STAT 663 (Spring 2024)

Generating Yu-Gi-Oh! Monsters

Eric Rios Soderman & Jiwon Shin

Motivation

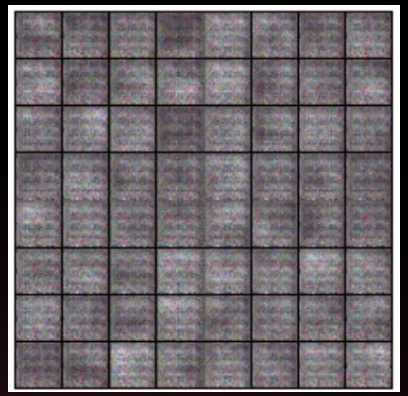
Why Yu-Gi-Oh! Cards?

++++

- Yu-Gi-Oh! cards offer a diverse array of vibrant color palettes and unique archetypes, contrasting with the more consistent designs of Pokémon cards. This visual variety provides an excellent dataset for testing and enhancing image generation techniques for datasets with inconsistent imagery.

Motivation

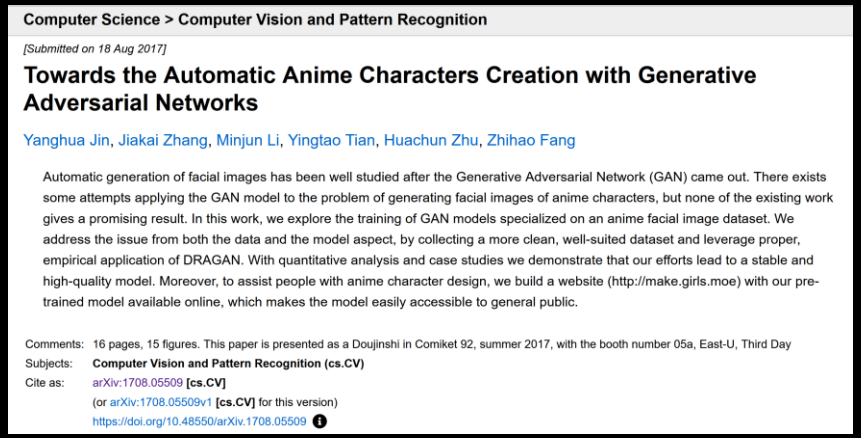
Why Generate Them Using GANs?



Generative Adversarial Networks (GANs) have proven effective in creating detailed and diverse images but still have potential for improvement, especially in managing complex visual data like Yu-Gi-Oh! cards.

Our project aims to advance work in this particular domain, by using GANs to explore the capabilities in generating intricate and varied visual elements, which could benefit both the AI field and the gaming industry.

Excellent tools, but our work focuses on archetype (data) selection



Computer Science > Computer Vision and Pattern Recognition

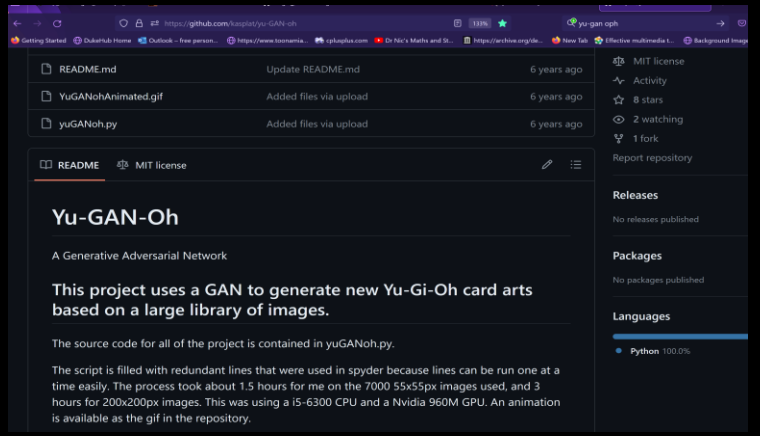
[Submitted on 18 Aug 2017]

Towards the Automatic Anime Characters Creation with Generative Adversarial Networks

Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, Zhihao Fang

Automatic generation of facial images has been well studied after the Generative Adversarial Network (GAN) came out. There exists some attempts applying the GAN model to the problem of generating facial images of anime characters, but none of the existing work gives a promising result. In this work, we explore the training of GAN models specialized on an anime facial image dataset. We address the issue from both the data and the model aspect, by collecting a more clean, well-suited dataset and leverage proper, empirical application of DRAGAN. With quantitative analysis and case studies we demonstrate that our efforts lead to a stable and high-quality model. Moreover, to assist people with anime character design, we build a website (http://make.girls.moe) with our pre-trained model available online, which makes the model easily accessible to general public.

Comments: 16 pages, 15 figures. This paper is presented as a Doujinshi in Comiket 92, summer 2017, with the booth number 05a, East-U, Third Day
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: arXiv:1708.05509 **[cs.CV]**
(or arXiv:1708.05509v1 **[cs.CV]** for this version)
https://doi.org/10.48550/arXiv.1708.05509



README.md — Update README.md — 6 years ago
YuGANohAnimated.gif — Added files via upload — 6 years ago
yuGANoh.py — Added files via upload — 6 years ago

README | MIT license

Yu-GAN-Oh

A Generative Adversarial Network

This project uses a GAN to generate new Yu-Gi-Oh card arts based on a large library of images.

The source code for all of the project is contained in yuGANoh.py.

The script is filled with redundant lines that were used in spyder because lines can be run one at a time easily. The process took about 1.5 hours for me on the 7000 55x55px images used, and 3 hours for 200x200px images. This was using a i5-6300 CPU and a Nvidia 960M GPU. An animation is available as the gif in the repository.

MIT license
Activity
8 stars
2 watching
1 fork
Report repository

Releases
No releases published

Packages
No packages published

Languages
Python 100.0%

Our Hypothesis : Archetypes

Our Hypothesis: There will be better fake image generation with thematically similar cards. ++++

Research Questions

1. Enhancing Yu-Gi-Oh! Card Image Generation through GAN Model Optimization

- How can we refine GAN model parameters to amplify image generation quality for Yu-Gi-Oh! cards by strategically manipulating the dataset?

2. Archetypal Influence on Generation Quality

- To what extent does incorporating archetype control mechanisms into the training dataset enhance the fidelity of generated images and textual content?

3. Augmented Precision via Focused Data Categorization

- Can the precise categorization of card attributes, particularly by archetype, in both image and textual datasets empower existing GAN and NLP models to yield more precise and comprehensive outputs?

Data

We sourced both images and detailed information, including image links and specific card details, for 10,763 English Yu-Gi-Oh! fansite cards using their API.

(1) Card_image_scraper.py

```python
def main():
    """
    Downloads images of Yu-Gi-Oh! cards with English text.
    Fetches card data from the API, prepares image links, downloads images, and stores them.
    """
    # Link to the API for around 11,000 cards
    url = "https://db.ygoprodeck.com/api/v7/cardinfo.php"
    request = requests.get(url)
    data_request = request.content

```

(1) Card_info_extractor.py

```python
def prepare_json(data_request):
    """
    Prepares the list of IDs and image links for cards with English text.
    Args:
        data_request (str): JSON data containing card information.
    Returns:
        list: A list of dictionaries containing card information.
    """
    data = json.loads(data_request)['data']
    card_info = []
    for card in data:
        if 'en' in card['desc']:
            card_dict = {
                "id": card['id'],
                "name": card['name'],
                "type": card['type'],
                "desc": card['desc'],
                "atk": card.get('atk', 0),  # Check if 'atk' exists, if not, set to 0
                "def": card.get('def', 0),  # Check if 'def' exists, if not, set to 0
                "level": card.get('level', 0),  # Check if 'level' exists, if not, set to 0
                "race": card['race'],
                "attribute": card.get('attribute', 'Unknown')  # Check if 'attribute' exists, if not, set to 'Unknown'
            }
            card_info.append(card_dict)
    return card_info
```

Fair Use

However, we only kept half for our work in order to comply with fair use.

(1) Card_image_scraper.py

```python
def main():
    """
    Downloads images of Yu-Gi-Oh! cards with English text.
    Fetches card data from the API, prepares image links, downloads images, and stores them.
    """
    # Link to the API for around 11,000 cards
    url = "https://db.ygoprodeck.com/api/v7/cardinfo.php"
    request = requests.get(url)
    data_request = request.content

```



(2) Card_info_extractor.py

```python
def prepare_json(data_request):
    """
    Prepares the list of IDs and image links for cards with English text.
    Args:
        data_request (str): JSON data containing card information.
    Returns:
        list: A list of dictionaries containing card information.
    """
    data = json.loads(data_request)['data']
    card_info = []
    for card in data:
        if 'en' in card['desc']:
            card_dict = {
                "id": card['id'],
                "name": card['name'],
                "type": card['type'],
                "desc": card['desc'],
                "atk": card.get('atk', 0),  # Check if 'atk' exists, if not, set to 0
                "def": card.get('def', 0),  # Check if 'def' exists, if not, set to 0
                "level": card.get('level', 0),  # Check if 'level' exists, if not, set to 0
                "race": card['race'],
                "attribute": card.get('attribute', 'Unknown')  # Check if 'attribute' exists, if not, set to 'Unknown'
            }
            card_info.append(card_dict)
    return card_info
```

Data

Leveraging Hugging Face for Data Upload and Loading

We uploaded our image and text data to Hugging Face and utilized them to train our models using the 'load_dataset' function.

```python
from datasets import load_dataset, DatasetDict

# Load dataset
ds = load_dataset("Jiwonny29/all_data")

# Organize into a single DatasetDict
ds = DatasetDict({
    'train': ds['train'],
    'test': ds['test']
})
```

```python
ds['train'][0]
```

```
{'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=624x624>,
 'id': 10000000,
 'name': 'Obelisk the Tormentor',
 'type': 'Effect Monster',
 'desc': "Requires 3 Tributes to Normal Summon (cannot be Normal Set). This card's Normal Summon cannot be negated.
cannot be activated. Neither player can target this card with card effects. Once per turn, during the End Phase, if
to the GY. You can Tribute 2 monsters; destroy all monsters your opponent controls. This card cannot declare an atta
 'atk': 4000,
 'def': 4000,
 'level': 10,
 'race': 'Divine-Beast',
 'attribute': 'DIVINE',
 'archetype': 'Egyptian God',
 'image_url': 'https://images.ygoprodeck.com/images/cards_cropped/10000000.jpg',
 'image_path': 'training_data_final/training_images/10000000.jpg',
 'simplified_type': 'effect monster',
 'image_id': '10000000.jpg'}
```

Works and Tools Known in this space

DALL-E

Stable Diffusion

MidJourney

But also tools powered by them

Excellent tools, but our work focuses on archetype (data) selection

Image Generation - Evaluation Metric

Our Metric : Frechet Inception Distance (FID)

FID is preferred for us over IS because it offers greater robustness in evaluating our synthetic images based on statistical measures, and it also indicates the difference in distribution between the actual and generated images.

Disregarded Metric : Inception Score

- Uses a model that was trained on different set of classes than ours
- Fictional characters aren't accounted for in said dataset

Experimental Design (ML)

Images

- We shrink all images to 64x64, to simplify training time
- batch size configuration and evaluation of 8
- compute averages of multiple FID calculations vs. one fixed test set for each archetype
- Sample a representative dataset to capture “all data” as comparison
- Tune Epochs

Text

- Hyperparameter tune epochs and learning rate
- Bleu Metrics to text generation for both card descriptions and titles

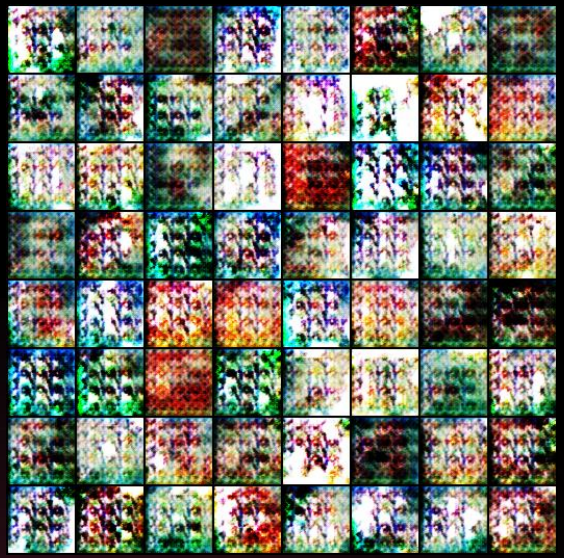
GAN/SDUNET Model Results (FID)

++++

Archetype Configuration	50 Epochs	100 Epochs	150 Epochs	200 Epochs
GAN Custom				
Dark Magician	440.35	490.11	526.13	517.16
Blue-Eyes	467.26	476.35	490.81	504.23
Elemental Hero	417.12	483.15	496.17	429.47
Sampled "All" Data	259.49	294.87	X	X
STABLE Diffusion Modified UNET				
Dark Magician	593.52	X	X	X
Blue-Eyes	X	X	X	X
Elemental Hero	X	X	X	X
Sampled "All" Data	X	X	X	X

GAN Model Results

++++



Elemental Hero

Sample All Data

SD Unet Model Results

++++

Dark Magician Scaled

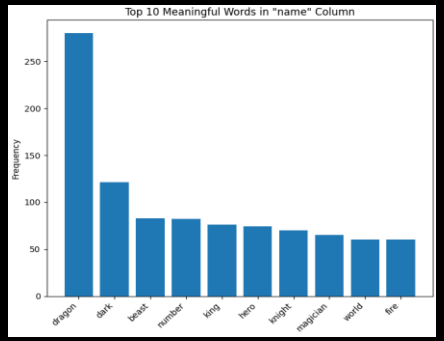
Sample All Data

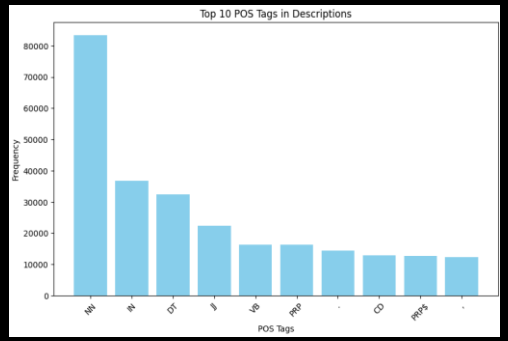
SD Unet Higher Resolution Experiment (128x128)

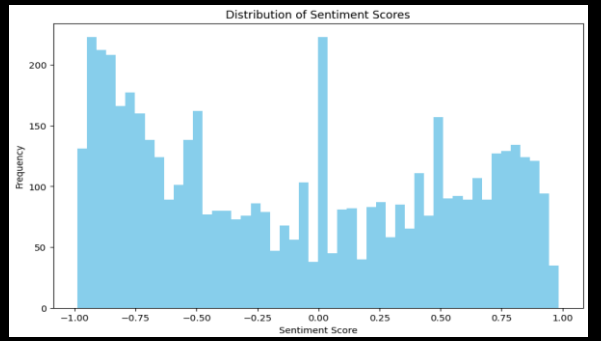
++++

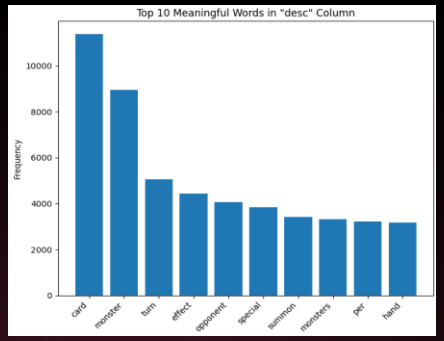
EDA on Text Data

++++









1. Frequency analysis of key terms in card names and descriptions revealed: Dragon appears most frequently in card names, followed by dark, beast, number, king, and hero.
2. Uncovered themes through LDA topic modeling: Topics include equip and enhancement mechanics for warrior-type monsters, modifying attack and defense stats, game mechanics, and point systems. Other topics encompass monster control, card effects, ritual summons, speed strategies, and player interactions.
3. POS tagging analysis on card descriptions: Nouns are the most frequent, followed by prepositions or subordinating conjunctions, determiners, and adjectives.
4. Sentiment distribution analysis: Histogram indicates a binomial distribution, suggesting descriptions tend to be strongly positive or negative with fewer neutral tones.

Works and Tools Known in this space

Show and Tell (CNN-RNN)

Extract features with CNNs and then uses a sequence model like RNN, to generate captions based on these features.

e.g. Show and Tell

Transformer-based Models

Recently active research ongoing
Known for exceptional performance in NLP tasks, to image caption generation.

e.g. VisualBERT, ViT, GIT

Text Generation - Evaluation Metric

Our Metric : Bilingual Evaluation Understudy Score (BLEU)

- We opted to use the BLEU metric for evaluating our image captioning model in GIT.
- Our aim was to assess the similarity between the model-generated captions and the ground truth text data, such as card names or descriptions.
- BLEU, known for examining precision by comparing n-grams in generated text with those in reference text, aligns well with our objective.
- By employing BLEU, we can quantitatively measure how closely the generated captions resemble the reference text, offering valuable insights into the effectiveness of our image captioning approach.

GIT Model Results (Bleu)

++++

Card Description Generation

Archetype Configuration	Epochs 50	
	Learning rate (5e-5)	Learning rate (1e-4)
Dark Magician	0.213	0.193
Blue-Eyes	0.162	0.200
Elemental Hero	0.276	0.274
Epochs 10 / Learning rate(5e-5)		
Sampled "All" Data	0.214	-

Card Name Generation

Archetype Configuration	Epochs 50
	Learning rate (1e-4)
Dark Magician	0.220
Blue-Eyes	0.139
Elemental Hero	0.183
Epochs 10 / Learning rate(5e-5)	
Sampled "All" Data	-

* Due to time limitation, we experimented on different learning rate within same epochs on above three card archetypes.
* Additional hyperparameter tuning can be conducted ahead.

GIT Model Results (Card Description)

++++

The ultimate wizard in terms of attack and defense

when a spell / trap card is activated : banish 1 face - up " elemental hero " monster you control until the end phase ; negate the activation, and if you do, destroy that card. if you control " elemental hero terra firm

you can ritual summon this card with " chaos form ". must be ritual summoned. your opponent cannot target this card with card effects, also it cannot be destroyed by your opponent's card effects. if this card attacks a defense position monster,

- Many correct generations of the text i.e. Dark Magician's description,
- but some strong hallucinations with some room for improvement.

GIT Model Results

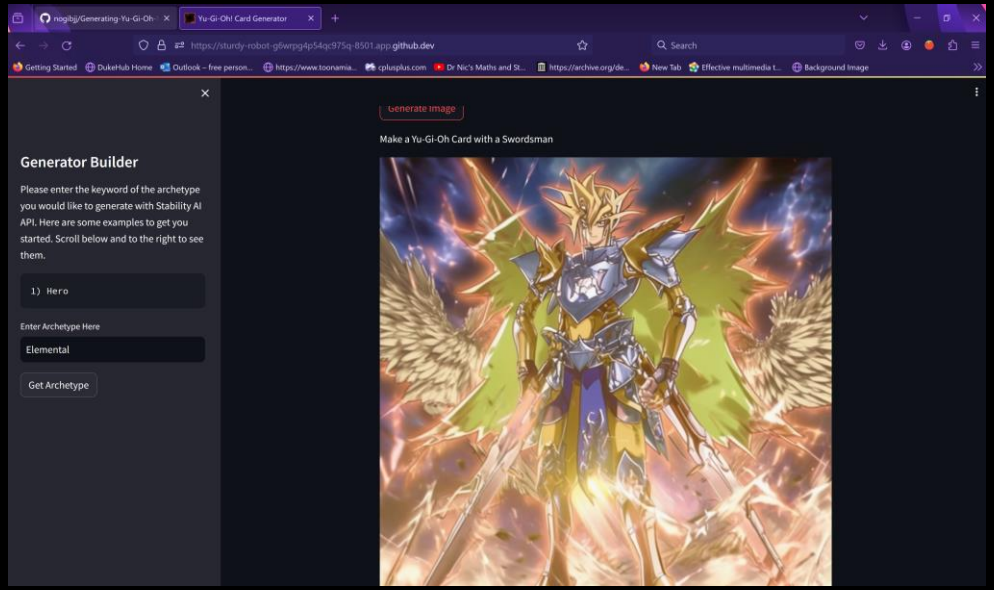
elemental hero phoenix enforcer

timaeus the united dragon

Hallucinations are strong on the right, nonexistent on the left.

Streamlit App Showcase

- Pip install the requirements and run the app. Ask for your archetype and even get the chance to play with the Stability AI api., as well as see some of our work from our repository.



Conclusion

- Singular Archetype selection appear to be not quite efficient for accurate archetype image generation (according to FID scores)
- Diffusion (UNET) does not produce better results for the archetype we saw, but more research is needed after optimizing for avoiding runtimes and memory usage over capacity.
- Text Analyses reveal that card data is somewhat dominated by other archetypes.
    - Maybe our approaches could consider under sampling instead.
- Bleu metric indicates some incomplete or incorrect descriptions; otherwise quite like human generated sentences.
    - However, it is just one metric. There are more, and the same applies for FID (Images)

Major Limitations

- Vicious Training Times for all models (min. 1 hour, max. 2 hours)
- Taxing FID calculations (1 hour long)
- Multiple package conflicts in Huggingface tutorials, as well as pytorch's infamous atagen: grid_sampler_2d issue that plagues style gans.
- Usage of data augmentation was not enough to boost FID, so more complex techniques must be considered beyond those already shown ()
- Text Analytics shows bias, as descriptors of characters bias both sentiment analysis and frequencies
    - Archetypes overshadow others
- How much did the resolution hurt the image analysis?
- Lastly, Time.

References

[1] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, Lijuan Wang, “GIT: A Generative Image-to-text Transformer for Vision and Language”

[2] https://huggingface.co/docs/transformers/main/ko/tasks/image_captioning

[3] https://devpost.com/software/yu-gan-oh

[4] https://medium.com/@jkleiber8/pokegan-generating-fake-pokemon-with-a-generative-adversarial-network-f540db81548d

Thanks

Do you have any questions?

eric.riossoderman@duke.edu
Jiwon.shin@duke.edu