Anime Image Feature Capturing and Image Generations with Generative Adversarial Networks

Yiheng Ye

Abstract

Image generation have also been a hoe topic in deep learning, and it sometimes comes out with several interesting applications, especially With the invention of Generative Adversarial Networks (GANs). There are so many applications of image generations. Those applications are not only focused on image generations on real world objects, but also artists' creations as well. In this article, I will further explore the potential of GANs on generating certain artistic style images and make some improvements on the original model.

1 Introduction

Image generation has been developing rapidly these years, especially after the invention of Generative Adversarial Networks (GANs) in 2014 [1]. There are many applications and developments originated from the GANs. People begin the experiment from MNIST data set and move on to face image generation, scenario generation, and text to image translation. It looks like GANs can really help us reach the ultimate goal of recreation of the reality, However, what about making artistic creations? Or, can we apply GANs on mimic certain artistic style with enough diversity on different topics and small features like the way of adding color to the image? Some of scholars have been exploring those fields and achieves some progress in generating anime character faces with certain adventure game (AVG) style [2], but I want to go further beyond that. In this article, I will makes image generations with GANs in a given group of artists and works and try to mimic their style and generating images that are not only anime girl faces but many other fantastic concepts like Wyrms or even stories. I will also try to generating images with a given theme concept in this article.

The research of generating anime style images mainly begins in 2017, when a group of researcher, mainly from Fudan University, trained their DRAGANs model on a Getchu data set [2]. Before them, there are a few researches on making anime girl images, but non of them reaches the high quality this group creates. They also set up their website and presented their work on Comiket 92 (C92), and the responses from the community is positive as the images they created are giving the nostalgia when AVG is at its high time. This represents their success in mimic the artistic style, but also shows the limitations of their work. Their work only contains a very specific style (Old school AVG) of certain kind of image (human faces), and the images here are lack of details relatively. Further more, as they also admitted, their data set has been influenced by the style shifting of anime images by the time and are making a mixture results. I believe that is also one of the reason why their results looks that "Old School". Therefore, what I want to do is to find a way of training GANs on anime image that has more details, more diverse style, and more topics. In the other word, I want to recreate the art we have in anime style instead of human faces. Our model should be captured "concepts" instead of actual "traits" of the anime image, and should mimic the artistic style. However, mimicking artistic style using GANs turns out to be difficult sometimes as some artists do not have enough artwork to work with. To solving this problem, temporarily, we can learn from other works who also want to

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

create Anime images. PokeGAN [3] tries to generate pokemon image using GANs, and they fix the data set problem by using mirror and color changes. This article successfully reproduce the shape of Pokemon and their design, but the colors on those pokemons are in a mess, which are not ideal since we lose significantly amount of artistic style here. We also can use data augmentation GANs (DAGAN) [4] to help us do the data augmentation work to avoid underestimating parameters but itself will suffer from the small data set we have. It is kind of pointless if one uses a GAN to further augmenting the data set and trains them again using another GAN structure model. I believe applying proper data augmentation techniques are good enough for our project, but it is possible to try out applying DAGAN on our work as well.

My solution to mimic the artistic style is very easy to understand. First of all, I find a training data set with various anime style but generally stays in the domain of modern anime images. I also divides them into different themes so the model can learn topics differently every time we want to create images for a specific themes. This will result in very small data set and thus I have to use data augmentation and/or DAGAN to solve this problem. At last, I am adjusting GANs to fulfill the need of analyzing complex image features as there will be way more details in my training data set. I believe with those implementations, my solution will provide a new way to generate anime artistic image even in a relatively small data set.

2 Related Works

GANs algorithm is first proposed by Goodfellow and his colleaques in their work "Generative Adversarial networks [1], as they developed a model that enable 2 neural network that "competes" with each other to generate new data from the existing data set. This model shows incredible potential in image generation and has been widely used in related domains. 2 years later (2016), a developed version of GANs emerged, which is deep convolutional generative adversarial networks (DCGAN) [5]. This is a type of unsupervised CNN models that is a strong candidate in image representation. After that, multiple GAN variants have appeared to dealing with multiple aspect in image generation and someone had already been trying to apply those models to Anime image generation.

In 2017, a group of researchers from Suzhou University built up a anime style transfer model for anime sketch using Auxiliary Classifier GAN (ACGAN) [6]. This is not a actual anime image generation as what they did was just filling the anime sketch with another similar colored anime image as reference, but it is still a notable try in applying GANs to anime image. Meanwhile, Yanghua et al. [2] developed the anime face generation models applying current GANs technique based on Getchu data set. After that, with the researches developing in this direction, there many interesting applications like PokeGAN [3]. Furthermore, in 2021, researchers from Shenzhen University and National Tsing Hua University combined built AniGAN [7]. This model provides a new possibility of generating anime face images by taking real-world people face as source data and anime faces as reference data to create new anime faces by "drawing" anime character faces based on the traits of the source real people faces. To sum up, there are several works during the years that are concentrating on anime image generations, especially on anime faces.

3 Methods

3.1 Data Augmentation

Data augmentation is the technique of increasing data amount by adding slightly modify data. In this task about image generations, we usually increasing our data size by flipping, rotation, scaling and cropping. For our task here, the benefit of doing so is that in order to mimic artistic style instead of reproducing certain objects (faces, for example), we actually do not care about modifying the shape of our images. We'd like to keep the color consistent throughout the data set but we can flip/rotate the images as we want. After all, neural network are not that "smart" and those minor changes can result in efficiently increasing our data set size for training. However, the concerns of destroying artistic

style in original images and not-so-distinctive data set are still remains, and it is possible that we have to find some more augmentation measure to improve our data set.

3.2 GANs

Generative adversarial networks are a class of neural network developed by Goodfellow and his colleagues [1] in 2014 to use neural networks to do unsupervised learning. Basically the model is made by two multilayer perceptrons, generator G and discriminator D, and they plays a two-play minimax game V(G,D) as follows:

$$\min_G \max_D V(D,G) = E_{x\ P_{data}(x)}[log(D(x)] + E_{z\ P_z(x)}[log(1 - D(G(z))] \tag{1}$$

The training propose of this model is to train D so that we have the best possibility of correct labels from the training examples as well as the samples generated by G. By using this evaluation, we can optimize our model to find a best generation process. Thus, the algorithm we get here for the Vanilla GAN model is:

Algorithm 1 Minibatch stochastic gradient descent training of adversarial neural network. k is 1 here as we are using the least expensive option here. This is the illustration of Vanilla GAN algorithm made by Goodfellow et al. [1]

for number of training iteration **do**
  for k steps **do**
    sample minibatch of m samples $\{z^{(1)}..z^{(m)}\}$ from noise prior $p_g(z)$
    sample minibatch of m samples $\{x^{(1)}..x^{(m)}\}$ for the data distribution $p_{data}(x)$
    update discriminator D

$$\nabla_{\sigma_g} \frac{1}{m} \Sigma_i^m [log(D(x^{(i)}) + log(1 - D(G(z^{(i)})]$$

  end for
  sample minibatch of m samples from noise prior $\{z^{(1)}..z^{(m)}\}$ from noise prior $p_g(z)$
  update generator G

$$\nabla_{\sigma_g} \frac{1}{m} \Sigma_i^m [log(1 - D(G(z^{(i)})]$$

end for

The converge of this algorithm is to reach a state where $p_g = p_data$, as this is the global optimal solution for this model.

Although GANs are very powerful tool for generating images, the vanilla GANs are still very hard to train and it requires large and diverse data. Also, it suffers from its non-convergence and diminished gradient, which makes the GANs even harder to learn from data. The mode collapse is also a major issue as GANs can only generate certain type of "optimal" samples instead of samples with larger variety. However, this is not the problem we get here as we hope the GANs accurately mimic the original images' art style so we can make better artificial images under this scenario.

3.3 DCGAN

One of the solution to the original vanilla GANs is DCGAN, which is one of the most popular structure used nowadays. Basically, the logic of this algorithm is to use strided convolution to replace any pooling layers in the above Vanilla GAN algorithm to preserve spatial information and get better image quality. The structure of this kind of GAN is shown below:

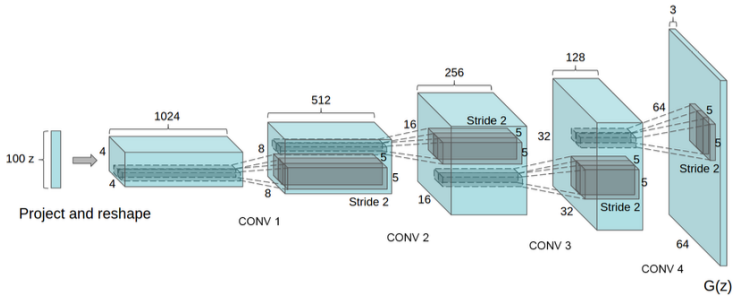


Figure 1: DCGAN generator structure [5]

For generator, ReLU is used as activation function except output, whihc is Tanh, and we use LeakyReLU for all layers in discriminator. Those activation functions are also making differences between the original GAN model, and it improves the model performance especially for high resolution images. LeakyReLU allows a small positive gradient for non-positive unit and its formula is as follow:

$$f(x) = x \quad if \quad x > 0; \quad 0.01x \quad else \quad wise \tag{2}$$

DCGAN indeed improved a lot compared with Vanilla GAN, but it still do not solve some problems in the GAN structure including mode collapse and vanishing gradient,which are embeded in the original cost function. It is still a model that cost many data to learn.

3.4 DAGAN

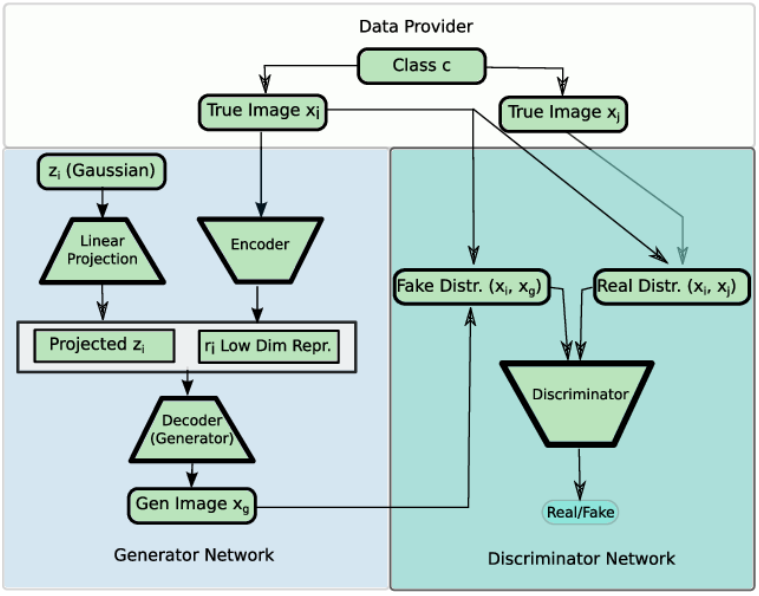


Figure 2: Data augmentation generative adversarial network structure [4]

Data augmentation generative adversarial network (DAGAN) uses a combination of UNet and ResNet as its generator. It uses LeakyReLU as the activation function and has a upscaling/downscaling layer which is following its all 4 convolutional layers. The discriminator of DCGAN is using a DenseNet. Looking at the original report, it has been proven viable on improving classification task using vanilla classifier on the same data set, but its ability to used with other GAN models is still under examinations.

3.5 General Discussion

In the subsection above, I have discussed the possible methods we can be used in this project and their benefits as well as drawbacks. We can find that though GANs are very hard to train and requires lots of data, we can still try to use multiple approach to reduce the data amount we need, which is the new point I want to explore in this project. I will try to use data augmentation+GAN/DCGAN to see if we can have a better image generation result and, inspired by DAGAN, try to combined multiple different type of GANs in generation process to see whether we can improve the image quality, which is also a possible solution to the GANs inherent problems.

3.6 Our Method: Improving discriminator performances with SE blocks,DCSEGAN

On training the GAN models, there are several concerned question especially when we are training the discriminator and it is important if it can provide more reliable classification result on real/fake model. Our solution to improve the performance of the discriminator is to introduce the squeeze-and-excitation (SE) blocks [8] to the discriminator. Although there is works about applying SE blocks to the generator [9], it is just using one blocks in the end of the generator CNN and we are working on discriminator with different scale of applications.

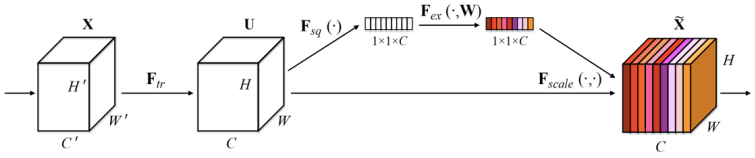


Figure 3: Structure of squeeze and excitation networks [8]

The algorithm of SENets is adding an additional blocks to each original CNN layer to improve how each channels are weighted in the learning process. Basically, it comes with a global average pooling layer and two fully connected layers whose activation functions are ReLU and Sigmoid. After calculating the updated weight of each channel, we scale the weight back to the original CNN network. I will implement those blocks on each layer of original DCGAN's discriminator to see if we can improve our models by a better discriminator.

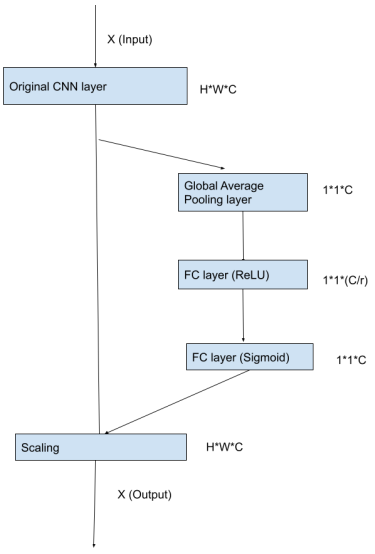


Figure 4: Structure of My CNN layer for discriminator after adding SE blocks

The benefit of applying the SE blocks to discriminator is that it provide a better performance without introducing a pooling method to original CNN layer, as it is using the pooling method to get scaling

parameter only. However, this addition to discriminator do not solve the problems we have discussed in the previous section about general problem of GAN models. The model is still hard to train and the mode collapse problem still exists, which are problems that can only be solved unless we find another innovation GAN architecture.

4 Experiments

4.1 Dataset and Set up

In this project, our main focus is to fulfill the purpose of recurring scene and make our produced images as close to original scene of the project as possible. To find the suitable data sets that contains numerous images of the same theme, we decide to use the card art works from Yu-Gi-Oh trading card game [10]. For someone who are not familiar with this game, I will make a brief explanation here. Basically, this is a trading card game who has over 10000 unique cards and most of the cards are belong to certain "archetypes", and each archetype has unique theme. For example:



Figure 5,6,7: Example card image from archetype "Fire Fist" (left), "Herald" (middle) and "Tenyi" (right).

We can find that the fire fist archetype is a series of warrior with animal spirits with them, herald archetype is a series of angles who have apparent ball-like body features, and the tenyi archetype is basically a series of Chinese dragons. In this project, we are going to work on images on those three archetypes and to see if our method can be generated better images compare to original DCGAN. Those images from the archtypes are going to be resized into 64*64 format and we will have our result images in those shapes as well. Furthermore, we will only use inception score (IS) to be our quantitative analysis standard as although we want to figure out if we can replicate the image theme, we have to realize that even images in original archetypes are varying a lot in terms of shape and some styles. Therefore, the major take way we want to have here is to at least have some consistency on the images we generated. In other words, we want to generating something new instead of replicating the original data set, and Fréchet inception distance (FID) actually cannot tell us that much useful information in our cases as it will be a complex question if we want a result is close to orignal theme but also innovative. Besides that, we will do some qualitative analysis regarding the image quality.

4.2 Pre-test on data augmentation

There is a problem with the data set we have is that it only have very few images for each archetypes. For example, the herald archetype only has 8 different angels, the tenyi only has 7 different dragons, and fire fist, being a little bit better, has 26 different warriors. That is to say, we have to think a way of augmenting of our images while do not influencing the art style of original data set too much. Otherwise we will get a complete replication of original data set. I have done an experiment on herald archetype about applying DCGAN directly or using DAGAN, and I get original images back eventually like this:

Figure 8: Result image for Herald archetypes on DCGAN for 100 epochs without data augmentation

Therefore, I have to make artificial augmentation even it will damage the image quality we have. I decide to use auto augmentation function provided with pytorch and random color jitter. An example of training image is given below:

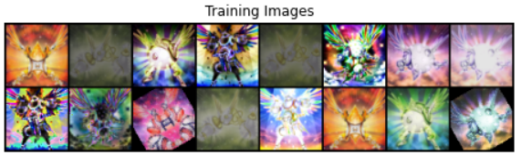
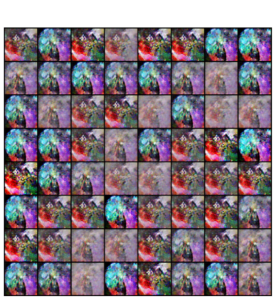
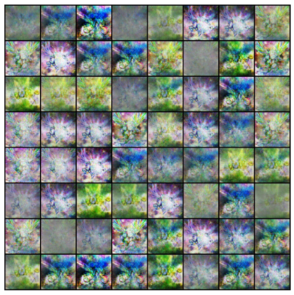


Figure 9: training sample image for herald archetypes.

4.3 Result with DCGAN only





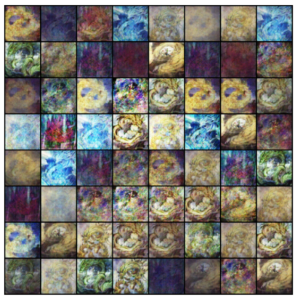


Figure 10,11,12: Result images for Fire fist (left), herald (middle), and tenyi (right) by DCGAN after 30 epochs

We can find that from image quality is tenyi>herald>fire fist. Tenyi has 6 different dragons displayed while the other is suffering from mode collapse (4 kinds of angle for herald and only 2 warriors for fire fist). The inception score is 1.4647540253821436 for fire fist, 1.3820201086955999 for herald, and 1.4886465027757008 for tenyi data set. This is consistent with the visual identification we have as the tenyi data set has the best image quality while the herald is producing some random pixels. Next, we will try to improve our result by training on DCSEGAN. We hope to see a better inception score and lower mode collapse.

4.4 Result with DCSEGAN

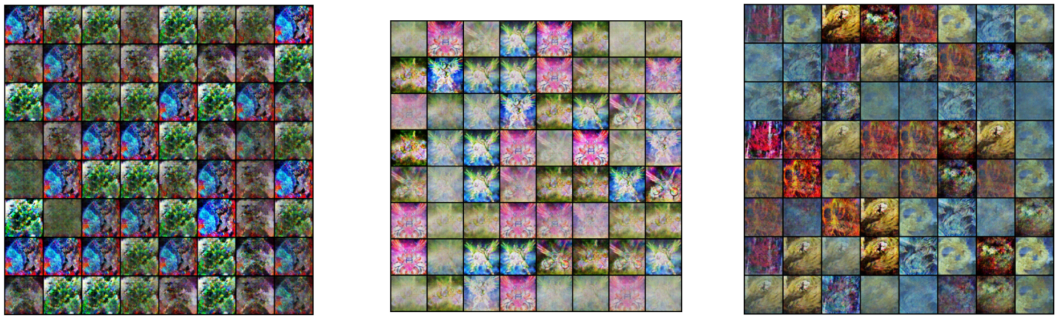


Figure 13,14,15: Result images for Fire fist (left), herald (middle), and tenyi (right) by DCSEGAN after 30 epochs

The inception score is for 1.5366562812808275 fire fist, for 1.5264631296130635 herald, and 1.6155820847534375 for tenyi data set. We can find from the inception score that all the score improved so this justify that SE module is beneficial in producing better images. We can also use qualitative analysis on each image we generated. Clearly, the fire fist image set has more variation now, with 3 4 different warriors and better classification between animal spirit colors as we can find that the output images are grouping warriors according to the colors (blue, green, and white). We can also find from herald result that there are actually no noise images like we previously have in DCGAN result as almost all of them are recognizable angel, and this result also have 5 groups of different outputs instead of 4. The tenyi result also has more distinction between different dragons as well. Overall, the results generated from DCSEGAN is providing better differences comparing with the original images we have (One can check this point by comparing them with the archetypes images in the Yu-Gi-Oh official database [10]. Overall, our strategy is proven to be successful compared with vanilla DCGAN.

4.5 Look into further: Comparing the loss variation in training process

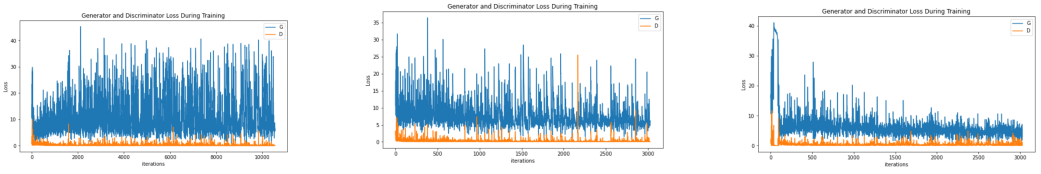


Figure 16,17,18: Training loss for Fire fist (left), herald (middle), and tenyi (right) by DCGAN during 30 epochs

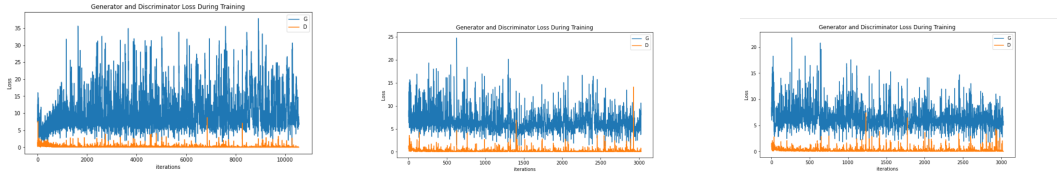


Figure 19,20,21: Training loss for Fire fist (left), herald (middle), and tenyi (right) by DCSEGAN during 30 epochs

To find out why SE is improving the image quality, we can compare the training loss we have in our experiment. Clearly, the DCSEGAN's training processes have overall low training loss for both generator and discriminator for all 3 archetypes, especially for the generator. We can find that the

range and the pike of the generator's training loss are lower in DCSEGAN's process, and we can conclude that SE module is helping us in reducing those losses.

5 Conclusion

In this project, I have conduct experiment on using DCGAN with squeeze and excitation module to make better generation of anime images that are representing specific theme. Our results, compared with the DCGAN, have been proved to be improved both quantitatively and qualitatively. This justify the benefit of SE module as it not only reduces the discriminator loss in training but also generator loss in training even we only apply it on the former. Thus, we can conclude that our model makes better image generation result on capturing anime image's themes comparing to the orignal DCGAN

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks *arXiv preprint arXiv:1406.2661*

[2] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, Zhihao Fang Towards the Automatic Anime Characters Creation with Generative Adversarial Networks *arXiv preprint arXiv:1708.05509*

[3] Justin Kleiber, PokeGAN: Generating Fake Pokemon with a Generative Adversarial Network `https://blog.jovian.ai/pokegan-generating-fake-pokemon-with-a-generative-adversarial-network-f540db81548d`

[4] Antreas Antoniou, Amos Storkey, Harrison Edwards, Data Augmentation Generative Adversarial Networks *arXiv preprint arXiv:1711.04340*

[5] Alec Radford, Luke Metz, Soumith Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, *arXiv preprint arXiv:1511.06434*

[6] Lvmin Zhang, Yi Ji, Xin Lin, Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN *arXiv preprint arXiv:1706.03319*

[7] Bing Li, Yuanlue Zhu, Yitong Wang, Chia-Wen Lin, Bernard Ghanem, Linlin Shen, AniGAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Generation, *arXiv preprint arXiv:2102.12593*

[8] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, Squeeze-and-Excitation Networks, *arXiv preprint arXiv:1709.01507*

[9] Zhang, Fangyan Wang, Xin Sun, Tongfeng Xu, Xinzheng. (2021). SE-DCGAN: a New Method of Semantic Image Restoration. Cognitive Computation. 13. 10.1007/s12559-021-09877-y.

[10] Yu-Gi-Oh card database, `https://www.db.yugioh-card.com/yugiohdb/`