Exploring GAN based rich latent feature generation in visual servoing deep models

By Deepankur Kansal
B.Tech,CSE (180226)
Under the supervision of Professor L. Behra
Mentor: Mr. Prem Raj (PhD Scholar, CSE)
Course: CS396A

1 Project Description and Motivation

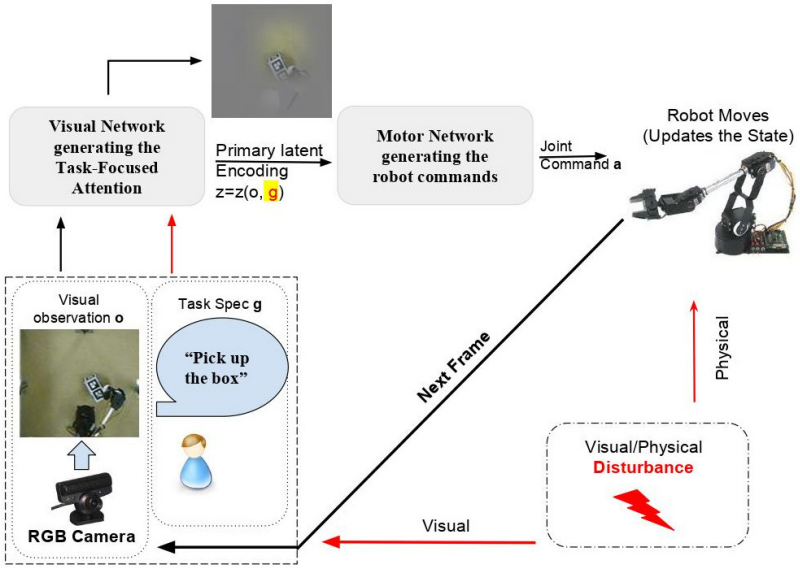


Figure 1: Problem Overview

The project is based on the work done by Abolghasemi *et. al.* in [1]. The idea of the paper comprises the following sub-tasks:

- Describing each input(sequence of images and text) of the task and generate masking of the images.
- Converting the attributed inputs to latent encoding.
- Using latent encoding to generate next state robot angles.
- End-to-end training without human intervention and state tracking.

2 Existing Baseline Implementation

The network components are summarized as follows:

- Variational Autoencoder(VAE):Converts each input image along with encoded text to a Latent Encoding Vector

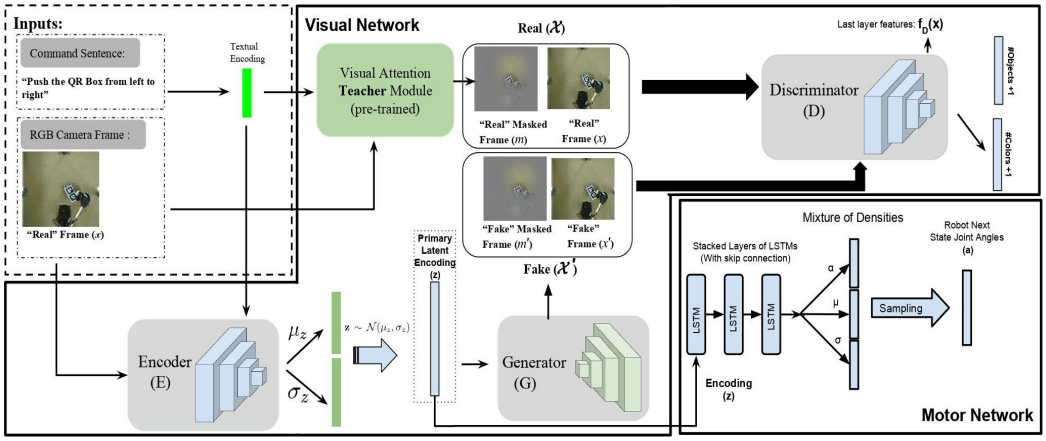


Figure 2: Overall Network Architecture

- GAN Network: Provides a loss function for training the encoding vector and generating masked images.
- Motor Network: Generates next state of joints from encoding vector.

2.1 VAE-GAN Network

The architecture of the model is derived from VAE-GAN which has the following design components:

- Encoder Architecture: Receives a raw frame x and a one =hot representation of the user command denoted by $I_c \in 0, 1^{|M|}$ where M is dictionary space.
  Through a multi-layer convolutional neural network(E), a two dimensional vector is generated .
  We assume that $z \ \ N(\mu_z, \sigma_z)$, and $[\mu_z|\sigma_z] = E(x, I_c)$ where μ_z, $\sigma_z \in R^{d_z}$ and d_z is the length of the Latent Encoding vector(z).
- Decoder Architecture: Implemented by a GAN model acting with its Generator part acting as a Decoder generating two images i.e. the reconstructed frame both with and without the masking.
  The Discriminator along with generating the loss function also acts as a classification model for the problem signifying the color of the object and the number of the objects in the frame.

2.2 Motor Network

- It takes as input the primary latent encoding, z, which is processed through a 3-layer LSTM network with skip connections. The memory cells of LSTMs get updated through the time by doing the task (frame by frame).
- The output of the final LSTM layer is fed into a mixture density network (MDN).
- MDN provides a set of Gaussian kernels parameters namely μ_i , σ_i and the mixing probabilities $\alpha_i(x)$, all $\in R^{|J|}$, and $1 \leq i \leq N_G$.
  Here, $|J|$ is the number of robot joints (specific to the robot) and N_G is the number of Gaussian components.
- The $|J|$-dimensional vector describing the next joint angles is sampled from this mixture of Gaussians.

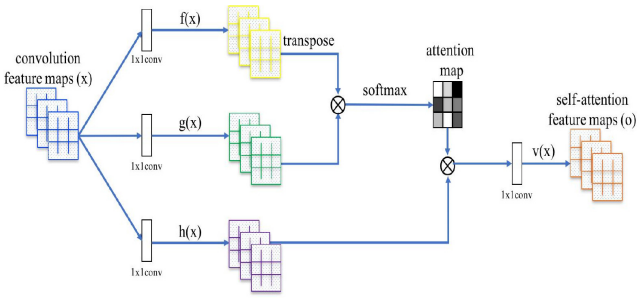


Figure 2. The proposed self-attention module for the SAGAN. The $\otimes$ denotes matrix multiplication. The softmax operation is performed on each row.

- Originally, self-attention to the GAN framework was introduced to enable both the generator and the discriminator to efficiently model relationships between widely separated spatial regions which we adapt as intuitively we are trying to find distinctions among objects as given by the textual input.
- The attention map are generated by taking column wise softmax over the transposition of row wise convolutions:
  $\beta_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{N} exp(s_{ij})}$ where $s_{ij} = f(x_i)^T g(x_j)$
  Here, N is number of feature maps from the previous hidden layer.

3 Proposed Improvements

3.1 GAN Architecture

- Originally, the Wasserstein GAN-GP [2] is employed, it maximizes the Wasserstein-I or Earth-mover distance, defined as

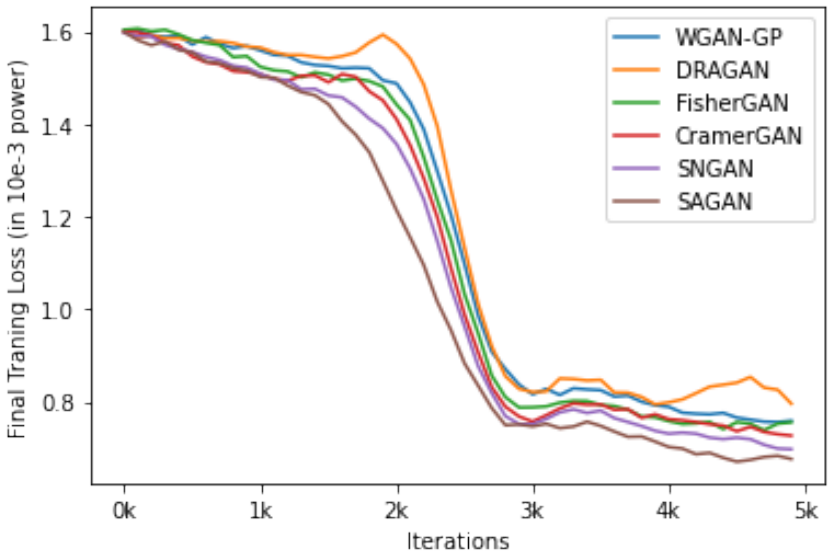
$$\max_{D} \mathbb{E}_{y\sim\mu}[D(y)] - \mathbb{E}_{x\sim\nu}[D(x)], \tag{1}$$

  where D: discriminator, ν/μ: distributions of fake/real samples.
- It faces issues with training non-convergence and slow update to loss metric.
- For our problem statement the convergence of GAN's should require minimum number of iterations as possible while also being robust.
- We experimented with different architectures of discriminators independently as followed:
  - DRAGAN [3] (Different Loss function)
  - FisherGAN [4] (Different Loss function)
  - cGAN(CramerGAN) [5] (Utilizes class labels)
  - SNGAN [6] (Utilizes class labels and Different Loss function)
  - SAGAN [7]'(best performance till experimentation)(Attention Driven with Feature Maps)

3.2 Self-Attention Based Model

3.3 Improvements to Motor Network

- The variant of LSTM used in the original work is LSTM with peephole connections which reduces the problem of vanishing gradient.
- I propose to implement BiLSTM with attention variant which converges faster and more accurately than traditional LSTMs. The attention mapping is done among the LSTM layers and monitor the overall state of the environment which would also be helpful in sampling.
- The motor network does not provide a trainable sampling module to sample from MDM distributions neither it learns from the past experience of disturbances(removal of object from its position during experimentation).



Comparison of different GANs using final training loss for a fixed number of iterations(5k) under the purview of fastest convergence.

3.4 Improvements to Sampling Module

- I propose to implement a policy-based Reinforcement Learning Module over the sampling algorithm which would in cases of disturbances will implement low probability events stochastically to try to come out or retrace the steps that have been taken by the motor network.
- The critic of the module is reinforced by the current state (S_i) and previous state of the (S_{i-1}) along with momentum terms $M(S_{i-2}...S_0)$ defined by the latent encoded vector.
- In case of disturbances, the latent encoded vector is intuitively expected to deviate drastically from S_{i-1}.
- The L2-norm difference of encoded vector is compared with a threshold depending on state, momentum and weight parameters which if exceeded inverses(log-inverse) the probabilites related to sampling of the joint angles, generating hyperlocalized changes stochastically.
- With the change we expect that the model will return to a stable state after repearted iteration of sampling.

4 Simulation Results

4.1 GAN Based improvements

From Fig. **??**, we can see that self-attention based architecture performs and converges faster on the dataset while training.

4.2 Overall Results

Note: In cases of tests with RL Sampling Module and with BiLSTM variant,some cases of success demonstration does not overlap therefore results might improve if stacked together. The paper was implemented from scratch and all the experimental results and proposals have been tested for one camera angle dataset of the original paper for the pickup task.

Tasks	In Benign Conditions	With Disturbances	With RL module	With BiLSTM variant
Red Bowl	75	70	75	75
White Towel	75	70	75	80
Blue Ball	60	60	65	65
Yellow Plate	50	50	65	55
Black Dumbell	80	75	80	80
Red	40	40	45	45
Mean	63.4	61.7	65	66.7

Table 1: Percentage of successful tasks while testing on the data set.

5 Summary of the work done

- Implemented the original paper from scratch and reproduced adequate results on the dataset provided.
- Tested the model architecture on a artificially created dataset generated by the research team.
- Tested the VAE encoding performance on multiple GAN architectures and selected SAGAN which makes learning faster due to the attention maps architecture.
- Tested the BiLSTM with attention variant to the motor network generating better results.
- Tested the RL sampling module which in cases of disturbances generated better results but more training and testing is required.

6 Possible Future Work

More rigorous testing can be done with the sapling module as well as the self-attention architecture. Transformers, a new category of models can also be explored and assimilated in the model.

Acknowledgments

I thank my supervisor, Prof. L. Behera and Project Mentor Mr. Prem Raj, for trusting me with this project and giving me the freedom to explore a topic . Their guidance and support enabled me to focus on the project and complete my work in time.

References

[1] Pooya Abolghasemi, Amir Mazaheri, Mubarak Shah, and Ladislau Bölöni. Pay attention! - robustifying a deep visuomotor policy through task-focused attention. *CoRR*, abs/1809.10093, 2018.

[2] Mart Arjovsky Vincent Dumoulin Aaron C. Courville Ishaan Gulrajani, Faruk Ahmed. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[3] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[4] Youssef Mroueh and Tom Sercu. Fisher GAN. *CoRR*, abs/1705.09675, 2017.

[5] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *CoRR*, abs/1705.10743, 2017.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018.

[7] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.