Generating YouTube Thumbnails with WGAN-GP

Saad Moro
Oregon State University
moros@oregonstate.edu

Abstract

Recently, generative adversarial models (GANs) have been successfully used to generate artificial data that passes for real data. YouTube thumbnails offer a preview or summary of video content that can entice users to click on and view a video, and because of this often exhibit certain eye-catching traits, such as bold text captions and exciting imagery, across many different video genres. In this paper, we construct a GAN that generates synthetic YouTube thumbnails. We first build a DCGAN architecture before implementing a WGAN-GP architecture in order to accomplish this, and explore the effect of using different thumbnail datasets on the effectiveness of our GAN output.

1. Introduction

Generative adversarial models (GANs) are a type of generative model that presents generative model as a competition between two separate networks: the first is a "generator" network that produces artificial data given some input noise, while the second is a "discriminator" network that attempts to discriminate between the synthetic data produced by the generator and the real data of the training dataset. As training progresses, the generator network attempts to fool the discriminator and is penalized for not doing so, and the discriminator network must learn how to outsmart the increasingly tricky generator network, separating real from fake data. In this paper, it is our interest to create a GAN that generates artificial YouTube thumbnails. Given a certain amount of training time, hopefully our GAN will produce artificial thumbnails that from a distance, retain the characteristics and general appearance that real video thumbnails have.

At first glance, the prospect of using a GAN to generate video thumbnails seems fraught with difficulty. This is because videos can be about any subject, and may contain anything which one can point a camera at, or which can be recorded on a screen. A GAN asked to generate general frames of film footage of these types would not be likely to succeed at generating convincing images unless it was fed a training dataset of tremendous size. However, YouTube thumbnails do have a certain logic to their graphical organization, since their purpose is to be an eye-grabbing bit of content that impels YouTube users to click on their respective video. Human beings are generally depicted from the chest or neck upwards, facing the camera. Action shots are typically in the foreground of the image. Video titles or attention-grabbing captions in the form of bold text objects superimposed over the background image are a mainstay of clickbait, review and video game content.

Moreover, there are hidden classes that YouTube videos (and thumbnails) belong to, in the form of video genre. Makeup videos typically are up close images of a glamorous face, while video game and reaction videos often feature a seated person (typically wearing a headset) making a face in response to some other form of media, typically displayed in the background.

In making a GAN that generates artificial YouTube thumbnails, we hope to capture some of these aforementioned qualities of thumbnails. We anticipate that our GAN will generate text captions common to many videos, or text-like (but otherwise illegible) regions on our generated thumbnail images. This is due to the fact that while bold text captions are extremely common on YouTube videos, unless a specific word or phrase is exceedingly common, the CNN-based architecture of our GAN has no tools to help it generate sensible words for different video classes.

1.1. Related Works

While the field of GANs is very large, work done on thumbnail generation is virtually nonexistent. In other machine learning fields, thumbnails have been utilized to create clickbait classifiers, and thumbnails have been generated based video data in order to produce video uploads that receive a maximal level of user engagement. The YouTube-8M dataset is an enormous dataset that compiles information about millions of YouTube videos, including the thumbnail, however to the best of my ability I have not seen any other large-scale attempts to aggregate a dataset of YouTube thumbnails. Besides one DCGAN-only project on GitHub, there has been no attempt to create a GAN that gen-

erates thumbnails. This project therefore enters somewhat unexplored territory in the application of GANs to computer vision datasets.

2. Technical Approach

Our dataset is a uniform *sample* taken from the gargantuan YouTube-8M dataset. Due to hardware constraints, only thumbnails from 20,000 of the videos in the dataset were included, and additional feature information originally present in YouTube-8M has been removed, leaving us only with a dataset of unlabelled random YouTube thumbnails belonging to any video genre. For perspective, below is a random sample of three thumbnails from the dataset. This tiny gives a snapshot idea of some of the thumbnails in the dataset, depicting human action in the form of a live musical performance, a man (Vsauce Michael?) facing the camera holding an apple, and an anime girl. In these three images, there are no thumbnails with superimposed bold text, however examples of such thumbnails can be found later in this paper.

Figure 1. Example thumbnails from the YouTube-8M dataset.

The base network architecture that we used was a deep convolutional generative adversarial model (DCGAN), as detailed in [1]. While I do not wish to burden the length of this paper detailing the DCGAN architecture, I will note that we used the same design as in the original DCGAN paper. The DCGAN generator consists of 4 deconvolution layers (ConvTranspose2d in Pytorch) each followed by a batch normalization layer which is in turn followed by a ReLU activation function. The output is then sent through one final deconvolution layer and a tanh activation function, which yields a 64x64 image output, using 3 color channels (RGB). The DCGAN discriminator is essentially the same network in reverse: four convolution layers each followed by a leakyReLU activation function, and then followed by a convlution layer with a sigmoid activation function. Although for problems soon to be touched upon, I did not intend to run the DCGAN for too long, I did train the DCAN on the YouTube-8M dataset for 23 epochs, producing the following sample artificial thumbnails:

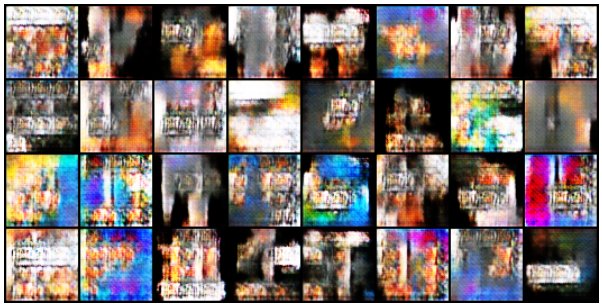

Figure 1. Generated thumbnails from our initial DCGAN architecture after 23 epochs.

These results certainly do not look too good. Every thumbnail is a blurry mess, although with some defined text-region beginning to appear on every thumbnail. One major shortcoming of the DCGAN architecture is that it is prone to *mode collapse*, wherein data is only generated from a single class of data. In the thumbnail dataset, classes are hidden, but existent- the presence or absence of superimposed bold text would itself be a class definition, and I would *conjecture* that had this DCGAN been allowed to train for longer, it would only care to produce thumbnails that depict superimposed text, without much attention being paid to background imagery: suffering mode collapse and only producing images from the "includes superimposed text" class.

In order to resolve this problem of mode collapse, and in order to improve the stability of the GAN training process while also reducing the need to fine-tune training hyperparameters such as learning rate, we then implement the Wasserstein GAN with gradient penalty (WGAN-GP) architecture on top of our DCGAN. The Wasserstein GAN attempts to solve a problem with the minimax objective loss used in other GAN architectures, being that the minimization of the generator's loss function is not always continuous with respect to its parameters, leading to training difficulties ([2]). The Wasserstein distance is an alternative metric which under some mild assumptions is continuous everywhere and differentiable almost everywhere, meaning that the continuity-related training issues faced sometimes by architectures such as DCGAN would no longer apply. Lipschitz continuity (the aforementioned mild assumption) is enforced by clipping the parameter weights of the *critic* (no longer called the discriminator, since the network no longer performs classification) within some neighborhood $[-c, c]$. However, this method of weight clipping can itself lead to optimization difficulties, and so the introduction of a weight penalty on the gradient norm of the critic has been found ([2]) to be an effective way to curb optimization difficulties. Practically, for us that means we should no longer suffer mode collapse, and should be able to generate thumbnails from every hidden class in the thumbnail dataset. We implement the WGAN-GP architecture as it is detailed in

Algorithm 1 of [2], using the Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.999$, and a learning rate of 0.0001. We use the default gradient penalty coefficient of $\lambda = 10$.

3. Results

Training our WGAN-GP on the YouTube-8M sample dataset, we certainly do not fall into the pitfall of mode collapse. In the generated thumbnails, we can clearly recognize humanoid figures with face outlines, and in some (but not many!) of the thumbnails we see the telltale pseudolanguage of GAN-generated superimposed bold text. The undecipherable, runic script of the GAN text appears left-to-right in a clear order, with a very distinctive appearance. However, many of the thumbnails also appear to be bogus, abstract patterns.

Figure 2. Generated thumbnails from our WGAN-GP architecture after 48 epochs.

3.1. Refinement

Examining the results above, we note that although there are recognizable features in some of the generated thumbnails (humanoid figures, text captions), the subject matter of most of the thumbnails is unrecognizable. Many images are simply blurry palettes of color. Upon inspection of the video thumbnails in the dataset, we notice that many video thumbnails *already depict* blurry mixes of colors. If many such images are in the real dataset, then how is the discriminator supposed to learn to identify poorly generated images as fake? If one of the two networks is unable to perform its task, then the GAN cannot produce satisfying output. By using a uniform random sample from the YouTube-8M dataset, our GAN has merely learned how to produce extremely blurry approximations of general videos: swatches of color, humanoid shapes, and text-like objects. Perhaps with a much (much, much, much) larger dataset, our discriminator would eventually learn how to distinguish between generated blurry images and genuine blurry thumbnails, however, due to hardware constraints I felt determined to improve the GAN output while using a small training dataset. Figure 3 below includes some thumbnails which in the low-resolution form that training was performed on might make the training of an effective discriminator network difficult.

Figure 3. Thumbnails which might detract from the effective training of the discriminator network.

In an attempt to amend this issue, I decided to change the dataset upon which our GAN was trained. Previously we have mentioned that all thumbnails fall into hidden classes in the form of their genre, where each genre of video has distinctive thumbnail characteristics. In the first attempt at using WGAN-GP to generate artificial YouTube thumbnails, the dataset used contained a random assortment of video genres. Live music concerts were compared with footage of flying birds, videos of thunderstorms were compared with tech review videos. In order to introduce a greater degree of uniformity in the potential range of characteristics in the thumbnail dataset, I limited the number of video genres to one: video games. I scraped the thumbnails of 20,000 YouTube videos exclusively uploaded with the "video game" tag and available on YouTube's gaming page. Thumbnails belonging to gaming YouTube channels often adhere to a certain style, seen in the example thumbnails below:

Figure 4. Example thumbnails taken exclusively from video game channels.

Typical characteristics include attention-grabbing superimposed bold text, and the heads of gamers reacting in some sense to whatever it is they're playing, superimposed over a background image taken from gameplay footage. There is still a high level of variance in possible characteristics in each thumbnail, however compared to the level of variance when all video classes were included, we have enforced a great reduction in thumbnail characteristics.

Running WGAN-GP on our sample of 20,000 gaming thumbnails, we received positive results, as shown in the sample of 32 generated thumbnails below:

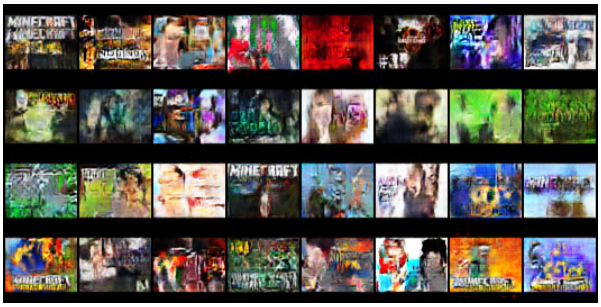

Figure 5. Generated thumbnails from our WGAN-GP architecture on a single-genre video thumbnail dataset, after 58 epochs.

This time, the WGAN-GP generated thumbnails began to look much more like the actual thumbnails they're supposed to mimic. Bright and bold text is prominent in many highlights, and we can observe that the classic thumbnail format of "gamer with headset on, playing video game" is well represented in our generated thumbnails:

Figure 6. Gamer heads; slightly blurry, but still gaming.

Clearly, videos captioned "Minecraft" in the thumbnail occur so commonly in the gaming section of YouTube that the artificially generated text on much of the thumbnails often replicate those letters in their entirety. We see the telltale signs of thumbnail text elsewhere: black or white outlined letterlike objects can be seen floating on many of the thumbnails. With the exception of a video game title such as "Minecraft" there would not be any particular consistency among text objects on uploaded gaming videos, and so in the thumbnails generated by our GAN we only see objects that look similar to text.

Our results are certainly blurry, but this might just be a constraint from using a relatively small set of training data, or from generating low-resolution 64x64 output (which was further shrunk from having black bars outlining the scraped thumbnails). Overall, we have observed a vast improvement in the quality and realism of the WGAN-GP generated thumbnails in compared to the output from the GAN trained on the YouTube-8M sample. By making our training data more specific, our GAN was able to generate thumbnails that actually just look like blurry versions of real gaming thumbnails.

As an additional experiment, I decided to even further narrow down the scope of the training dataset. While my previous narrowed dataset consisted entirely of thumbnails uploaded to gaming channels, I allowed videos from any video game to be displayed. However, due to the nature of the medium, what might be depicted in video games changes quite a bit depending on which game is being played- both stylistically and content-wise. A first-person shooter is likely to look very different from a multiplayer online battle arena game, and both would look very different to a top-down farming simulator. Inspired by the abundance of the "Minecraft" caption in the previous set of generated thumbnails, and due to the popularity and abundance of videos uploaded showing Minecraft footage, I once again scraped the thumbnails of 20,000 YouTube videos of the game Minecraft.

Figure 7. Example thumbnails taken exclusively from channels dedicated to Minecraft gaming footage.

Certain elements, once again, are very prevalent in Minecraft YouTube videos. Images of the 3D player model, tools and weapons iconic to the game, attention-grabbing red arrows, and the signature voxel graphics of the game terrain are displayed prominently in nearly every upload.

Training our WGAN-GP, a sample of 32 generated thumbnails, and a closer look at 3 noteworthy thumbnails is shown below:

Figure 8. Generated thumbnails from our WGAN-GP architecture on Minecraft-only video thumbnail dataset, after 55 epochs.

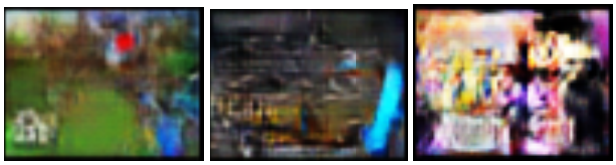

Figure 9. Generated thumbnails depicting an stylistically Minecraft-y scene replete with grass blocks and trees, a sword, and a player character.

By even further limiting the scope of our training data set, we have greatly improved the quality of generated thumbnails. While our experiment using just gaming videos gave us the vague outlines of human bodies and faces (and some largely-indecipherable bold text), our Minecraft-only dataset has produced realistic in-game footage. The hallmark characteristics of thumbnails are all there: superimposed bold text, series-numbers, and scenes of action. Tools and weaponry from the game are shown, as are the characteristic blocky avatars that players play as. The red arrows commonly used in attention grabbing thumbnails make an appearance as well, although the direction they face seems to be a bit random.

4. Conclusion

In this paper, we applied the use of the WGAN-GP architecture in order to generate YouTube thumbnails. YouTube thumbnails can display nearly anything, however due to their attention-grabbing purpose generally adhere to certain characteristics, such as the use of block text superimposed over a background. By using WGAN-GP to generate thumbnails, we avoided the issue of mode collapse and were able to generate (blurry) thumbnails of many different subjects. By narrowing down our training dataset, we were able to significantly improve the efficacy of our GAN, having it produce recognizable thumbnails in the video gaming genre, and thumbnails for a specific game that replicated gameplay elements such as terrain, tools, and player models, while retaining the general structure of thumbnails.

5. References

[1] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028, 2017.