Google Summer of Code 2019 Proposal

Enriching Model Zoo with Deep Learning Models

Student : Manjunath Bhat

Mentors : Dhairya Gandhi, Elliot Saba

About Me

Contact Information

Name :	Manjunath Bhat
University :	Indian Institute of Technology (IIT), Kharagpur
Email :	manjunathbhat9920@gmail.com
Github Handle :	thebhatman
Blog :	https://medium.com/@manjunathbhat9920
Phone :	(+91) 7384434093
Timezone :	Indian Standard Time (UTC +05:30)

Platform Details

OS	Ubuntu 16.04
Editor	Sublime Text
Version Control	Git

Personal Background

I am Manjunath Bhat, a second-year undergraduate student from the Department of Mechanical Engineering at Indian Institute of Technology (IIT), Kharagpur.

I find programming and mathematics very interesting as they are one of the best means to exercise the brain. I am proficient in C, C++, Python, MATLAB and Julia Language.

My fields of interest are Machine Learning, Deep Learning, Computer Vision, Path Planning, and Robotics. I was introduced to the field of Computer Vision, Machine Learning and Robotics in my first year at IIT Kharagpur, when I joined the Kharagpur RoboSoccer Students' Group(KRSSG) and since then I have been exploring these fields. I am an active member and contributor of the Artificial Intelligence Team of KRSSG, where we participate in an International Competition called RoboCup SSL (Small Sized League). The codebase of our team is based upon the Finite State Machine(FSM) architecture.

As a part of KRSSG, I have worked on robot path-planning algorithms such as RRT (Rapidly Exploring Random Trees), RRT-Star, RRT-Connect, and RRT-Star with Artificial Potential Field (https://github.com/thebhatman/RRT_Star_With_APF) . Currently we are working on using Deep Learning in path planning to predict the ideal value of Potential in different obstacle fields. This value of potential is used as a heuristic in RRT-Star with Artificial Potential Field (https://github.com/thebhatman/ML_RRT-_APF). I have been acknowledged for my work on RRT Star with APF in the paper Potential and Sampling Based RRT Star for Real-Time Dynamic Motion Planning Accounting for Momentum in Cost Function: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part VII.

After having found interest in ML and DL by completing Andrew NG's ML course, I wanted to learn more in the field of Deep learning and its applications in Computer Vision. I took up the courses CS230n and CS231n offered by Stanford University. I further explored the algorithms and intricacies of Deep learning by reading blogs and research papers. I am proficient at using Tensorflow, Keras, and scikit learn.

Experience with Julia

I was introduced to Julia about three months ago, and got comfortable with coding in Julia very quickly. I like Julia because of its performance, straight-forward syntax, and its unique features in Metaprogramming. I am comfortable with the Flux API, and have gone through its backend and working. I have worked on Gated Recurrent Convolutional Neural Networks (GRCNN) using Flux. It is currently a work in progress. https://github.com/thebhatman/model-zoo/blob/grcnn/vision/cifar10/GRCNN.jl

Past Contributions

I have been contributing to Flux.jl and model-zoo since February, by solving issues, and sending PRs to add new features. Here are a few of my contributions:

1. Implemented AlphaDropout (Merged)

   I contributed to the Flux library by adding AlphaDropout, which is used in self-normalizing neural networks to preserve mean and variance of input, in the output.

2. Fixed issue of DepthwiseConv for Float64 input

   DepthwiseConv was breaking for float64 type input. I fixed this issue through this pull-request.

3. Added new loss functions

   I have added the KLDivergence loss function, Poisson loss, logcosh loss and Hinge loss in this pull-request.

4. Added Adaptive Pooling layers

Through this patch, I have added adaptive max pooling and adaptive mean pooling layers.

5. Fixed breakage of conv for unequal kernel width and height (Merged)

   This PR solves the error that was present in output size calculation in conv layer of NNlib.jl.

6. Solved problem of train!() with TrackedReal

7. Opened an issue regarding flip() calling reverse() without dims argument

The Project

Enriching Model Zoo with variants of GANs, Spatial Transformer networks and deep learning models

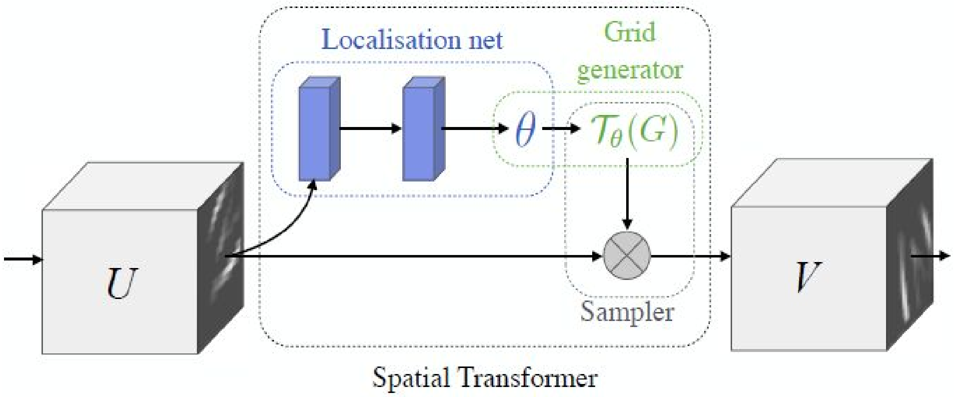
Currently, Model zoo needs to be enriched with unsupervised deep learning models, in particular the variants of Generative Adversarial Networks(GANs). I propose to add the following models to the model zoo:

- Spatial Transformer Network
- VAE-GAN
- StarGAN for Facial expression synthesis
- Energy Based GAN
- Gated Recurrent Convolutional Neural Network for OCR

Approach and implementation

- **Spatial Transformer Network** :

  Spatial Transformer Networks (STN) remove spatial invariance from images by applying a learnable transformation followed by interpolation. The problem with Convolutional Neural Networks is its lack of robustness towards spatial invariance in images. CNNs learn invariance using pooling layers and convolution layers, which are not very effective, as they have a small receptive field. Pooling layers are unable to provide rotational and scaling invariance. STN blocks can be placed in a CNN and it works almost independently. STN is a mechanism that rotates or scales an input image in order to focus on the target object and to remove rotational variance.



There are mainly three types of transformations learnt by STN : Affine Transformation, projective transformation and thin plate spline transformation.

Spatial Transformer Networks are mainly composed of three parts : the localisation network, grid generator and sampler.

Localisation net takes the original image as input and outputs the parameters of the affine transformation that must be applied to the input image. The grid generator generates a grid of (x,y) coordinates that correspond to a set of points where the input should be sampled to produce transformed output. Bilinear Sampler takes as input the original image and the grid produces the output image using bilinear interpolation.

[Paper Reference]

- **StarGAN**

StarGan incorporates multiple datasets containing different label sets to perform image translations using any of these labels. The goal is to train a single generator G that learns mappings across different domains. G is trained to translate an input image x into an output image y conditioned on the target domain label c. The target domain label c is generated randomly so that G learns to flexibly translate the input image across all domain labels. I will be using the CelebA datasets and RaFD datsets. The input image will be from the CelebA dataset and the facial expression features(‘happy’, ‘sad’, ‘angry’, etc.) will be learned from the RaFD dataset. The Discriminator produces distributions over both sources and domain labels. The objective loss function to be minimized can be split into three separate losses:

1) **Adversarial Loss:** This is to make the generated images as realistic as possible.

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x,c)))]$$

Where, G generates a fake image given an input image x, and target domain label c, while D tries to distinguish between real images from the dataset and fake images generated by G.

2) **Domain Classification Loss:** This loss has two terms. One loss term calculates classification loss of real images from source dataset, and the other loss term is the classification loss of fake images over the target domain labels. c’ represents labels in source dataset and c represents labels in the target domain.

$$\mathcal{L}^r_{cls} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)] \quad \text{(real images)}$$

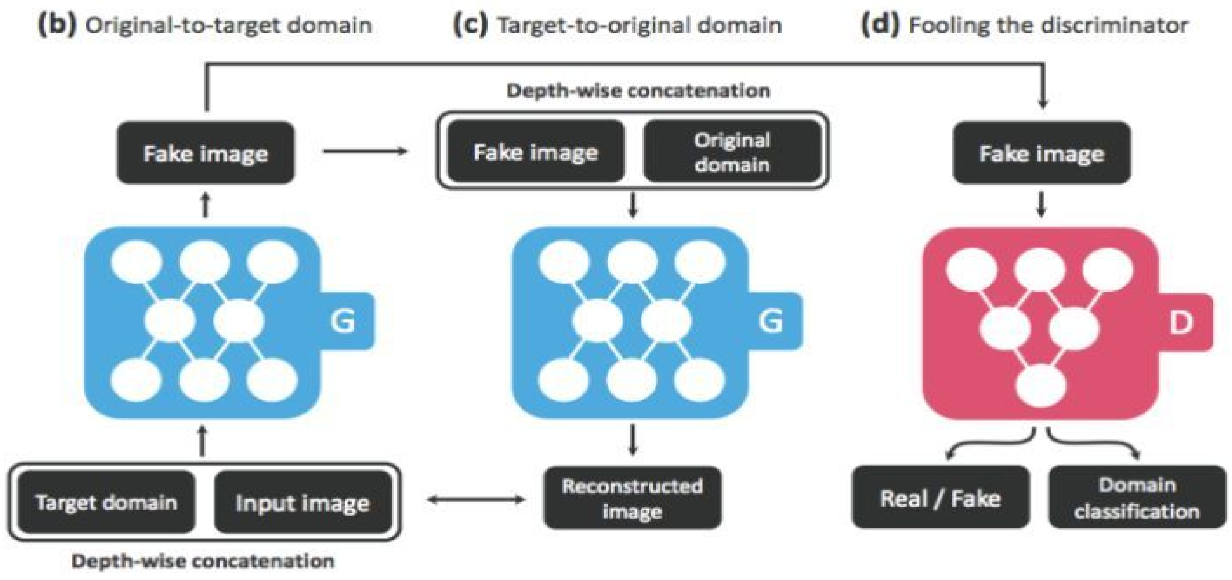
$$\mathcal{L}^f_{cls} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))] \quad \text{(fake images)}$$

3) **Reconstruction Loss:** Minimizing this loss preserves the content of input image in the generated output image.

$$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1]$$

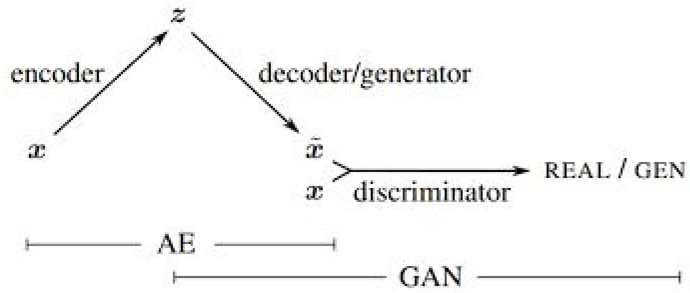
Link to paper.

The figure below represents the training process of a StarGAN.



- **VAE-GAN :**

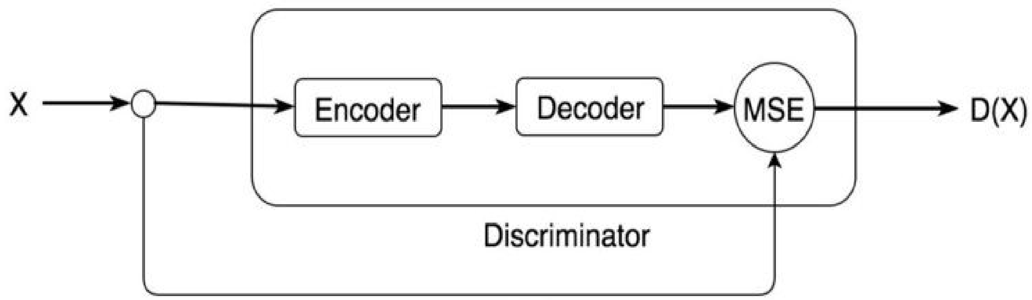
As the name suggests, this model is a combination of two types of generative models: Variational Autoencoders(VAE) and Generative Adversarial Networks(GANs). In this model, the input image is fed to an encoder network, which results in a lower dimensional latent output. This latent output vector is fed to the decoder of the VAE, which basically functions as a generator and tries to reconstruct the input image. The Discriminator of GAN is used to classify whether the reconstructed image is real or fake. I will be using this [paper](#) for



reference.

- **EBGAN** :

  Energy based GAN uses an autoencoder as its discriminator. The Discriminator is trained on the real images in the dataset to reduce the mean square error as shown in figure below..



The Discriminator of EBGAN outputs the reconstruction error instead of probability like in ordinary GANs. The objective loss function consists of two goals: a good autoencoder and a good discriminator.

$$\mathcal{L}_D(x, z) = D(x) + [m - D(G(z))]^+$$
$$\mathcal{L}_G(z) = D(G(z))$$

$$\text{where } [\,\mathrm{u}\,]^+ = max(0, \mathrm{u}\,)$$

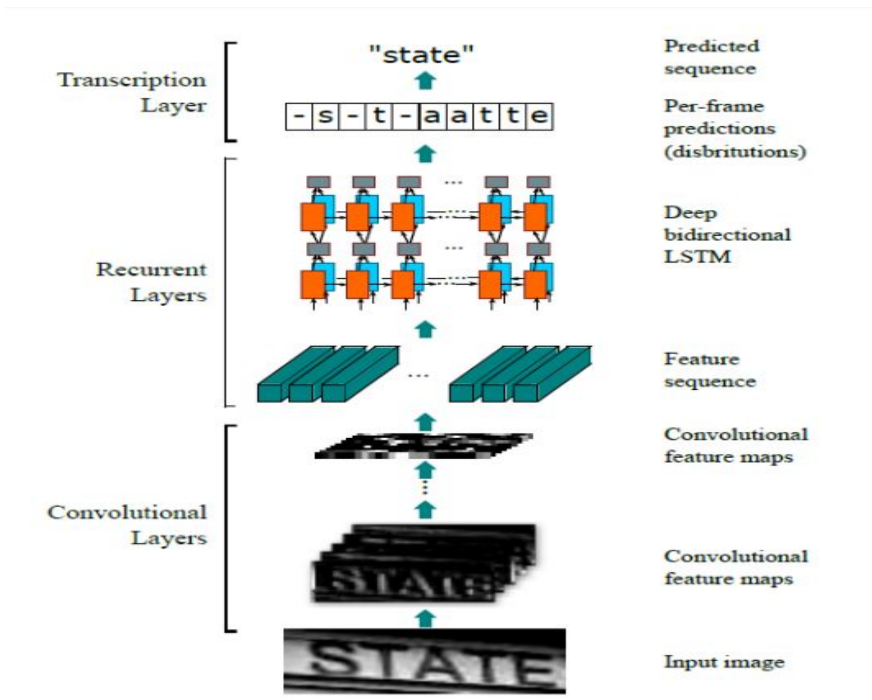
Minimizing the second loss function ensures that the generated images are realistic and close to the images in the training dataset.

Minimizing the first loss function ensures that the discriminator is properly able to distinguish between real and fake images. Since discriminator outputs the reconstruction error, the discriminator is penalised if the error drops below a particular value m.

This [paper]() is used as reference for idea and implementation.

- **Gated Recurrent Convolutional Network for OCR** :

This model aims to recognise text from images. The fundamental layers of this model are Gated Recurrent Convolutional layers (GRCL), CNNs and BiDirectional LSTM. The approach and implementation is discussed in this paper. The convolution networks generate a feature map, which is split into several feature sequences and provided as input to the recurrent layers. The recurrent layers predict the character in each frame. The model architecture is as shown.



Timeline

This Timeline is tentative and lists the objectives I aim to complete within the timeframe specified. There could be situations in which the project could get delayed or reach early completion. To handle such cases, I have made sure to have buffer weeks to complete the incomplete tasks. I am very enthusiastic about working with Flux, model zoo and the Julia Community in general, and would be contributing by taking up issues outside of my GSoc project as well.

- **Pre-GSoC Phase 0 (upto May 6):**

  I plan to spend my time understanding the working of Flux, NNlib, Tracker, and Zygote at a deeper level, so that I can make the best use of Flux API while implementing the proposed models, at the same time taking up issues and contributing to the community. My end semester exams at college will be held during the last ten days of April, and I will be inactive during this period.

- **Community Bonding Period (May 6 to May 27):**

  During this period, I will discuss implementation ideas with my mentors, and form a solid roadmap for the project. I also plan to setup the requirements of GPU, and datasets, with the help of my mentors. I will spend my time reading research papers and blogs related to the proposed models, and understand them to the core. I will also get myself accustomed to the coding style.

- **Week 1 (Coding Period begins):** I will get started with the implementation of Spatial Transformer Networks.
- **Week 2:** I plan to complete the implementation of Spatial Transformer Network. Debugging and tuning hyperparameters for training will be my next task.
- **Week 3:** Documentation, writing tutorial and updating blog for Spatial Transformer Network. I also plan to get started with the implementation of VAE-GAN.
- **Week 4:** Complete the implementation of VAE-GAN, debugging and training.

- **Week 5:** Buffer week. I plan to complete backlogs (if any). Else I will proceed with the documentation of VAE-GAN.
- **Week 6:** I plan to start the implementation of StarGAN.
- **Week 7:** Implementation of StarGAN and debugging.
- **Week 8:** Tuning hyperparameters and training the model. Start documentation of StarGAN.
- **Week 9:** Implementation of Energy Based GAN.
- **Week 10:** Debugging and training of EBGAN. Complete Documentation.
- **Week 11:** Implementation of GRCNN.
- **Week 12:** Debugging, Training, and documentation.

Answers to listed questions (as listed in application guidelines)

1. **What do you want to have completed by the end of the program?**

   By the end of the program, my aim is to enrich model-zoo with new and interesting deep learning models, variants of GANs, and their applications. Model zoo will also have the Spatial Transformer Network, which enhances CNNs by learning spatial invariance in images.

2. **Who's interested in your work, and how will it benefit them?**

   In general, anybody interested in the field of Deep Learning, and wants to use Julia to learn and implement deep learning models and their wide range of applications, will be benefitted by this project. Flux is a powerful framework for Machine Learning and Deep Learning, and model-zoo will show how to make the best use of the powers and features offered by Flux. It will serve as a great resource for deep learning enthusiasts, as it contains techniques, ideas, and models from the latest research papers.

3. **What are the potential hurdles you might encounter, and how can you resolve them?**

A potential hurdle for me would be to train the implemented deep learning models and GANs. I do not have access to a GPU server other than Google Colab, and since training DL models takes a lot of time, I would require support for GPU servers. Also training the GANs to converge will prove to be a difficulty. This can be solved by setting hyperparameters similar to the ones used in pretrained models of tensorflow or pytorch.

I currently do not have access to the RaFD dataset as they only provide access to staff members of accredited universities.

4. How will you prioritize different aspects of the project like features, API usability, documentation, and robustness?

Please refer to the “Timeline” section where I have described in detail about the sequence of carrying out the project.

5. What milestones can you target throughout the period?

At the end of week 4, I expect to complete the implementation of Spatial Transformer Networks and VAE-GAN. At the end of week 12, I expect to complete all the models listed in my proposal.

6. Are there any stretch goals you can make if the main project goes smoothly?

Yes. I am also interested in implementing an object detection algorithm and add it to model zoo. It would be great if Flux had a region proposal network which is used very frequently in many object detection algorithms. Since I have already gone through the backend of Flux, and understand its working, I would also love to work on the integration of Zygote with Flux, and understand the techniques of Automatic Differentiation. Apart from this, I will actively contribute to the backend of Flux and NNlib by taking up issues.

7. What other time commitments, such as summer courses, other jobs, planned vacations, etc., will you have over the summer?

I expect to work full time on this project, that is 40 or more hours a week. My university vacations align with the timeline of GSoC. Even though my college reopens on July 17th, I would still be able to devote around 40 hours a week, as I'd still not have any tests to study for, till September. In case of me being unavailable during the GSoC period due to reasons such as travelling, or any other unavoidable circumstances, I will inform my mentors beforehand about it and make sure I make up for lost time.

I have been part of the Julia Community for the past three months, and I have enjoyed the regular discussions in the community. I am really excited to be part of the Community, and contribute to JuliaLang. I have learnt a lot through discussions, solving issues, and asking questions. I am looking forward to an amazing learning opportunity.

THANK YOU!