# Text-to-Image Synthesis TOWARD ‘TIP’

Stack-GAN - Two stage test to image generator using two Generative Adversarial Networks(GAN)

[[1612.03242] StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks (arxiv.org)](https://arxiv.org/abs/1612.03242)

Captioning Tool: <http://10.73.24.39:5000/> Password: your nickname

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experiment Details** | | **Eoch** | **Stage-1 (64 x 64)** | **Stage-2 (256 x 256)** | |
| **25 May 2023: Report:** | | | | | |
| exp1: sixray\_300\_caption\_stage\_1\_2023\_05\_19\_22\_44\_14  Pre-trained Embedding Model | | **1K, 1K** | **fake_samples_epoch_999**  Figure 1 | **fake_samples_epoch_999**  Figure 2 | |
| exp2: sixray\_600\_caption\_stage\_1\_2023\_05\_19\_22\_44\_14  Pre-trained Embedding Model | | **1K, 1K** | **fake_samples_epoch_999**  Figure 3 | **fake_samples_epoch_999**  Figure 4 | |
| exp3: sixray\_1107\_cap\_432\_file\_stage\_1\_2023\_05\_22\_18\_18\_01  Pre-trained Embedding Model | | **1K, 1K** | **fake_samples_epoch_950**  Figure 5  **Figure 3.1** | **fake_samples_epoch_950**  Figure 6  **Figure 3.2** | |
| exp4: sixray\_1347\_cap\_fasttext300D\_stage\_1\_2023\_05\_23\_15\_15\_11  Pre-trained Embedding Model | | **1K, 1K** | **fake_samples_epoch_999**  Figure 7  **Figure 4.1** | **fake_samples_epoch_999**  Figure 8 | |
| exp5: sixray\_1347\_cap\_stage\_1\_2023\_05\_24\_12\_57\_46  Pre-trained Embedding Model | | **500,500** | **fake_samples_epoch_499**  Figure 9  **Figure 5.1** | **fake_samples_epoch_499**  Figure 10 | |
| exp6: sixray\_1867\_ftt\_1000D\_skipgram\_stage\_1\_test\_2023\_05\_26\_10\_39\_26  **Trained** Embedding Model | | **500,500** | **fake_samples_epoch_499**  Figure 11  **Figure 6.1** | **fake_samples_epoch_499**  Figure 12 | |
| exp7: sixray\_2183\_ftt\_1000D\_skipgram\_stage\_1\_test\_2023\_05\_27\_04\_57\_09  **Trained** Embedding Model | | **600,600** | **fake_samples_epoch_999**  Figure 13  **Figure 7.1** | **fake_samples_epoch_599**  Figure 14 | |
| exp8: sixray\_2500\_ftt\_1000D\_cbow\_stage\_1\_test\_2023\_05\_29\_21\_29\_52  **Trained** Embedding Model | | **1K, 100** | **fake_samples_epoch_999**  Figure 15  **Figure 8.1** | **fake_samples_epoch_100**  Figure 16 | |
| **25 May 2023: Conclusion:**  Requires more caption to train. Data collection is in progress.  **25 May 2023: Future Direction:**  Sharpening Algorithms – Classical  Complete 2500 image captioning | | | | | |
| **01 June 2023: Report:**  **Image pre-processing Update:** Instead of cropping the images now we are re-sizing while maintaining aspect ratio.  **Text encoder update:** Text encoder algirithm updated form **skipgram** to **cbow**  **(**[NLP 101: Word2Vec — Skip-gram and CBOW | by Ria Kulshrestha | Towards Data Science](https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314)**)** | | | | | |
| exp9: sixray\_2500\_ftt\_2048D\_**cbow**\_nocrop\_stage\_1\_test\_2023\_05\_30\_01\_19\_54  **Trained** Embedding Model | | **200,200** | **fake_samples_epoch_199**  Figure 17  **Figure 9.1** | **fake_samples_epoch_199**  Figure 18 | |
| exp10: sixray\_2500\_ftt\_1024D\_**cbow**\_nocrop\_stage\_1\_test\_2023\_05\_30\_16\_47\_57  **Trained** Embedding Model | | **500,300** | **fake_samples_epoch_499**  Figure 19  **Figure 10.1** | **fake_samples_epoch_295**  Figure 20 | |
| **SkipGram vs CBOW Encoder**    Figure 21 | | | | | Figure 22 |
| **Analysis:** Compared to skip-gram we can see using CBOW text encode we have achieved equilibrium much faster. With skip-gram it took 200k iterations while CBOW only took 70K iterations. Given that fact we can also infer form the plot that the discriminator loss decreased much faster for CBOW than skip-gram. Besides the overall loss is lower for CBOW as well. To summarize CBOW(Figure 24) is better than skipgram(Figure 23) so far. | | | | | |
| **SkipGram** with **1k** Epoch | **fake_samples_epoch_100**  Figure 23 | | | | |
| **CBOW** with **200** Epoch | fake_samples_epoch_295  Figure 24 | | | | |
| **01 June 2023: Conclusion:** As we have seen with the change of text encode we have achieved a significant performance boosts while generating images, we are planning for visual-text encoder training in order to generate more realistic images. | | | | | |
| **01 June 2023: Future Direction:**  Priorities   1. **GPT** Based Encoder - “Estimated Effort: small” 2. **Visual -Text Encoder** Training - “Estimated Effort: Big”    1. Paper: [Learning Deep Representations of Fine-grained Visual Descriptions.](http://arxiv.org/abs/1605.05395)    2. Git: [Redcof/CVPR2016\_DA\_SJE\_LanguageModel: Learning Deep Representations of Fine-grained Visual Descriptions (github.com)](https://github.com/Redcof/CVPR2016_DA_SJE_LanguageModel) 3. **Use of pre-trained Visual Text Encoder CLIP** - “Estimated Effort: Big”    1. Paper: [CLIP: Connecting text and images (openai.com)](https://openai.com/research/clip) 4. **Diffusion/GAN** Based Super Resolution. - “Estimated Effort: Not Estimated”   **Data Collection:**  Original Dataset: ~2300 Images  Unique Images Captioned: 500 Captioned  Mappitng: 500 to 2300 images to increase dataset  Current: 500 images 2500 captions  Estimated :~2300 images and 11K captions | | | | | |
|  | |  |  |  | |
|  | |  |  |  | |
|  | |  |  |  | |
|  | |  |  |  | |
|  | |  |  |  | |