Building Annotated Image Dataset for Myanmar Text to Image Synthesis

Nan Kham Htwe
Natural Language Processing Lab
University of Computer Studies, Yangon
Yangon, Myanmar
nangkhamhtwe@ucsy.edu.mm

Win Pa Pa
Natural Language Processing Lab
University of Computer Studies, Yangon
Yangon, Myanmar
winpapa@ucsy.edu.mm

***Abstract*—** Text to image synthesis is the translation of images from the input language text. The learning process can become easier when the spoken words can visualize with the images. It is one of the popular research field in combination of NLP and computer vision. Generative Adversarial Networks (GAN) have growth in the generation of images from text descriptions. We build the baseline system of Myanmar text to image synthesis and a type of annotated images dataset because there is not efficient annotated image dataset to be used in this implementation. It was created by using partial part of Oxford-102 flowers dataset. Word2Vec algorithms is used to convert word to vectors for the input sentence to GAN. GAN is applied for generation of images from Myanmar language text. This is the first text to image generation using GAN in Myanmar. The two-evaluation metrics are used to measure the quality of images. The quality of the generated images is evaluated using Inception score. The Fréchet Inception Distance (FID) is used to measure the distance between the real images (images from original dataset) and the generated images from the model.

Keywords—GAN, Word2vec, Inception Score, FID

1. INTRODUCTION

The generation of images from text descriptions is one of popular research field in combination of computer vision and natural language processing. The understanding of the relationship between visual contents and natural languages is an essential step for artificial intelligence [1] e.g. image search and video understanding. Natural language processing provides a general and flexible interface for presenting objects in the visualization form of the categories. [4].

GAN has widely used to generate various and high-quality of images according to natural language descriptions [2]. GAN has also used in many applications such as image-to-image translation, clothing translation, face aging and text to image generation. There are many annotated image dataset (Oxford-102 flowers, CUB birds, and COCO datasets, etc.) in generation of realistic images from English text descriptions. There is the first annotated image dataset in Myanmar language[17] that is extended from Flickr8k dataset. But this dataset is not efficient in the training of translating text to images because there is multi-objects and contributions of each object in this dataset is very little to effectively learn their features. Therefore, the first contribution is building annotated images dataset using Oxford-102 flowers dataset [13] for generation of images from Myanmar text. The captions for each image in the flower dataset are created manually. There are five captions for each image in the dataset.

Text to image generation requires two types of data: text and image. In translation of text to images, problems can be subdivided into two types: image generation and text representation. Text representation requires to efficiently learn feature representation of words within the sentence. In this paper, we used Word2vec model to generate vector representation of words for the input sentence to GAN.GAN is used to generate the images from the input text.

In this paper, section 2 is related works, and the theoretical backgrounds are described in section 3. Section 4 contains the explanation of the training steps, and result and evaluation are in section 5. The final section describes the conclusion and future works of this paper.

1.ပန်းပွင့်များ၏ပွင့်ချပ်များသည်ပန်းရောင်ရှိပြီးအဝါရောင်အလယ်ဗဟိုရှိတယ်
2.ခရမ်းရောင်ပွင့်ချပ်များသည်အဖြူရောင်အရိပ်ရှိတယ်
3.ခရမ်းရောင်ပွင့်ချပ်များရှိသည့်ပန်းတစ်ပွင့်ဖြစ်တယ်
4.ဒီပန်းပွင့်အဝါရောင်စင်တာနှင့်အတူသေးငယ်တဲ့ပန်းရောင်ပွင့်ချပ်ရှိပါတယ်
5.ဒီပန်းပွင့်တွင်ကြီးမားသောပန်းရောင်ပွင့်ချပ်ရှိပါတယ်

Fig 1: The real image and its related captions in Myanmar

2. RELATED WORKS

There is one difficulty in deep learning is that the converting of images from natural language descriptions is highly multimodal because there are many possible configurations of pixels that can correctly visualize text descriptions. To solve this problems, Scott Reed et al. [1] proposed a new architecture for generating images from text. In this paper, they used GAN for generating images and char CNN-RNN encoder for encoding text descriptions. This GAN can interpret text descriptions to images of birds and flowers.

Adithya Viswanathan et al. [8], describes learning process becomes easier when the spoken words can visualize with images. Therefore, they proposed a new novel implementation that can visualize image from language descriptions. In this

paper, text descriptions are encoded with RNN-CNN encoding method along with GAN. This model can generate the flower images from text.

The Reed et al. [1], proposed text-to-image GAN that can only output the images in size of 64x64. Later, Zhan et al [10], proposed StackGAN++ extended from StackGAN that can output the images in size of 256 x 256. But StackGAN++ requires multiple generators and discriminators (64 x 64, 128 x 128, 256 x 256) for generation of images. Thus, Xin Huang et al. [3], proposed Hierarchically-fuse GAN with one discriminator to generate 256 x 256 images. In this paper, they also proposed Attentional Multimodal Similarity Model of AttnGAN that converts a sentence condition from text descriptions and two-word conditions.

Aifa Nur Amalia et al. [7] describes most text to image generation with GAN are taken from real images. But GAN has not been widely used for the dataset with artistic values. Therfore, they proposed a new kind of dataset that contains batik patterns for generation of images. Batik is not only a cultural heritage but also Indonesian identity.

In this paper, a new type of annotated image dataset in Myanmar language is created manually using the partial part(images) of the Oxford-102 flowers dataset. Text descriptions are embedded in the preprocessing steps to faster the process of generation images. Only one of the embedded captions of each image in the training batch is randomly chosen to train the model. Each image in the batches are flipped to horizontal to efficiently extract features of the images and to generate the higher quality of images in the training step.

3. METHODOLOGY

In this paper, word2vec is used for converting words to vectors, and GAN is used to generate the images. Therefore, all above theoretical information will be discussed in this section.

A. Word2Vec

Word2vec is a method in NLP and uses a neural network model to convert vector representations for words in a large text corpus. This is done by making target and context word pairs depending on the window size. It represents each word in a sentence with a list of real numbers called a vector. The resulting word vectors can be used as features in many machine learning and NLP applications. The two categories of word2vec are (1) continuous bags of words (CBOW) and Skip-gram.

In CBOW, the word is predicted by using windows of surrounding context. All words use the same projection layer. For example, the word will give w_i, if w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2} are given words current or contexts. CBOW contains three layers as shown in Fig 3. CBOW predicts w(t) from its context words w(t-2), w(t-1), w(t+1), w(t+2). From this point, the maximum likelihood and the objective function of the CBOW is:

$$L = \sum_{w(t)\in C} \log P(w(t)|w(c)) \qquad (2)$$

In equation, C denotes as corpus and w(c) is the context of w(t).

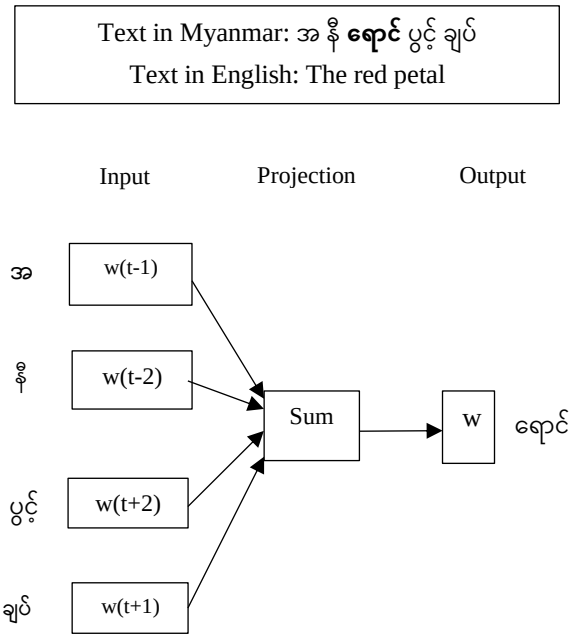

Fig 3: Architecture of CBOW

In Skip-gram, the surrounding words are predicted by using the current word in the sequence. It is one of the unsupervised learning techniques use to find the related words for a given word. The input word is the current word and the output is the context also called the surrounding words. The structure of Skip-gram is shown in fig 4:

Give a sequence of training words w(t-2), w(t-1), w(t), w(t+1), w(t+2), the maximum likelihood and the objective function of skip gram is:

$$L = \sum_{w(t)\in C} \log P(w_j|w_i) \qquad (4)$$

Where w_j is context word, w_i is the current word and C is the corpus.

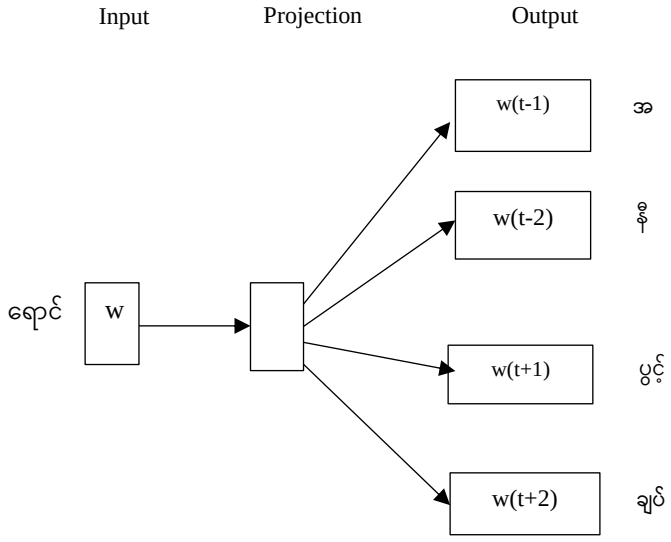

Fig 4: Architecture of Skip-gram

B. Generative Adversarial Networks (GAN)

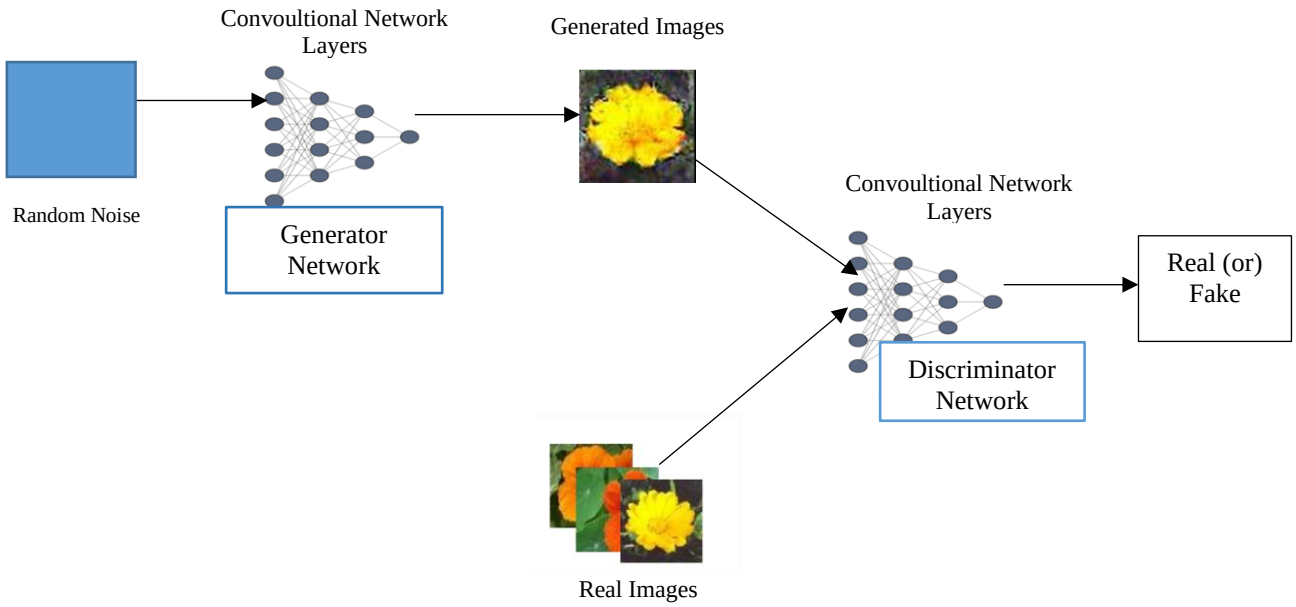

Fig 2: Framework of Generative Adversarial Network

GAN consists of two networks: generator (G) and discriminator (D). These two networks are composed of convolutional network layers. The generator transforms random noise samples (z) into fake images G(z). The discriminator network receives the two inputs: (1) the training dataset x (real images) (2) the fake samples generated by the generator. Then the discriminator tries to classify whether the images from the generator as genuine or not by computing probability distribution D(x) for real training dataset and D(G(z)) for fake samples from generator.

On the other hand, D(x) represents the probability that are originated for real data distribution P_{data}. Hence, D(G(z)) can be used to evaluate the quality of generated image with related to real images x.When the discriminator is optimal, it may gain the generator G and the training have continue to lower the accuracy level of the discriminator (D). The discriminator can stop working if the generator distribution is able to match the real data distribution completely and the images generated by generator cannot be distinguished from the real training dataset.

In GAN, discriminator (D) try to maximize the probability of correctly classifying the image as fake and otherwise, Generator (G) try to minimize for that probability. The min max game of G and D is described as follows:

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

4. Implementation Details

The preprocessing steps and implementation of the training for generation of images based on text descriptions are described in this section.

A. Creating the dataset

In this step, we created the text corpus using Oxford-102 flowers dataset. We made manually captions for each image in the dataset. There are over 8000 images in the Oxford-flowers dataset. But in this training, we used only 1500 images and 5 captions for each image. The captions for remaining images have continued to create and used in the next implementation.

B. Preprocessing the dataset

Word segmentation and creating vectors for each word in a sentence are taken in this step. There is not separated white space in Myanmar sentence. In this paper, sentence segmentation is done by using Myanmar syllable segmentation [11]. The segmented words are converted into word vectors by using word2vec algorithm. All of the extracted vectors for each caption are then stored in one hdf5 file. The data set stored in this file is used in the training step.

C. Training Steps

The system is implemented using DCGAN [16]. There are two components generator network and discriminator network. The generator is designed to output the artificial images based on natural language descriptions. The role of discriminator is to correctly classify the image generated by the generator is real or fake.

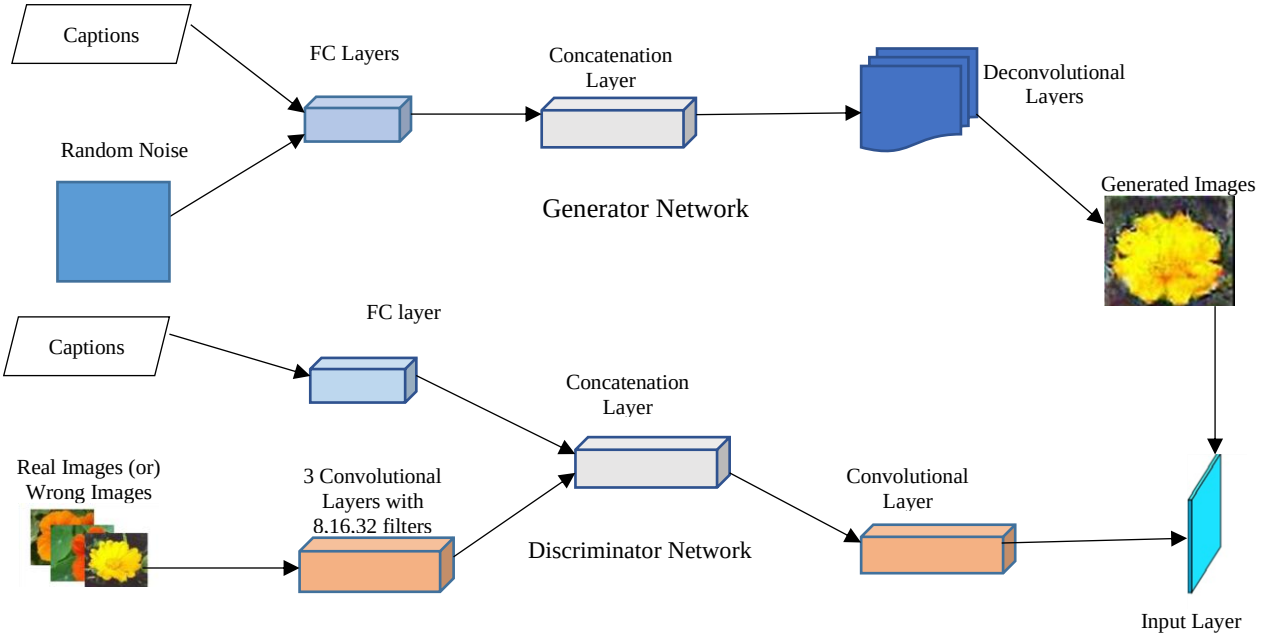

Fig 5: Implementation of Generator and Discriminator

In generator, we first sample the random noise vector whose dimension is 100 and randomly select one of the captions among five captions of each images in the training batch. The captions were already converted to word vectors by using word2vec in the preprocessing steps. The selected word vectors are compressed to the dimension of 128 by using FC (fully-connected) layer and leaky-Relu. And then it is concatenated by noise vector. The concatenated output with dimension of 8192 is then reshaped into 4 x 4x 512. The resulting output is then feed-forward through 4 deconvolutional layer, each layer contains 2D batch norm layer and Relu activation. The result is forward to the output layer. The output is returned in range of [-1,1]. The image generation in generator is conditioned on text descriptions and a random noise sample.

There are 3 convolutional layers (8,16 and 32 filters) and Fully-connected layers in the discriminator. The convolution is applied with a stride of 2 with special normalization. The activation function for these layers is LeakyRelu. There are three types of inputs to the discriminator: (1) the captions with matching real images (images from training data), (2) wrong images (shuffle of the real images) with captions, and (3) fake images (artificial images generated from generator) with captions. The input image size to the discriminator is 64 x 64 dimensions. The real images and wrong images are concatenated with one of the randomly chosen caption among the captions of each image (captions had already embedded to word vectors with Word2vec). Concatenation is performed after convolution of the images. The resulting output is reshaped and applied convolution followed by sigmoid activation.

Finally, the two concatenated outputs: (1) real images with captions, (2) wrong images with captions, and fake images with captions from the generator are given as inputs to the discriminator. The final score is calculated from discriminator. The calculated score from the discriminator is used to update the weights of these two networks. We use ADAM solver in learning and the rate of learning is 0.0002. The batch size and the training epoch are 64 and 600 respectively. The dimension of the generated images is 64 x 64.

5. Results and Evaluation

A. *Experiment Results*

After training steps, the query text descriptions are fed to model to test that model can generate the matching image with text. Firstly, the query text description is saved in a single text file and encoded these descriptions by using word2vec algorithm. Secondly, the converted word vectors are saved in hdf5 format and fed to model to generate the images. The model can only generate the image with the dimension of 64x 64. It can't learn the features of multi-objects. Therefore, it can only output a single domain object such as flower images. The results of the generated images are shown in figure [6]:

Query Text	Generated Images
အဖြူရောင်ပွင့်ဖတ်နှင့်ပန်းမှာအဝါရောင်ဝတ်ဆံရှိတယ် The flower with white petals has yellow stamen.	
ပန်းရောင်ပန်းပွင့်သည်ရေပေါ်တွင်ပွင့်နေသည် The pink flower is floating on water.	

ပန်းပွင့်သည်အညိုရောင်အလယ်ဗဟိုရှိပြီးအဝါရောင်ပွင့်ဖတ်ရှိတယ်

The flower has the brown center and yellow petal.

ပန်းပွင့် မှာ အဖြူရောင်ပွင့်ဖတ်နှင့်အဝါရောင် မျိုးစေ့အိမ်ရှိတယ်

The flower has white petal and yellow ovary.

ပန်းပွင့်သည်လိမ္မော်ရောင်ဖြစ်ပြီးလိမ္မော်ရောင်ပွင့် ဖတ်ရှိတယ်

The color of flower and its petal is orange.

ပန်းပွင့် ဟာ တောင်ပံပုံစံ အ သွင်အပြင် နဲ့ ခရမ်း ရောင် ရှိ ပါတယ်

The wing-shaped flower has purple in color.

Fig 6: Results of generated images

B. Evaluating Metrics

The quality of the generated image is evaluated using Inception Score and Fréchet Inception Distance (FID)

(i) Inception Score

The Inceptions score [9] is used to measure the quality of the generated images from the generative models. It is calculated by using Inception v3 Network and calculates the statistic as output for the generated images, which can be denoted as:

$$I = \exp\left(\mathbb{E}_{x\sim p_g} D_{KL}\left(p(y|x)||p(y)\right)\right) \qquad (6)$$

where x is the artificial image, and y is the label predicted by that model. The larger inception score represents the generated images has higher quality.

(ii) Fréchet Inception Distance (FID)

FID [16] is used to measure the FID score between the generated images and the real images. It has more robust to noise and can easily evaluate the diversity of images than Inception Score. The smaller FID score means the better quality of generated images. The equation to calculate FID score is:

$$FID = \|\mu_{r-}\mu_g\|^2 + T_r(\sum_r + \sum_g - 2(\sum_r\sum_g)^{1/2}) \qquad (7)$$

The equation represents the FID score between the real images, r, and the generated images, g. The inception v3 Network model is used to extract the feature maps of the images in calculating FID score. Then the multivariate Gaussian distribution is model to learn the distribution of the feature maps. It has a mean of μ and a covariance of ∑, which are used to calculate the FID score.

C. Result Evaluation

The experiments of of image generation was made by querying input text. The testing of image generation was made by querying the number of 50 text descriptions.The inception score and FID metrics is used in evaluation the results of the generated images. The quality of the generated images is evaluated using inception model .The distance between the real images and fake images is measured with FID score. The result of evaluated scores are as shown in Table [I].

Table I : Inception score and FID score

Score	
Inception Score	FID
1.72 ± 0.01	222.34

CONCLUTION AND FUTURE WORKS

In this paper, we build the first Myanmar text to Image Synthesis using GAN. And we have created an annotated image dataset to be used in this implementation. The captions are embedded using Word2vec in the preprocessing steps to faster the training process and images are generated using GAN framework. The generated images are then evaluated by inception v3 model and FID. The dataset is continuing to create with the remaining images of the Oxford-102 flowers. Moreover, the feature of images that are generated from GAN will have to improve by using many annotated images.

REFERENCES

[1] M. Zhu, P. Pan, W. Chen and Y. Yang, "DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-To-Image Synthesis," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), USA, 2019, pp. 5795-5803.

[2] M. Yuan and Y. Peng, "Bridge-GAN: Interpretable Representation Learning for Text-to-image Synthesis," in IEEE Transactions on Circuits and Systems for Video Technology.

[3] X. Huang, M. Wang and M. Gong, "Hierarchically-Fused Generative Adversarial Network for Text to Realistic Image Synthesis," *2019 16th Conference on Computer and Robot Vision (CRV)*, Kingston, QC, Canada, 2019, pp. 73-80.

[4] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, andH. Lee, "Generative adversarial text to image synthesis,"arXiv preprint arXiv:1605.05396, 2016.

[5] T. Qiao, J. Zhang, D. Xu and D. Tao, "MirrorGAN: Learning Text-ToImage Generation by Redescription," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA,USA, 2019, pp. 1505-1514.

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B.,Warde-Farley, D., Ozair, S., Courville, A., and Bengio,Y. Generative adversarial nets. In NIPS, 2014.

[7] A. N. Amalia, A. F. Huda, D. R. Ramdania and M. Irfan, "Making a Batik Dataset for Text to Image Synthesis Using Generative Adversarial Networks," *2019 IEEE 5th International Conference on Wireless and Telematics (ICWT)*, Yogyakarta, Indonesia, 2019, pp. 1-7, doi: 10.1109/ICWT47785.2019.8978233.

[8] A. Viswanathan, B. Mehta, M. P. Bhavatarini and H. R. Mamatha, "Text to Image Translation using Generative Adversarial Networks," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, 2018, pp. 1648-1654.

[9] M. Barratt, Shane, and Rishi Sharma. "A note on the inception score." *arXiv preprint arXiv:1801.01973* (2018).

[10] H. Zhang *et al*., "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1947-1962.

[11] Ye Kyaw Thu**,** Andrew Finch, Yoshinori Sagisaka and Eiichiro Sumita, "A Study of Myanmar Word Segmentation Schemes for Statistical Machine Translation", *Proceedings of the 11th International Conference on Computer Applications (ICCA)*, February 26~27, 2013, Yangon, Myanmar, pp. 167-179

[12] M. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, Bhubaneswar, 2008, pp. 722-729, doi: 10.1109/ICVGIP.2008.47.

[13] D. C. Dowson and B. V. Landau. The Fréchet distance between multivariate normal distributions.Journal of Multivariate Analysis, 12:450–455, 1982.

[14] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan and Y. Zheng, "Recent Progress on Generative Adversarial Networks (GANs): A Survey," in *IEEE Access*, vol. 7, pp. 36322-36333, 2019, doi: 10.1109/ACCESS.2019.2905015.

[15] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium", *Proc. Adv. Neural Inf. Process. Syst.*, pp. 6626-6637, 2017

[16] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. 2015 Nov 19.

[17] San Pa Pa Aung, Win Pa Pa, Tin Lay Nwe, "*Automatic Myanmar Image Captioning using CNN and LSTM-Based Language Model*", 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020), Merseille, pp 139–143, France, 2020