TEXT-TO-IMAGE GENERATION

Milestone 2

1. PROBLEM STATEMENT

2. REVIEW ON MILESTONE 1

3. PROGRESS AND RESULTS

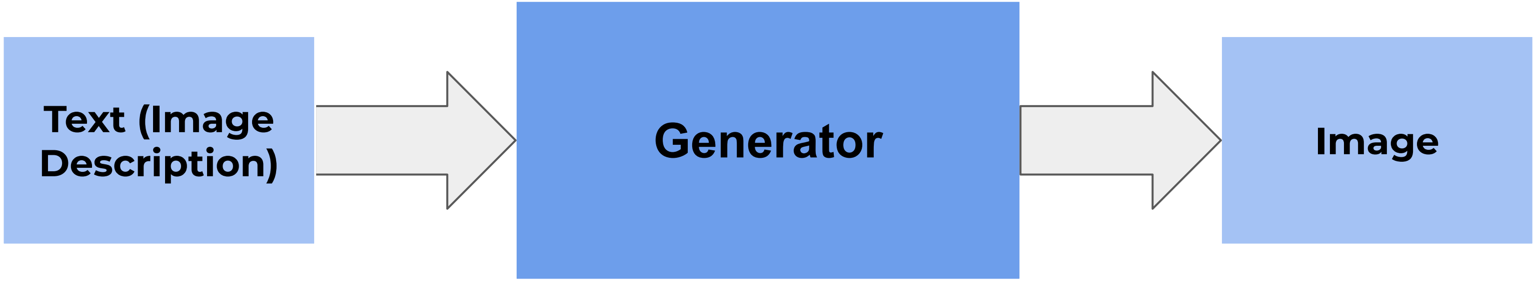
4. NEXT STEPS

1

PROBLEM STATEMENT

Problem

Translate text in the form of human-written description into image that is indistinguishable from realistic one



Examples[1]

a flower with long pink petals and raised orange stamen.

a sheep standing in an open grass field.

2

REVIEW ON MILESTONE 1

Related Work: DC-GAN (Reed et al., 2016)[1]

- Train a DC-GAN conditioned on text features encoded by a hybrid character-level Convolutional RNN

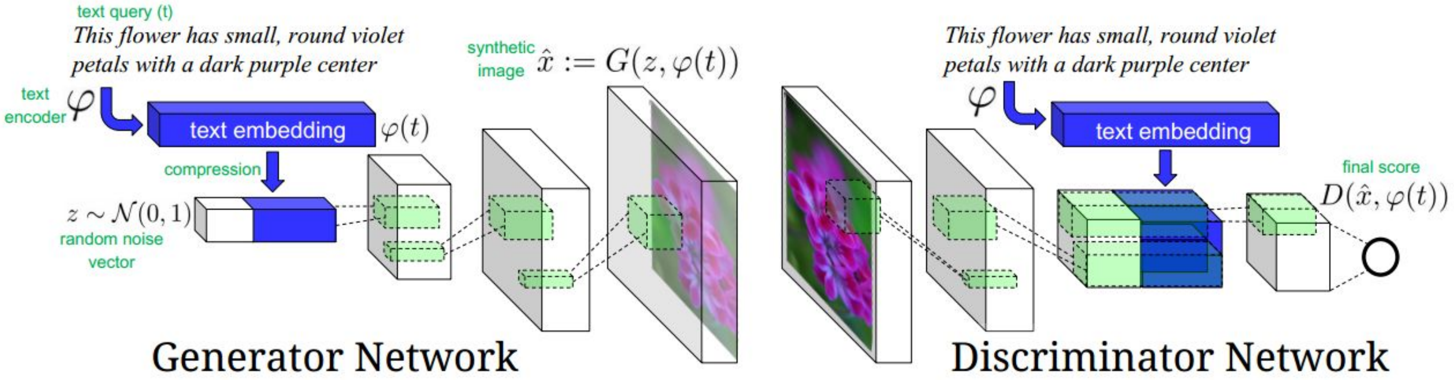
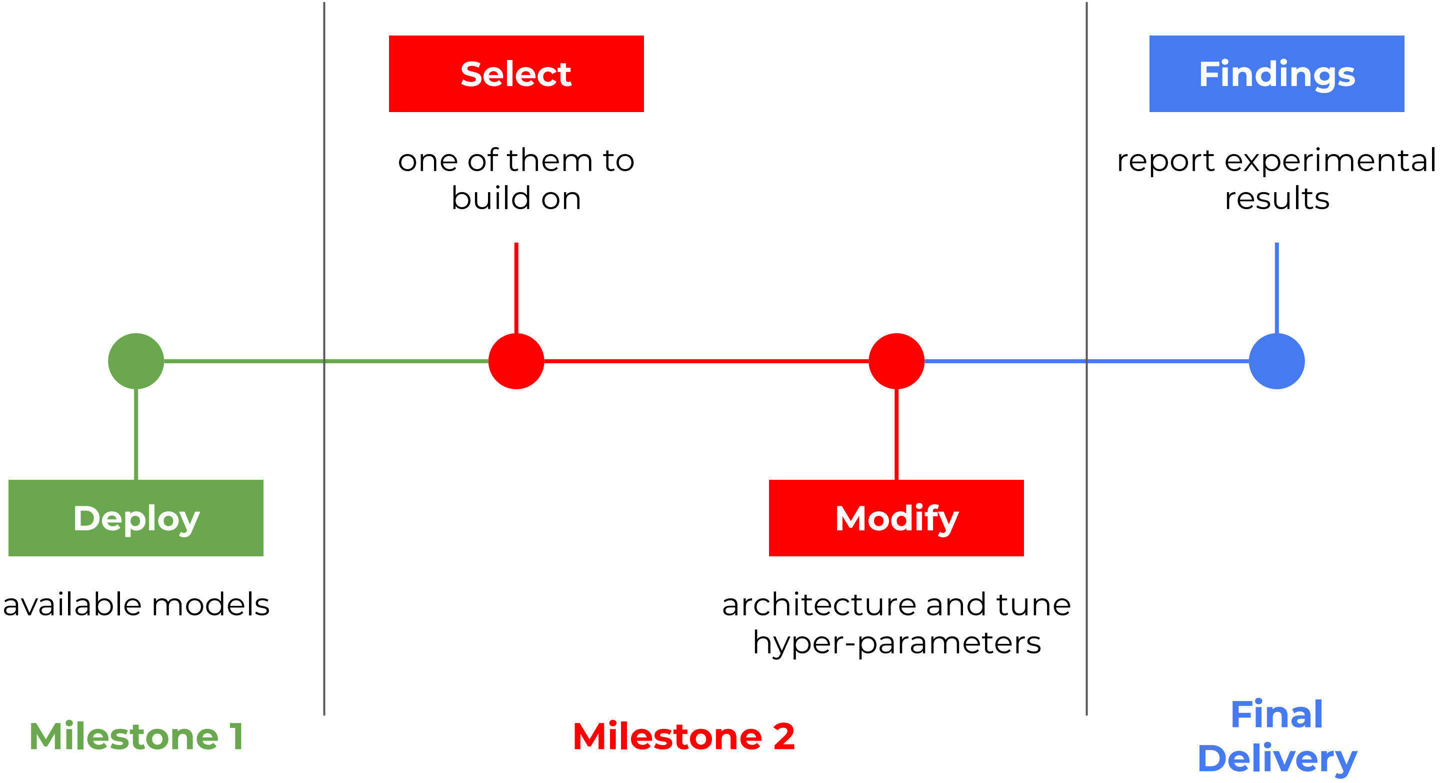


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

Time Plan



Deploy

available models

Select

one of them to build on

Modify

architecture and tune hyper-parameters

Findings

report experimental results

Milestone 1

Milestone 2

Final Delivery

Deploying original model

- Found 2 implementations of the paper in TensorFlow
  - https://github.com/paarthneekhara/text-to-image
  - https://github.com/zsdonghao/text-to-image
  - Try to deploy them, but found some errors

- Found implementation of the paper (by the author) in PyTorch
  - https://github.com/reedscot/icml2016
  - Deployed successfully, but can’t understand the code

3

PROGRESS AND RESULTS

Training the model

- Building on text-to-image[2] repository (TensorFlow)
- Training process take too much time and resources
- Tried 2 different approaches to overcome training time
    - Trained over specific classes of the dataset 102flowers[3]
    - Trained over randomly selected images (2000 images out of 8189)

Understanding what is going under the hood in details

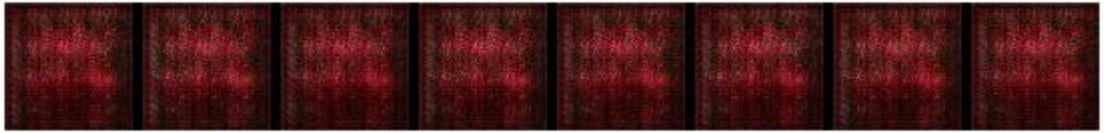
- Training phase
  - Sentence embedding is generated through skip-thoughts[4] for all the captions
  - Resulting Sentence embedded vector is fed to the CNN as its condition
  - Text-to-image GAN is then trained over the dataset
- Generative phase
  - Sentence embedding the required Input caption
  - Feed the input caption vector to our trained model

Hyper-Parameter tuning

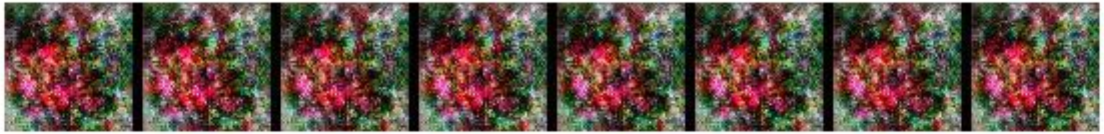
- Tuned different parameters which are
  - Learning rate
  - Batch size
  - Epochs number
  - Generator number of conv in first layer
  - Discriminator number of conv in first layer

Results

- For a model only trained on specific class contains 61 images
- Caption: this flower has petals that are pink and has a yellow center
  - 200 epochs

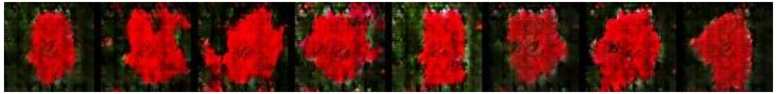


  - 600 epochs

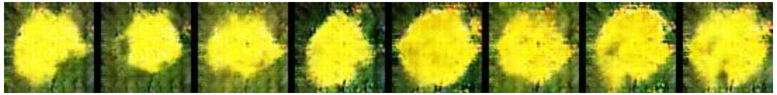


Results

- For a model only trained on quarter of the dataset we didn’t expect much
- Trying different captions
  - This is red flower



  - This is yellow flower



4

Next Steps

Deploying final model

- On our chosen hyper parameters we will train our model over the full dataset
- Train the model over different dataset

References

1. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016b). Generative adversarial text to image synthesis. Proceedings of the International Conference on Machine Learning (ICML). Available: https://arxiv.org/pdf/1605.05396.pdf

2. Paarth Neekhara text-to-image implementation in Tensorflow. https://github.com/paarthneekhara/text-to-image

3. Oxford 102-flowers dataset. https://www.robots.ox.ac.uk/~vgg/data/flowers/102/

4. Jamie Kiros skip-thoughts. https://github.com/ryankiros/skip-thoughts#getting-started

OUR TEAM

Mohamed Sharaf	(54)
Mahmoud Tarek	(63)

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

Thanks!