RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

TEXT TO IMAGE TRANSLATION

Spring 2021 534 Computer Vision

Submission by -
Tianzhi Cao (tc796)
Abhishek Bhatt (ab2083)

Supervised by -
Prof. Ahmed Elgammal

PROBLEM OVERVIEW

- **Problem**

$\text{Text} \xrightarrow{\text{text-to-image (T2I)}} \text{Image}$

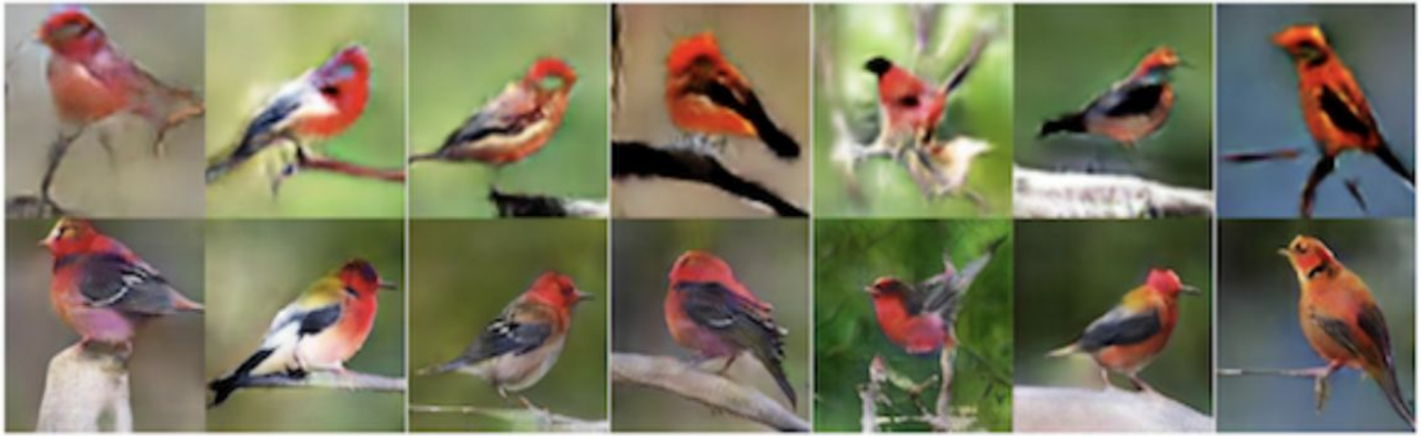
Text:
- Natural language description of image
- Sequence of words

Image:
- Whether generated images match the text description
- Quality and variety of images

- **Applications**

- Data Augmentation
- Photo-editing
- Computer-aided Designing

The small bird has a red head with feathers that fade from red to gray from head to tail



T2I KEY CHALLENGES

Challenge 1

Constructing good text embeddings

context of the word as represented by Word2Vec doesn’t capture the visual properties very well

Challenge 2

Multimodal learning

- many different images of birds with correspond to the text description “bird”

- many different accents that would result in different sounds corresponding to the text “bird”

- symmetric structured joint embedding of images and text descriptions
- adaptive loss function in GANs, well-suited for multi-modal tasks

PROJECT GOALS

- Explore various approaches to the text to image translation problem
- Analyze pre-implemented models and their generated results
- Suggest, implement and experiment with enhancements over existing methods
- Develop functional and implementational know how of CNNs, Sequence models and GAN architectures

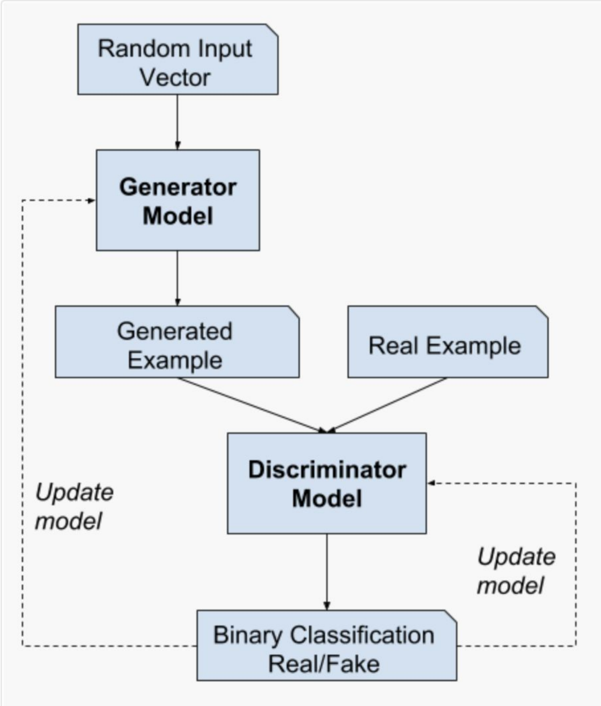
USING GANs FOR T2I

GANs : model architecture for training a generative model [Ian Goodfellow, et al. 2014]
Deep Convolutional GAN or *DCGAN* [Reed et al. 2016]

- *GENERATOR* : used to generate new examples from the problem domain [Inference]

- *DISCRIMINATOR* : classifies examples as real (from the domain) or fake (generated)

- *ADVERSARIAL* : generator is trained in a zero-sum game to compete against the discriminator, until discriminator is fooled about half the time [Supervised Learning]



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

RELATED WORK

- Approach 1: Generative Adversarial Text to Image Synthesis (Reed et al. 2016)
    - Developed a GAN architecture and training strategy for compelling T2I
    - Synthesized 64x64 images matching at least part of the caption and reflecting color information
    - Generated scenes are not usually coherent

- Approach 2: StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks (Zhang et al. 2017)
    - Used two-stage GANs (StackGAN) to generate high resolution 256x256 images
    - Proposed a novel Conditioning Augmentation for stabilized training and diversity of generated samples
    - Generated images accurately reflect color changes and plausible shapes

- Approach 3: AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks (Xu et al. 2018)
    - Developed a novel Attention-based GAN architecture for fine-grained T2I synthesis
    - Created a Deep Attentional Multimodal Similarity Model to capture text-image correspondences for high quality generation
    - Significantly improved performance on complex scenes (e.g. COCO dataset)

PROPOSED ENHANCEMENT

Enhancing text embedding for GAN-based Text-to-Image Translation

- PyTorch Implementation
- Observations and Discussion

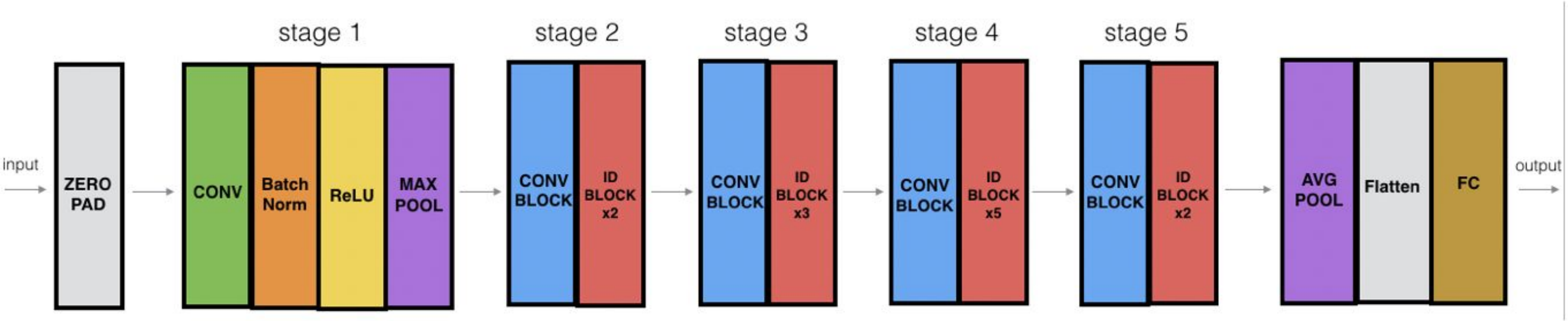
APPROACH

- Dataset
  - Oxford-102 flowers Dataset
  - 102 flower categories,
    each category 40 to 258 flower images
  - Randomly pick 1 of the 10 text descriptions corresponding to each image

- Preparation
  - 8190 image-caption pairs
  - Build text vocabulary
  - Train-dev-test split : 80%, 10%, 10%
  - Training images : resize to 64x64 due to computational constraints

APPROACH - Model

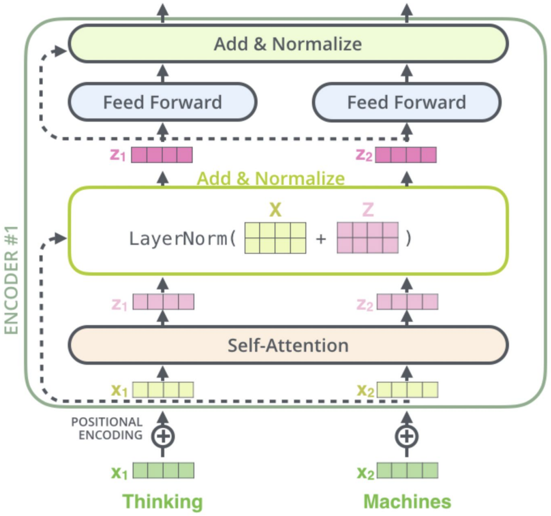
- Image Encoder
  - ResNet-50 convolutional model
  - Text encodings used to compute Mean Squared Error against the text embeddings
  - All parameters learnt jointly with the GAN

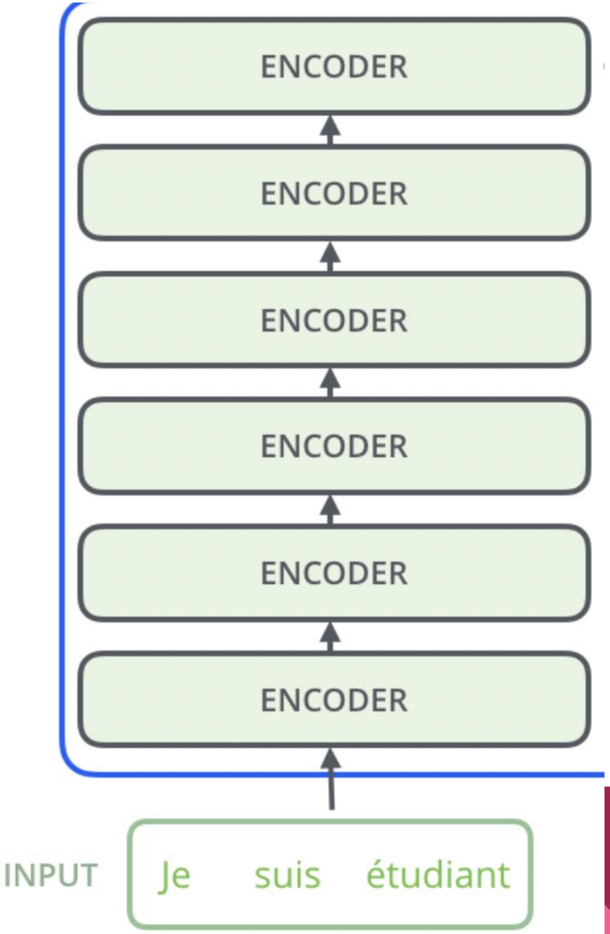


ResNet-50 Model

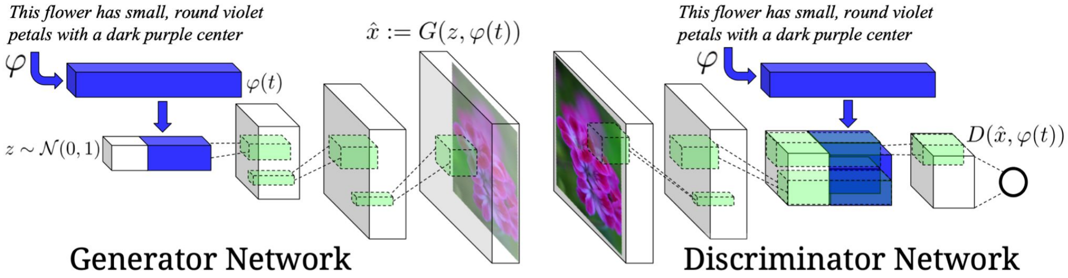
APPROACH - Model

- Text Encoder
  - Stack of transformers as text encoder
  - Multi-head self-attention with positional encodings
  - All parameters learnt jointly with the GAN





APPROACH - Model


This flower has small, round violet petals with a dark purple center
φ
$\varphi(t)$
$z \sim \mathcal{N}(0,1)$
$\hat{x} := G(z, \varphi(t))$
Generator Network
This flower has small, round violet petals with a dark purple center
φ
$D(\hat{x}, \varphi(t))$
Discriminator Network

- GAN
  - Vanilla DC-GAN
    - Generator
      - Transposed convolutions with batch normalization and ReLU / tanh activations
      - Input - output from our text encoder appended to random noise vector
      - Output - 64x64 synthetic RGB images
    - Discriminator (Image classifier)
      - Convolutions with batch normalization and LeakyReLU activation
      - Input - batch of images
      - Output - labels 1 (real image in dataset) or 0 (fake generated image)

APPROACH - Loss Functions

- Discriminator : Binary Cross Entropy loss between predicted and ground truth labels for real as well as fake images

$$\mathcal{L}_D = -\frac{1}{N}\sum_{i=1}^{N} \log D(I_i) + \log(1 - D(G_i))$$

- Generator : BCE loss between predicted labels for fake images and ground truth labels for real images

$$\mathcal{L}_{Generative} = -\frac{1}{N}\sum_{i=1}^{N} \log D(G_i)$$

- Text encoder : Mean Squared Error between the input image encodings (from image encoder) and the corresponding text embeddings (from text encoder)

$$\mathcal{L}_{Embedding} = \frac{1}{N}\sum_{i=1}^{N} (imageEnc(I_i) - textEnc(T_i))^2$$

$$\mathcal{L}_G = \mathcal{L}_{Generative} + \mathcal{L}_{Embedding}$$

$$\theta_D^*, \theta_G^* = \arg\min_{\theta_D, \theta_G} \mathcal{L}_D(\theta_D) + \mathcal{L}_G(\theta_G)$$

RESULTS

Dev Set

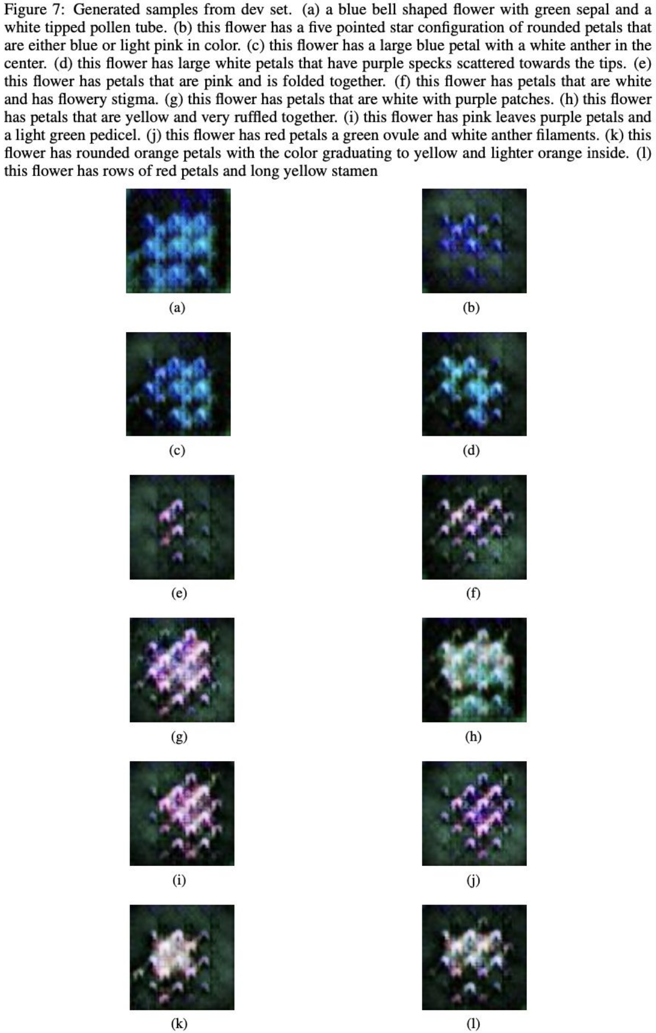


Figure 7: Generated samples from dev set. (a) a blue bell shaped flower with green sepal and a white tipped pollen tube. (b) this flower has a five pointed star configuration of rounded petals that are either blue or light pink in color. (c) this flower has a large blue petal with a white anther in the center. (d) this flower has large white petals that have purple specks scattered towards the tips. (e) this flower has petals that are pink and is folded together. (f) this flower has petals that are white and has flowery stigma. (g) this flower has petals that are white with purple patches. (h) this flower has petals that are yellow and very ruffled together. (i) this flower has pink leaves purple petals and a light green pedicel. (j) this flower has red petals a green ovule and white anther filaments. (k) this flower has rounded orange petals with the color graduating to yellow and lighter orange inside. (l) this flower has rows of red petals and long yellow stamen

Test Set

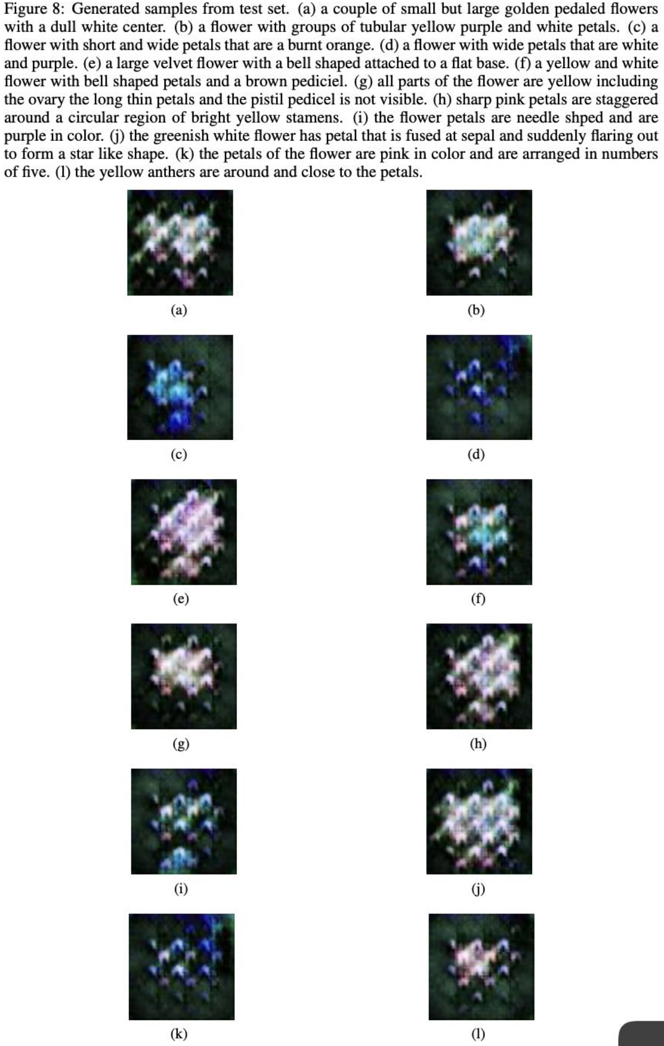


Figure 8: Generated samples from test set. (a) a couple of small but large golden pedaled flowers with a dull white center. (b) a flower with groups of tubular yellow purple and white petals. (c) a flower with short and wide petals that are a burnt orange. (d) a flower with wide petals that are white and purple. (e) a large velvet flower with a bell shaped attached to a flat base. (f) a yellow and white flower with bell shaped petals and a brown pediciel. (g) all parts of the flower are yellow including the ovary the long thin petals and the pistil pedicel is not visible. (h) sharp pink petals are staggered around a circular region of bright yellow stamens. (i) the flower petals are needle shped and are purple in color. (j) the greenish white flower has petal that is fused at sepal and suddenly flaring out to form a star like shape. (k) the petals of the flower are pink in color and are arranged in numbers of five. (l) the yellow anthers are around and close to the petals.

RESULTS

- Observations
  - Most of the generated samples simply contain colored patches or textures, especially when descriptions are more generic than specific
  - For some specific descriptions, the model does capture color and structural information, samples look like a colored flower-shaped object
  - Generated images are low resolution, and the scenes are not coherent

- Possible causes of underperformance
  - Small training size for learning word embeddings
  - Resizing input images for training
  - Very basic DC-GAN architecture
  - Global MSE loss between text and image encodings does not capture local correspondences

KEY TAKEAWAYS

- Learning better joint image-text embeddings crucial for T2I using GANs
- Current results do not reflect the effectiveness of our proposed enhancement
- Scope for improving implementation with better architecture and more compute
- Insights into T2I problem, SoTA models and challenges involved
- Introduction to research methodology to further explore T2I

REFERENCES

Articles

- https://jonathan-hui.medium.com/gan-some-cool-applications-of-gans-4c9ecca35900
- https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
- http://jalammar.github.io/illustrated-transformer/
- https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

Publications

- https://arxiv.org/pdf/1406.2661.pdf
- https://arxiv.org/pdf/1605.05396.pdf
- https://arxiv.org/pdf/1612.03242.pdf
- https://arxiv.org/pdf/1711.10485.pdf