of generating videos from a natural language description can be divided in two separate problems: videos to description and caption to videos. This conversions are considered as multimodal problems too. Due to after learning videos to captions, there can be many different scenario or results for a single caption. It is like one caption "A guy moving on playground" can be imagined by mind with many different views and scenarios. Same way, This text to video includes the multimodal problems in it. Although providing proper caption in proper sequence prevents this problem easily. This applications are being used in multimedia commercials and cartooning too. However there is very wide scope in this field to go ahead and make more generic algorithm with less training using different algorithm. Despite of advancement of converting text into video/animation, this paper proposes conversion of natural language to images and the images to animation with using external algorithms. The proposed method runs on a GAN algorithm to generate image synthesis from text specifically Deep Convolutional GAN[4]. Here, it is optional to choose DCGan, any GAN algorithm that performs well for text to images synthesis can be used. DC-GAN uses a deep convolutional architecture for the generator as well discriminator. The details on generator and discriminator is explained in further sections. there are similar algorithms and new approaches to do same task like: GAN-CLS[6] and Stacked-Gan[7].After generation of images from text, using object detection, desired background is replaced behind objects. After background change, with the help of motion estimation, in between images are found. To find in between images, the paper approches motion estimation on the bases of Block matching and interpolation to create animation/video from all images. Separately, it can be converted to Gif or different framwork of video using external tools. Considering present improvements for text to video, the proposed algorithm has some constraints. The biggest constraint is that running DCGan seperataly and then running images to video algorithm. The DCGan

is implemented algorithm that runs on one platform. While, after getting all necessary results, images are converted into video using another platform due to more convenience and seperating task from each other. However, regardless of this constraint, it performs well. Before, explaining whole method in detail, the proposed method is compared with present available methods and necessary background information is provided in following two sections.

2. Related Work

Creating a scene from natural language, there are some of algorithms are already proposed such as VideoGAN[1], pixel generation[5] and Composition, Retrieval and Fusion Network (Craft)[8]. For pixel generation, the Generative Adversarial Networks (GANs)[2] is used to synthesize images. One of important thing is semantic and temporal coherence in designing GANs to generate videos. Pixel generation uses TGANs-C, in which the input to the generator network is a concatenation of a latent noise vector and caption embedding, and then is transformed into a frame sequence with 3D spatio-temporal convolutions. Here, the discriminator network consists of three discriminators: video discriminator classifying realistic videos from generated ones and optimizes video-caption matching, frame discriminator discriminating between real and fake frames and aligning frames with the conditioning caption, and motion discrimi- nator emphasizing the philosophy that the adjacent frames in the generated videos should be smoothly connected as in real ones. Craft can learn video-caption data and apply to generate video using general language/caption. Using characters and objects, spatio-temporal entity segments are learned from a video database. Craft is evaluated on semantic to caption, composition consistency, and visual quality. Craft is better in generating videos while videos or captions are nto matching with any trained models. For Craft, more trained videos provide more better result and undoubtedly it performs better then pixel generation method. Aprt from pixel generation and Craft algorithm, the proposed algorithm is almost similar to videoGAN where using a GAN algorithm image synthesis are found from the text, then in GAN structure only videos are generated including foreground and background layers. Those foreground and background layers automatically detects the objects and with a required background and using char-CNN-RNN encoder it forms a frames in a squence that generates a video. Indeed, the proposed algorithm uses the almost same steps but not in the process of GAN to make a video. Onve text to image synthesis are found, it separates the total method and adopts other algorithms to find inbetween images and form a video with the help of interpolation. The difference is more significant in terms of accuracy of making good quality animation but still it include more manual steps like object detection and background change.

3. Background

3.1. Generative Models

Text to image synthesis is one of the problem generative models can solve it. Currently, Generative Adversarial Networks (GANs) provide the best results for text to image, a particular type of generative model. Before introducing GANs, generative models are briefly explained. Some notations are listed for understanding purpose, Consider a dataset X = $\{x^{(1)}, ..., x^{(m)}\}$ composed of m samples where, x(i) is a vector. In the particular case of this report, x(i) is an image encoded as a vector of pixel values. The dataset is produced by sampling the images from an unknown data generating distribution Pr, where r stands for real. A generative model is a model which learns to generate samples from a distribution Pg which estimates Pr. The model distribution, Pg, is a hypothesis about the true data distribution Pr. Most generative models explicitly learn a distribution Pg by maximising the expected log-likelihood $E_{x\ P_r} log(Pg(x|\theta))$ with respect toθ, the parameters of the model.maximum likelihood learning can be achieved by putting more probability mass around x where there are more samples of X and less around less samples of X. Assuming Pr and Pg are densities. One of the valuable properties of this approach is that no knowledge of the unknown Pr is needed because the expectation can be approximated with enough samples according to the weak law of large numbers. The way they work and how they compare to other models is explained in the next section.

3.2. Generative Adversarial Networks

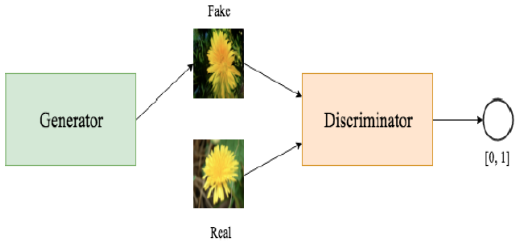


Figure 1.1: High-level view of the GAN framework. The above diagram shows process of producing synthetic images.The GAN framework based on the discriminator and the generation. Image is inputted in discriminator and that provides image with the probability of being real. While, A generator tries to forge painting, the discriminator tries to find imitations. The motive of the described framework is that the generator can produce a realistic images. To understand working of GAN, suppose X is dataset and $x(i)$ is in X such as the space of images $[-1, 1]^n$. The discriminator learns using function $D_w : X- > [0, 1]$ which takes an image x as input and outputs probability of image being real. Let z be the range a random vector Z and fixed distribution such as Pz =N(0,1). Generator learns with a function

$G_\theta : z- > X$ whuch maps the state of random vector Z corresponding to a random vector X. The states X P_g correspond to the images the generator creates.This way generator learns mapping of noise in images that proposes an quation 1.1 [9]

$$V(D,G) = E_{x\ p_r}[log(D(x))] + E_{z\ p_z}[log(1 - D(G(z)))] \quad (1)$$

The above function represents a value function V(D,G) that represents the payoff of the discriminator. The discriminator tries to maximize value(V) while generator tries to minimize it.

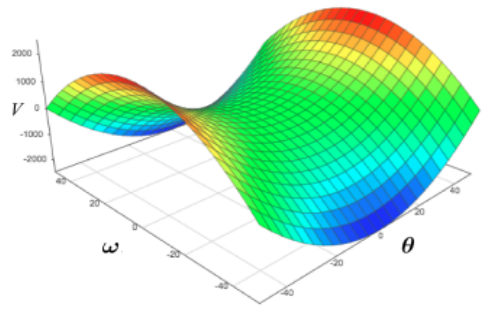


Figure: 1.2 The Nash Equilibrium corresponds to the space where V is at a minimum with respect to θ and at maximum with respect to w.

corresponds to the space where V is at a minimum with respect to θ and at a maximum with respect to w. The optimal parameters of the generator can be described as in equation:

$$\theta^* = argmin_\theta max_w V(D,G) \quad (2)$$

The above minimax optimization can be solved by Nash Equilibrium:

$$L_G = -E_{z\ p_x}[log(D_w(G_\theta(z)))]$$
$$L_D = -E_{x\ P_r}[log(D_w(x))] - E_{z\ p_z}[log(1 - D_w(G_\theta(z)))] \quad (3)$$

Here, the networks are trained alternatively, for only one step each.

3.3. Text Embedding

To use the nature text, it has to be formed in the vectors. This process is called text embedding. There are already text embedding on the database of MS-COCO[1], which is used in the proposed method.

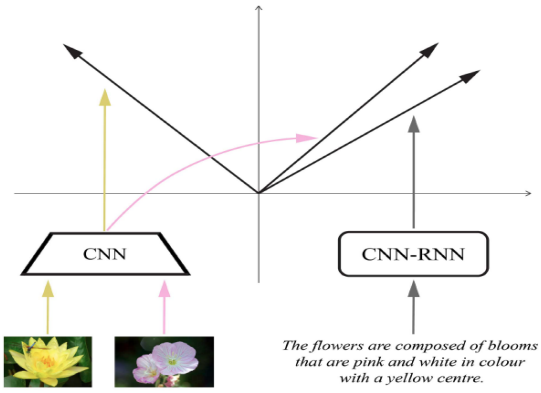


Figure 1.3: Mapping images using the char-CNN-RNN encoder. The char-CNN-RNN encoder[11] computes the text embedding. So,images and the captions are mapped/matched to a common space using encoder where they are vectorised with a inner product.

Here, Convolutional Neural Network(CNN) is used for images. Convolutional-Recurrent Nueral Network(RNN) is used for the transformation of the text descriptions.

In the proposed method, the approach of Reed et al.(2016) is used which is using deep convolutional and recurrent text encoders. Thus vactorization between text and images is achieved through the following function:

$$F_t = \frac{1}{N}\Delta(y_n, F_v(v_n) + \Delta(y_n, F_{tt}(t_n))) \quad (4)$$

Where,
The training dataset = $(v_n, t_n, y_n) : n = 1, ..., N$
Δ = 0-1 loss
v_n = images
t_n = the matched text descriptions
y_n = class labels

$$F_v(v) = argmin_{y \in m} E_{t\ T(y)}[\phi(v)^T \varphi(t)] \quad (5)$$
$$F_v(v) = argmin_{y \in m} E_{v\ V(y)}[\phi(v)^T \varphi(t)] \quad (6)$$

Where,
ϕ = image encoder
φ = the text encoder
$T(y)$ = the set of text description
$V(y)$ = images

The model is trained using equation(4) in many trained models(e.g. Akata et al.(2015), Reed et al.(2016)) . Skip-Thought Vectors[12] is one of the alternative method that maps similar syntax matching vectors.

3.4. Block matching algorithm

Block matching method finds the matched blocked on the images. Block matching can be achieved by following steps:

1 Subdivide current frame into blocks.

2 Finding displacement vector for each block.

After finding all possible positions to get the best matches in the search range, motion estimation can be predicted using Taylor Approximation.

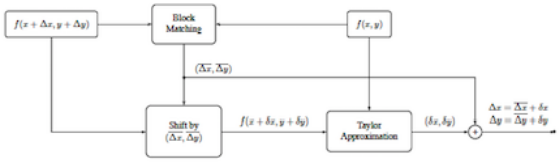


Figure: 1.4 There are some advance algorithms are proposed to avoid full searches on the images such as bilateral ME[3] and three step search[2].

3.5. Optical Flow

The below equations are core working od classical optical flow[4]. Suppose having two frames f(x,y) and g(x,y) we can approximate it with Taylor series approximation.

$$g(x,y) = F(x+\Delta x, y+\Delta y)$$
$$g(x,y) = F(x,y) + \Delta x \frac{\delta}{\delta x} F(x,y) + \Delta y \frac{\delta}{\delta y} F(x,y) \quad (7)$$

The solving below minimizing problem can find optimal shift:

$$minimize_{\Delta x, \Delta y} \Phi(\Delta x, \Delta y) \quad (8)$$

Where,

$$\Phi(\Delta x, \Delta y) =_{x,y} (g(x,y) - f(x,y)) - \Delta x \frac{\delta}{\delta x} f(x,y) - \Delta x \frac{\delta}{\delta y} f(x,y))^2 begin equation \quad (9)$$

Due to above minimization problem is linear least square problem we can take darivations of x and y as below, $\frac{\delta \Phi}{\delta \Delta x} = 0$ and $\frac{\delta \Phi}{\delta \Delta y} = 0$

Transforming problem in:

$$\begin{pmatrix} \sum_{x,y} (\frac{\delta f}{\delta x})^2 & \sum_{x,y} \frac{\delta f}{\delta x} \frac{\delta f}{\delta y} \\ \sum_{x,y} \frac{\delta f}{\delta x} \frac{\delta f}{\delta y} & \sum_{x,y} (\frac{\delta f}{\delta y})^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} _{x,y}(g-f)\frac{\delta f}{\delta x} \\ _{x,y}(g-f)\frac{\delta f}{\delta y} \end{pmatrix} \quad (10)$$

Solving above problem, it can determine the optimal solution which will be used to find Optical flow.

4. The proposed method

4.1. Text to image synthesis using DC-GAN

In past, GAN with CNN was not able to model images. After researches, a specific architecture was developed and tested on different datasets which gave more stable results with higher resolution. The architecture was developed with the three significant changes on CNN architectures. Changes on convolutional net: To allow network to learn its own spatial sampling, the spatial pooling function in CNN is changed with strided convolutions. Second change was to eliminate fully connected layers on top of convolutional feature. Third change is introducing Banch Normalization: It normalizes the input to have zero mean and unit variance. While for unexepted outputs ReLU activation is used with the help of Tanh function.

For Generator, we can see in the following diagram how layers are formed to train the images.

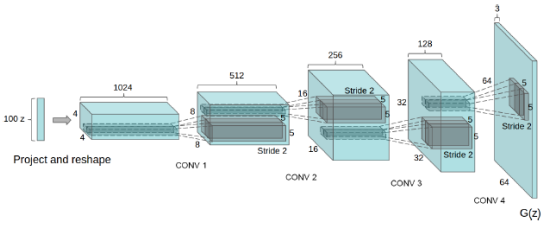


Figure 1.5: The diagram of DCGAN generator on LSUN modeling. The above diagram staisfy the DCGan architecture steps which are listed in the below section.

4.1.1 The architecture overview of Deep Convolutional GANs

1 Instead of pooling layer use strided discriminator and fractional-strided generator.

2 For discriminatore abd generator use batchnorm.

3 Eliminating all fully connected hidden layers.

4 Apart from output from generator use ReLU for all layers.

5 All layers of discriminatore includes LeakyReLU activation.

4.1.2 Network Architecture

The generator network is denoted $G : R^z R^t -> R^D$. The discriminator as D: $R^D R^T -> 0, 1$. Where,
D = Dimension of images
T = Dimension of the text embedding
Z = Dimension of noise on G input

The whole process starts with the generator G that sample from noise prior $z \in R^z$ $N(0,1)$. text query(t) is encoded by text encoder φ A fully connected layer is used to compress description embedding to small dimension by leaky- ReLU and that is concatenated to the noise vector(z). Here, it proceeds a normal deconvolutional network: Synthetic image is generated via $G(z, \varphi(t)) - > x$. Feedforrward inference in the generator G is corresponded by

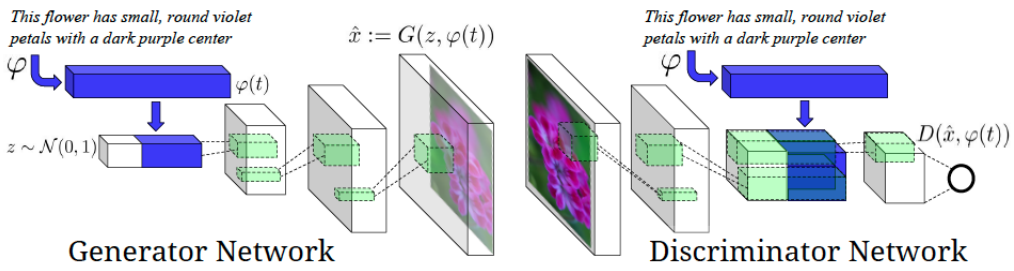


Figure 1.

Image generation that works on given text query and noise sample.

In the discriminator D, several layers of stride with spatial batch normalization is used where results are followed by Leaky ReLU. Dimentionality of the description embedding ($\varphi(t)$) in a fully connected layer is generated by rectification. The spatial dimension of the discriminator is 4 * 4. DCGan performs 1 * 1 convolution with rectification and final results are calculated by 4*4 convolution from D.

4.2. Training

Adam optimiser is used for training as it maintains a separate learning rate for each parameter of the model. The algorithm uses the gradient statistics for gradient scaling. The parameters B_1 and B_2 handles the moving averages. The learning rate is set to 0.0002 for both networks . To generate 600 epochs, model is trained with $B_1 = 0.5$ and $B_2 = 0.9$. For tect feature, a deep convolutional recurrent text encoder should be pre-trained to joint embedding of text caption. I have used the MS-COCO images where pretrained model was available.

To tain same GAN architecture was used for the MS-COCO datasets. The training image size was set to 64 * 64 * 3. 1024-dimensional embedding was produced by text encoder. Those embedding were used with 128 dimensions in both generator and discriminator.

4.3. Testing

In general to test the models, it learns visual semantics text description.Description- image mappings is used to generate images $G(z, (1-t)e_1 + te_2)$ from interpolations between two text embedding e_1 and e_2 where t becomes 0 to 1.

On general scale, already implemented results were:

4.4. Separate the object and background with external tools

Once I got the results from the text to images with objects and separately a background using text to images. The images were modified to make a same background which was automatically done in VideoGAN algorithms that is shown in following diagram:

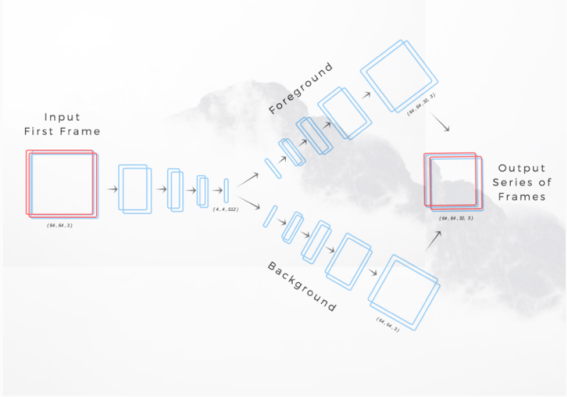


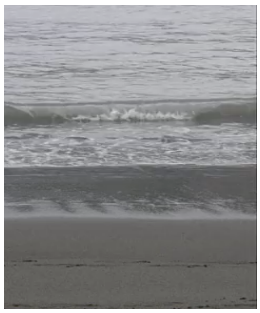
Figure: 1.6 After changing the generator quantities to 1 and specifying my caption to generate images for objects and background following results were opt which were used to generate animation while the resulting frames were modified manually to generate same background frames:

4.5. Results

The below results are straight result got from text to image using high quality dataset is trained manually and using MS-COCO data set for training.

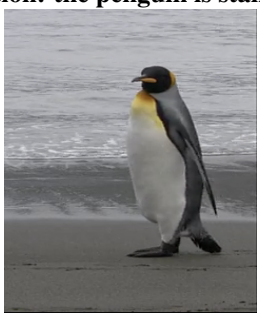
The below caption is used just for the purpose of background.

Caption: beach with white sand and water

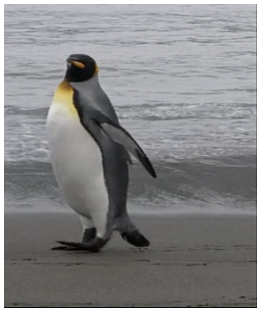


The below results are after editing the output images of text to images with the help of Photoshop for making same background as we got from text to images.

Caption: the penguin is standing



Caption: the penguin is walking



4.6. Motion Estimation using Optimal Flow

To estimate motion for finding in between frame,the proposed estimation algorithm is using block matching algorithm and the optimal flow algorithm those are explained in the sections 3.4 and 3.5.

First of all. block matching algorithm is used to find out integer pixel displacement ($\overline{\Delta x}$,$\overline{\Delta y}$). the found displacements is considered as a good integer estimate of $(\Delta x, \Delta y)$

After finding displacements, image block of particular displacement is moved to its direction(For x moves in x direction similarly with y direction for Δy) In further process, Taylor series approximation is used to refine the search. Due to shifted image is different from the real image as below: $F(x + \delta x, y + \delta y) is real image$ Then shift will be $(\delta x, \delta y)$ where both shifts are less then 1. Thus, The overall shift can be found by below derivations:

$$\Delta x = \overline{x} + \delta x \Delta y = \overline{x} + \delta y \tag{11}$$

Any block matching algorithm that uses a interpolation prevents algorithm to use any separate interpolation while finding motion estimation. For example, bilateral ME, three step search and others can be used for block matching algorithms where already interpolation is used.

The following formula shows the optimal displacement

$$\widehat{\Delta x} = \frac{\sum_x f'(x)[g(x) - f(x)]}{\sum_x [f'(x)]^2}. \tag{11}$$

$(\widehat{\Delta x})$
Where,

$$g(x) = f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{1}{2} f''(\xi)(\Delta x)^2. \tag{13}$$

This way, proposed motion estimation algorithm finds out in between image using result images of section 4.5.

4.6.1 Results

The above results shows the motion estimation. The motion was not correctly calculated which is img_m image in the chart above. In Chart, it also shows difference between two frames 0 and 1 with a effect using PSNR.

5. Comparison with VideoGAN

For Video generation I used external bilateral Interpolation (implemented in matlab) to make more frames in between and better video motion[optional] (used for Assignment purpose only).
The Gif result in frames by VideoGAN algorithm.

The Gif result in frames using proposed method

6. Conclusion

Overall, still text to animation or text to videos have vast opportunities to develop them where concept GAN is one the crucial part for most of the technique. Still there are not satisfied results achieved for continuous manner. Experimenting already implemented algorithms such as pixel generation, VideoGan, and from the result of CRAFT, it seems the big models of videos are needed to trained. The one of the drawback lead me to go to downward method and search a more appropriate way to generate text to animation. Although, Proposed method or experimented method have more manual work like need of using photoshop or other tools for background changes and object placement. However, using DCGan to generate text to images, then resetting background using tools gives enough motion frames and from those using motion estimation and interpolation, final result is achieved.(Additionally, gif file of video is made using video tools.)

7. references

[1]Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In ECCV. 2014..

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NIPS, 2014.

[3] Liu, Z.; Yeh, R.; Tang, X.; Liu, Y.; and Agarwala, A. 2017. Video frame synthesis using deep voxel flow. ICCV

[4] M. Figueiredo, R. Nowak, and S. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 586–597.

[5] To Create What You Tell: Generating Videos from Captions. (2018).

[6] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text-to-image synthesis. In Proceedings of The 33rd International Conference on Machine Learning, 2016.

[7] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In ICCV, 2017.

[8] arXiv:1804.03608v1 Imagine This! Scripts to Compositions to Videos , 2018

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 27, pages 26722680. Curran

Associates, Inc., 2014.

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv, 2015

[11] Scott Reed, Zeynep Akata, Bernt Schiele, and Honglak Lee. Learning deep representations of ne-grained visual descriptions. In IEEE Computer Vision and Pattern Recognition, 2016.

[12]A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In CVPR, 2015.