e-YSIP 2022

Automated text-translation and data visualization using Generative Adversarial Networks (GANs)

Intern 1 Name: Samya Bose
Intern 2 Name: Rahul Raghav A
Mentor Name: Aditya Panwar, Andrea Furtado, Kalind Karia
Duration of Internship: 06/06/2022 – 23/07/2022

2022, e-Yantra Publication

Project Name

Abstract

This project mainly revolves around data visualization techniques involving various state-of-the-art machine learning models and Generative Adversarial Networks (GANs). Charts and graphs are an effective way to convey information but today's modern algorithms cannot effectively parse them. These existing methods fail when faced with minor changes or variations in the appearance. Here we worked with a specialized dataset DVQA which involves processing questions and answers that are unique to a particular image (chart). In this project, we have developed our own model loosely based on the proposed models by the authors of DVQA and performed a performance analysis of it, with the current market dominating model. We have also highlighted a few points marking the limitations of both the approaches of solving the DVQA problem. We have also worked with a Generative Adversarial Network capable of developing a texturized, meaningful image when faced with a rough sketch or facade of some sort. The model used for this was extensively trained on a huge dataset, which allows the user to potentially draw shapes which barely resemble the desired object and still receive a meaningful depiction. We have also developed an interactive web interface for the model, such that its novelty can be experiences by everyone without much of a hassle.

Completion status

The project and all of its aspects have been completely covered and successfully achieved.

1.1 Software used

- CUDA
- Detail of software: 11.3
- Installation steps -

```
conda install pytorch torchvision cudatoolkit=11.3 -c pytorch
```

- PyTorch
- Detail of software: 1.6.0
- Installation steps -

```
conda install pytorch torchvision cudatoolkit=11.3 -c pytorch
```

1.2 Software and Code

Github link for the repository of code.

All the codes are segmented into well documented Colab notebooks and links, with proper documentation for every step,

- SAN-VQA Demonstration - Utilises pre-trained SAN-VQA weights to assess its performance on DVQA dataset.
- SAN-VQA on VQA - Trains a SAN-VQA on a subset of VQA dataset to assess its performance measure.
- SAN-VQA on DVQA - Trains a SAN-VQA on the entire dataset of DVQA to assess its performance measure.
- Pre-processing of DVQA - Pre-processes the entire dataset of DVQA and generates all possible question-answer pairs that would be needed in the next steps.
- Binary Classification on DVQA - Trains a binary classifier on DVQA dataset capable of classifying the questions and predicting whether OCR would be required to answer it.

- Regression Model on Attended Features - Trains a regression model which accepts the output of SAN-VQA and predicts a bounding box for the answer. It also contains the OCR feature which reads the answer from the bounding box.
- Performance Analysis of PReFIL - Performs a thorough analysis of PReFIL's performance over DVQA dataset which highlights its accuracy and supremacy over previously proposed models in all categories of questions.
- Demonstration of pix2pix - Trains a pix2pix model on a very small dataset and highlights its performance.
- Detailed Study of pix2pix - Dives deeper into the architecture of pix2pix while training and converting a model into its tensorflow.js counterpart.

1.3 Use and Demo

Final Setup Image User Instruction for demonstration -

- For DVQA, user can use the Google Colab notebooks.
- For pix2pix, user can also use the Google Colab notebooks, but for the interactive web page simply execute the serve.py file with python.

```
python3 serve.py --port 8000
```

Drive Link of demonstration video

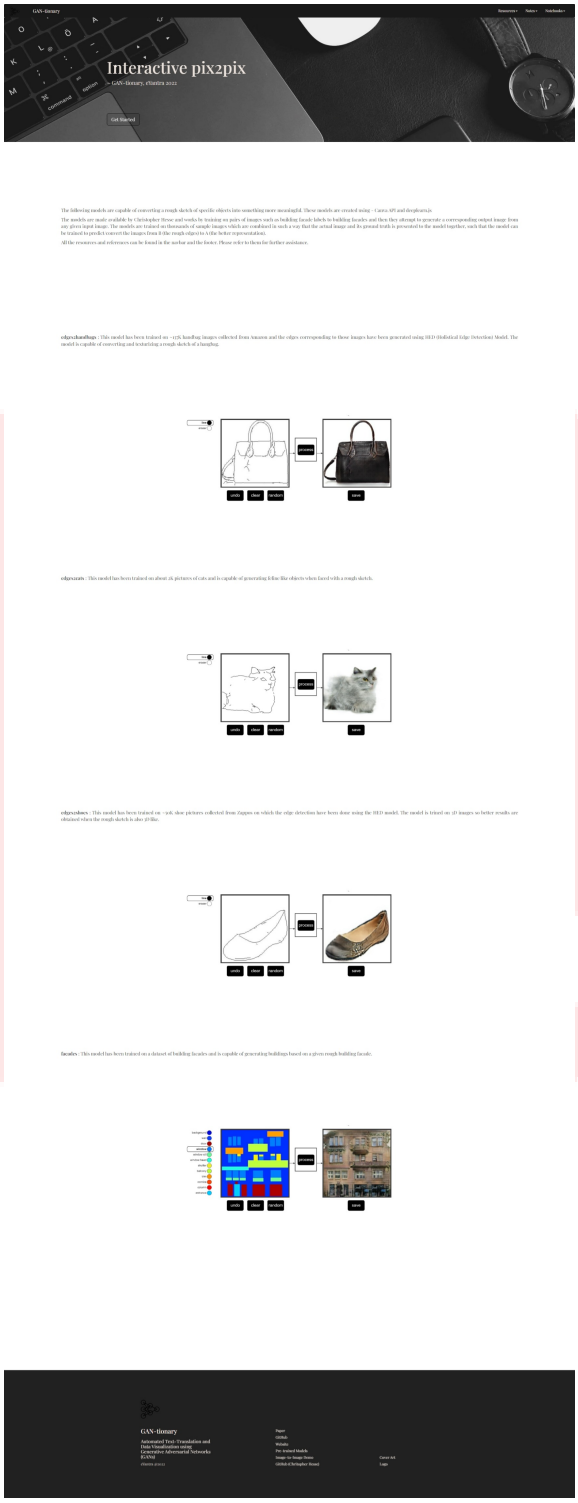

Figure 1.1: Performance Measure

1.4 Future Work

- The current approaches to DVQA problem involves processing questions and answers specific to their corresponding images, as a result it compromises generalization of the whole problem. Developing a more generalized approach to the problem can be extremely useful when thinking of providing it as a service.
- The current model used for creating the datasets for pix2pix involves holistically detecting the edges from the pictures. But the current model is under some construction so finding an alternative or developing one, can help create more datasets for pix2pix.

1.5 Bug report and Challenges

Any issues in code and hardware -

- Some of the functions are deprecated so lowering the versions of the technology stack in Google Colab will help resolve the user warnings.
- Update Pillow to its latest installment every time when using Jupyter notebooks to resolve an issue with displaying and processing images in Jupyter.

Any failure or challenges faced during project -

- The current approaches to DVQA problem involves processing questions and answers specific to their corresponding images, as a result it compromises generalization of the whole problem. So it can only predict relevant answers for questions and images which are constructed from elements it has already encountered that is, its vocabulary.
- The current model used for creating the datasets for pix2pix involves holistically detecting the edges from the pictures. But the current model is under some construction so finding an alternative or developing one, can help create more datasets for pix2pix.

Bibliography

[1] Zichao Yang, Xiadong He, Jianfeng Gao, Li Deng, Alex Smola (2015), Stacked Attention Networks for Image Question Answering.

[2] Kushal Kafle, Brian Price, Scott Cohen, Christopher Kanan (2018), DVQA: Understanding Data Visualizations via Question Answering.

[3] Kushal Kafle, Robik Shrestha, Brian Price, Scott Cohen, Christopher Kanan (2019), Answering Questions about Data Visualizations using Efficient Bimodal Fusion.

[4] Vahid Kazemi, Ali Elqursh (2017), Show, Ask, Attend, and Answer: A Strong Baseline For Visual Question Answering.

[5] Kushal Kafle, Robik Shrestha, Brian Price, Scott Cohen, Christopher Kanan (2019) - PReFIL.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros (2016), Image-to-Image Translation with Conditional Adversarial Networks.

[7] Phillipi - PyTorch pix2pix.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros (2016) - Image-to-Image Translation with Conditional Adversarial Networks.

[9] Christopher Hesse - Pre-trained pix2pix models.

[10] Christopher Hesse - Tensorflow pix2pix.