Albart: An Album Art Generating System

Wes Ackerman, Samuel Giraud-Carrier, S. Jacob Powell
Computer Science Department
Brigham Young University
Provo, UT 84602 USA

Abstract

We present a system capable of generating interesting and novel album art for a given song. We select images based on the song lyrics and collage these onto a background generated by a GAN. Once assembled we pass this image through a cycleGAN to perform style transfer based on the genre of the song.

Introduction

Computational Creativity is a widely studied and extremely broad field. This is due to the countless number of activities, works, and processes that humans consider to be creative. Creative artefacts include things like paintings, music, films, video games, architecture, speeches, etc. and can include less-traditional forms such as a style of athletic play or clever problem solving. This wide range of things considered ”creative” poses an interesting problem for creative systems, as oftentimes humans have a hard time clearly defining what is ”creative”. Not all creative tasks are equally simple, and because of their differing levels of complexity each is judged by different standards. Because of the differing measurements of quality, it is required to consider a creative system in the context of the creative task it is performing, and in the case of Albart, this is generating interesting album art that matches the ”feel” of the song and artist it is portraying.

Albart is a creative system that generates album art for a given song. Its goal is to generate album art that is interesting, and ideally novel art that can be easily tied back to the artist and song it relates to. This measure of how well it does can be seen by colors and images that it includes within the album art image. Album art is an interesting creative problem to try to solve, because when done by humans, it often makes little sense. Music itself is very symbolic and ethereal, and album art often matches music's abstract nature. There are, however, common themes and connections that frequently appear in certain genres and by certain artists that make sense to the viewer/listener. Thus, they either evoke emotion directly, or remind the viewer of the type of music they are portraying, and the same emotion is experienced that is in the music itself. It is also not uncommon to have completely unrelated album art for the sake of being different or interesting; this is considered equally creative and interesting by the creators and viewers of those types of album art. Being both very flexible and very specific makes the problem easier and harder in different ways. It might seem like the flexibility of the album art, sometimes being completely unrelated, makes the problem easy to solve; but this introduces the ideas of framing, intent, and deliberate choices. These are general problems when it comes to computational creativity, and are at the core of the album art problem. An artist is being very deliberate when they choose a specific symbol that relates directly to the music in the album, but another artist is considered equally as deliberate when they choose something very abstract or unrelated at all. Both are considered creative, with equal amounts of intention, and the viewer knows that the artist made certain choices for a reason, whether that is known or not. With that being said, a human may not treat art the same way if they knew it was created by a machine; therefore should they be told the artist was by a machine at all? Should it be framed as work done by a human? It is also possible to try to produce artefacts well enough that they are indistinguishable from human works, but this may be missing the goal entirely of being creative.

In our case, we choose to enjoy the artefacts that Albart produces and not attempt to specifically quantify each of its actions into differing levels of creativity or intention. For the most part, it is clear why Albart makes certain choices and places certain pictures and colors in different places, since it is still in the relatively beginning stages of functionality. Albart essentially creates an interesting colorful background, that was learned based on the general idea of many instances of album art. It then analyzes the lyrics of a song in order to find small images that it can collage on top of the base background. Having placed related images on the art, it places the song title and artist on the image and finishes by performing a style transfer of the album's genre, based on album art of the same genre. This system is basic, but produces visually interesting images that are fun to look at and think about in relation to the song they represent.

In this paper we will discuss the task of album art creation in further detail, and then cover similar systems and previous work done in the field of art and album art. We will then describe Albart's system in detail and how it makes decisions and produces each part of its images. We will then discuss the implications of the system and the art it produces and show example results. We also will describe future work that we would like to do with Albart and the general direction we would go given more time. This will also include some of the ideas of the original intended system, the challenges it has, and how it would work with the intended mechanisms.

Task

In the human world, creation of album art is extremely fickle and unpredictable. It also occupies a very unique space in the realm of creative works, in that an artefact is creative and artistic in its own right, but also may convey information about, similar to, or simply in conjunction with a piece of music, which is also its own independent artistic piece. The album may also attempt to capture the entire album in a picture, taking what we have identified as one or a few of three possible approaches. Typically, album art falls under one of the following three styles or themes: connection to the real world, connection to the music, and aesthetically pleasing.

When an album cover is directly related to the artist, it might display the artists themselves or a well known logo or insignia of the band. The cover could be just a photograph of the artist, or their name spelled out in text. When the album art relates to the genre it would typically make use of visual styles that make it clear what the genre is, such as color schemes. Reggae albums, for example, are known to use reds, yellows, and greens, and often use the symbol of a lion. These features would define a piece of album art as part of the reggae genre.

When the cover is a depiction of the intent of a song or the collection of songs in an album, the image tends to need more interpretation. The artwork used can be either symbolic or literal. Sometimes the cover will make use of elements in the lyrics themselves (like a literal wall for Pink Floyd's album ”The Wall”, even though the wall that the artist is writing about is mostly symbolic in nature). At other times, an album that conveys certain emotions will use colors or images that reflect these emotions. Such as reds for anger, blue for sadness, or a scary image if the principle feeling of the music is fear. For the depiction of principles (political or otherwise) the cover may allude to well known images such as covering of the eyes, ears, and mouth, or simply have text that points to a specific topic.

Finally, album art that is purely aesthetic album art is simply fun to look at or attempts induce emotions directly. Usually they have no clear connection to the album or its music at all. The artist may have simply made use of design techniques and styles that led to an interesting cover.

These three approaches to album art, connection to the real world, connection to the music, or pure aesthetic art, are a good way of capturing the idea of album art in order to roughly judge how good it is in the context of the artist's general intention. An artist who includes a picture of themselves on the album likely intends for the listener to learn what that person looks like to help build their reputation. Another artist may choose to connect to their audience not directly with a picture of themselves, but with a feeling or symbolic image that represents a relatable situation or emotion. It doesn't quantify the measurement of quality art, but it does help frame the problem.

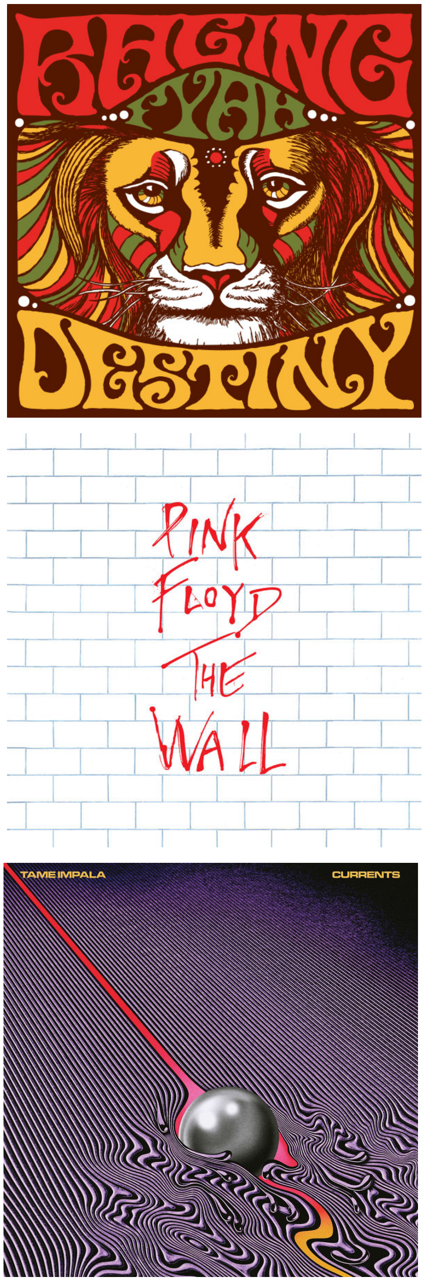

Figure 1: Examples of connection to real world (top), intentional (middle), and aesthetic (bottom) album art. From top to bottom: ”Destiny” by Raging Fyah, ”The Wall” by Pink Floyd, and ”Currents” by Tame Impala.

When it comes to generating art with a creative system, the problem becomes broad and specific at the same time. It is broad in the sense that there are lots of different albums with art that is considered good, yet fall into different classifications of the three techniques listed above. For example, two pieces of album art may both be considered very good for their albums, while one is completely clear how it connects to the music, (symbolically, literally, in the genre, etc.), while the other is very abstract (new artist or entirely aesthetic). This means that computationally, there are two entirely different problems being solved, which is where the problem becomes more specific. A general album art generator would need to choose one of these paths, with some sort of intention, and solve that niche problem in order to be a general album art generator. For certain genres or artists it may be very literal, while for others very symbolic or aesthetic. Our system aims to capture at least one part of the broader problem, which is more of a clear connection to the music, it doesn't do so naïvely or singularly. Albart also aims to add some color and aesthetic value, to make the art more interesting and fun to look at. In our future work section, we expand more on how this problem can be accomplished more broadly as well as more effectively. In general, we consider any album art that invokes interest or emotion to be successful on some level. Sometimes a connection to the music or artist is desirable and other times it's not, so we don't use this in our initial judgment of the system extensively; but it may come into play much more drastically as the system grows to solve the more general problem.

Previous Work

At first glance, a Generative Adversarial Network (Goodfellow et al. 2014) seems a logical choice for generating novel images. GANs have proven to be extremely useful in many contexts, and we also make use of them in our system. However, GANs are limited by the data they are trained on, making it difficult for them to generalize to broader contexts without needing to retrain. We wanted a system that would be able to generate novel images, but also be able to do so consistently and allow for generalization to a broader scope.

One of the best examples of generating images with meaning is the Painting Fool (Krzeczkowska et al. 2010). In this initial paper, their system learned to build a collage of images based on news articles. In the next iteration (Cook and Colton 2011), they further developed this idea by adding emotional response to the article. Their system builds an image as a collage of other images selected based on the words found in the body of the article. Using the most common words and a user provided emotion, the system creates an image that has intentional meaning. They also provide some basic Non-Photorealistic Rendering techniques to make the resulting image seem more painterly. However, one drawback to the Painting Fool's collage generation is the restriction of templates for image placement in the final output.

Along with collaging, our system also uses style transfer techniques. In particular we used the CycleGAN (Zhu et al. 2017) architecture to recognize patterns in a set of images, and regenerate these patterns on a different set. The CycleGAN is limited by the correlation of all the images in both datasets, so training on a set of images that are not well related will generate poor results. The more prominent the features the image set shares, the better the system will learn to transfer these features to another set with its own set of related features. After training, if an image is provided that does not closely match the dataset on which it was trained, the system will not be able to successfully transfer features from one set to the other.

Architecture

Our system consists of several related pieces. The system scrapes data from the Spotify API (Spo) to obtain the names and information of recent popular songs. We scraped the most popular albums of 2017, and then scraped all the songs on each album obtained. We removed unneeded fields, and ones that were too verbose, from each track and stored the information in a database. The final database we used contains 700,000+ tracks. The Spotify database stores links to the album art rather than the image itself, so we wrote a scraper to obtain the art for each album. It iterates over each stored album, visits the image's URL, downloads the image, then encodes the image and stores the result in our database.

We also wanted to allow our system to use the lyrics of a track to make decisions about what to generate. Since Spotify data lacks lyrics information for tracks, we made use of the Genius API (Gen) to obtain lyrics for each track our system runs on. To find the lyrics for a given track, we search the artist name and track name using the Genius API, and use the first result if both names match acceptably. Since Spotify and Genius sometimes represent track titles and artist names differently (e.g. Genius listed one artist's name as ”Michelle Ingrid Williams” while Spotify listed the same artist as ”Michelle Williams”), we considered the fields a match if one contains all substrings in the other (e.g. ”Michelle Ingrid Williams” contains both ”Michelle” and ”Williams”, so the two will match). As long as both artist name and track name match in this manner, we consider the lyrics to be correct, and use the lyrics when generating art for the track.

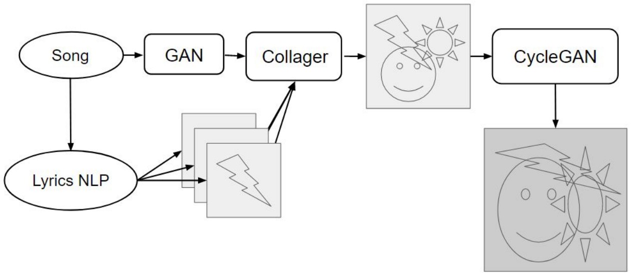

Figure 2: Model of Albart system components.

Our system uses a generative adversarial network (GAN) to create the background for each generated artefact. Our final version of the GAN used the DCGAN architecture described in (Radford, Metz, and Chintala 2015). It was trained for 300,000+ iterations on album cover images from our Spotify dataset. The latent feature space used by the system is 128-dimensional, with a learning rate of 0.001 for the generator and 0.005 for the discriminator. The discriminator is trained 5 times for every 1 time the generator is trained. The hope was that it would generalize well enough to create new and interesting album art images. Though we trained the GAN many times with different hyperparameters, we were unable to generate complex images. We hypothesize that the space of album art is too varied for a GAN to be able to produce results that are both complex and variable.

However, the GAN learned texture and color variations, so it was used to generate backgrounds.

Using the lyrics from the Genius API, we performed some Natural Language Processing in order to extract useful subjects for image search. The python library SpaCy (Spa) provides an easy-to-use API for doing in-depth language processing, and so we used it in our pipeline. The lyrics for a chosen song were loaded up and then the lemmatization of each of the nouns counted. The lemmatized form of a word is a word without any inflections; that is, for a noun, it is in its singular form (plurality of a noun is an example of an inflection). Although the words were counted by their lemmatized form (in order to use one counting number for words that are simply in different forms, such as ”dog” and ”dogs”), the form of the first occurrence of a given lemmatization, with its inflection is used as the final noun that is listed. Each noun is counted and sorted by frequency of occurrence, with more frequent nouns appearing first. This made it much easier for images to be found based on these nouns. As discussed in the Future Work section, we would like to extend this functionality to extract more semantic and specialized meaning from the lyrics rather than just the most frequently occurring noun, but the most frequent nouns often reflect the main subjects of the song, so these worked nicely.

After extracting the nouns in order of frequency (with the inflected form that occurred first in the lyrics saved as the value in the list of nouns), a simple Google image search was performed. We decided to use Google because of its simplicity, availability, performance, and functionality. It was very simple to use and doesn't require any keys or tokens; it doesn't have any rate limiting, and can provide as many images as needed; it is great at finding a large number of images in a very short amount of time; and finally it provides lots of filtering functions and ways to specialize, filter, and limit search and search results so that the best possible images can be found for the given problem. In our case, we found that using Google's built in transparency filter worked quite well with our intended collaging step, since the images would not all appear as simple squares, but as objects. This made it so we didn't have to make or find an object segmenter in order to extract the object from the image. We also made use of the clip art filtering functionality because it seems to provide more semantic representations of the objects in the search. This way, we could easily connect the object to the word, and therefore hopefully understand the intention that the system is using to pick its images based on the lyrics. This also made the end result much more aesthetically pleasing since there aren't just a bunch of squares overlapping each other.

The next step is to take the top few images found in the Google search, and place them on the GAN-generated background. We used a mostly random approach. We place the first image in the center of the background, and then randomly choose points in the image that aren't too close to the edge. We also experimented with different random sizes in order to vary the final result. The collaging step is quite straightforward at present, but could easily be extended to produce many varying results for more interesting outputs. We also add text over the album art describing the song and the artist for that artefact, but we wait to add the text until after the genre style is performed on the image in the next step.

For our final step, we found an existing implementation of CycleGAN (Zhu et al. 2017) that can be used to apply any set of images' style to another set. We decided to use the genre information of an album in order to do this. It required more scraping than we originally had, but we were able to obtain the genre information for every track/album that we had, and then learned the styles based on the album art for each genre. We elected to train a CycleGAN for each genre, and were able to complete models for country, pop, rap, reggae, indie, and hip hop genres. Once the genre style transfer model for each of the genres was trained, we simply ran album art produced from the previous stage through the CycleGAN corresponding to the track's genre, and then added the previously described text on top of it. This step was intended to help blend the collage together with the background and help the entire image feel like one solid image. It also adds frequently used colors for a given genre to make it fit more with what a viewer might expect from album art of a given genre, while not taking away too much of the novel imagery. A good example of this is in Reggae, lots of times the colors red, yellow, and green are used, so the style transfer for the Reggae genre reflects those tendencies.

Overall, the system then produces one or a number of images for a given track or collection of tracks. These are then displayed, sorted, or even filtered by humans and used as desired. One of the things we would have liked to do was use the GAN's discriminator to pick the ones that fit the best, generating multiple images and choosing ones that the GAN approves of. This is a great method for filtration, and is described more in the Future Work section. Currently the system produces interesting images, and we are pleased with its results. Eventually it will be even more interesting to see which images it chooses when comparing different generated images.

Results and Evaluation

In Figure 3 we highlight some of our better results. The image chosen by our model and placed on the image has a clear correlation to the song's main idea. Each image is directly related to the title of the track, so the correlations are clear. The first shows a whiskey glass, the second a frightened face, the third a knife stabbing, and the fourth a Groucho Marx mask. These chosen images represent times when our model was able to use the lyrics of a given song to find an image that relates well to that song. Here, the model meets our goal of representing well a part of the song in an image. Someone could listen to these songs and identify these generated images as relating to the song, because they convey an idea from the song.

In these cases, the style transfer also helps these images convey emotion and to feel more like other albums in their genre. The first, ”Ain't Just the Whiskey Talkin'” is a country song. The art has been given a brown, woody texture. Many country albums have earthy colors and nature scenes, so this image takes on more of a ”country” feel from the

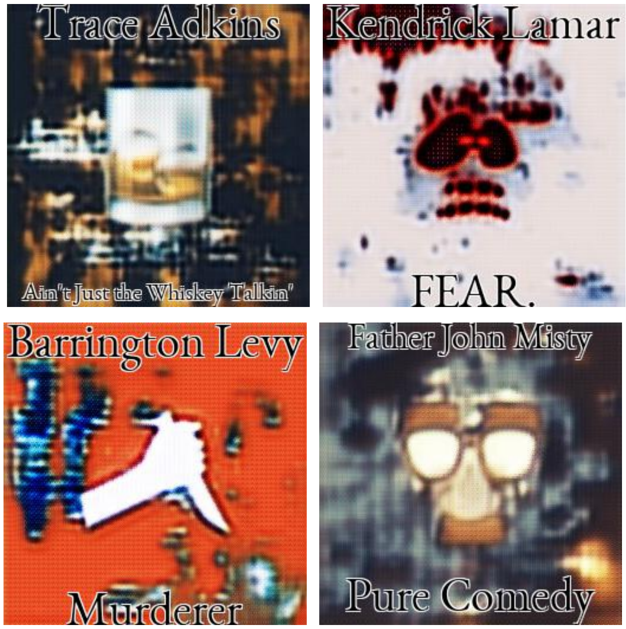

Figure 3: Some of the better results of our system.

browns and pattern given it by the CycleGAN. The second, ”FEAR.” is a rap track. The style transfer has brought more emotion to the art by getting rid of all but the stark, contrasting colors. The black and red face on a strange white background helps amplify the fearful look on the cartoon face. Rap albums often contain lots of red and black. The third, ”Murderer”, a reggae track. It's style only partially matches, since reggae uses lots of reds and greens. And last, ”Pure Comedy” is an indie rock track. It has been given a ”lo-fi”, grainy texture, and subdued colors; styles that are used by many indie-style albums.

We assert that these images partially reach our goal of representing a song via an image. The objects shown in the art are central ideas from the track. This isn't always how humans make album art, but as mentioned above, it is one common way. The style transfer used helps the art to follow genre conventions more closely, and also amplifies the emotion of some of the objects. The weak area in our generation of these images is that much of the creativity still comes from outside sources. The model has no way to represent the ideas/emotions expressed in an image, or to make choices about what clip-art to use relating to those ideas.

We also include some of our poor results to explain the issues with our system. Because we select images based on the most common words in a song, the images are sometimes difficult to interpret. Many words can have multiple meanings in different contexts and the image chosen for a word may not fit the actual meaning when used in the song. For example, the first image in Figure 4 depicts one of Dr. Seuss' things on the cover. The word choice was more than likely ”thing” but because this is an ambiguous word the image choice did not convey the proper meaning. Another example of poor picture selection is in the cover created with a Weepin' Bell pokémon in the center. This was produced from the word ”weepin” which in proper context was referring to crying, not to a video game creature.

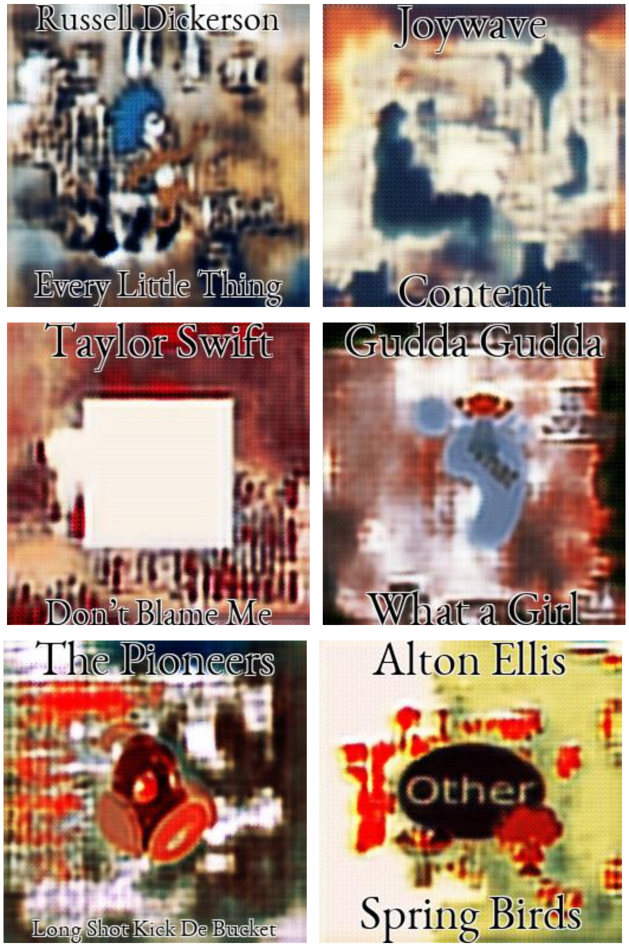

Figure 4: Some of the bad results of our system.

Another issue was images that were closely related to the meaning of the word, but were not particularly good images for use on a cover. Examples of these can be seen in the middle row. The large footprint with the word ”What” on it comes up whenever a song makes excessive use of the word ”What”. The word itself is not particularly helpful in representing the song, so the use of this picture does not help us in our objective to make album art that has meaning for the song. The same can be said for images that hold no information. The blank square that makes up most of the cover on the left does not help us understand anything about the song it was using as its basis.

Other issues we notice happen when the image becomes too cluttered by having too many images superimposed on each other such as the bottom right example. Because of our very basic collaging system, the composition of multiple images is not always ideal, leading to severe overlapping and obstruction in the final result. We also have issues with the style transfer blurring the collage so that the images were no longer distinct as in the top right artefact. These issues are some areas we would need to explore and improve in future work.

Future Work

Our future direction would begin with us revisiting our initial attempts to build our original model. We spent a lot of time getting the different modules to work within the original system, but it simply required more time than we had allotted. The original system (which is also our intended future work) would start with Spotify song analysis data as well as lyrical information in the same way that our current system has them. The pipeline of data flow is slightly different from training than it is for testing, so we will cover the training pipeline first, which begins with sparse auto-encoders.

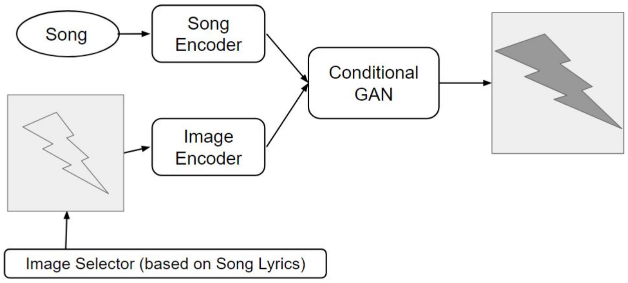

Figure 5: Original structure of the Albart system.

The first of the two is simply trained on images. We wanted the image encoder to be able to encode general images to learn sort of the general semantic meaning of what makes an image an image, and not noise. This way we could ideally have two similar images be close in vector space when encoded. For a similar reason, we would encode the Spotify data sparsely as well, so that similar songs would have similar encodings. These two encoders would be used as the first step both in training and testing when dealing with images or songs (we used the Spotify analysis information as the main representation of the song, trusting that their machine learning data was accurate in analyzing the tracks). We then would train a Conditional GAN based on the encoded song, and encoded album art image (encoded by the general image encoder; the image encoder is not specific to album art), and the genre of the album/track that it is training on. The Conditional GAN allows the generated image to be scored based on how well it matches the genre rather than just whether it passes for album art or not. The GAN would also take in a random vector concatenated with the encodings, where, during testing, environmental, temporal, or ambient readings and data can be used in order for different artefacts to be produced at different times, essentially representing the ”mood” of the system. The Conditional GAN would eventually be trained sufficiently in order to produce images that pass as album art that fits within the given genre. This concludes the training portion of the description of the pipeline.

It should be noted that there are also modules that are responsible for scraping the Spotify track data, Genius lyrics information, and general images from Google for both training and test. These are straight forward for the most part, but the image searches are based on a noun count of the lyrics. This is how we currently are performing the image search, and in the future we would expand this to do more advance Natural Language Processing (NLP) on the lyrics to find more interesting images or images that fit the lyrics much better.

The testing portion or generative version of the pipeline is similar in general to the main system, but has vital differences. The input image to the image encoder is going to be either a human-chosen image (sort of an inspiration object given by the person wanting album art) or chosen entirely by the system. This could be done randomly, or based on the lyrics as explained previously. The track would then be looked up using the Spotify API or previously scraped Spotify data and encoded normally. The system would also then perform ambient sensor readings such as temperature, weather, time of day, political climate, news reports etc. (can be expanded arbitrarily) to obtain interesting data points to use for the ”random” data used before, introducing a notion of mood. The Conditional GAN could then produce multiple images from different inputs, such as different general images or ambient readings, or just produce a single image. Producing multiple images would allow the Conditional GAN to score each generated image to see if it matches the desired genre, and the best of the generated images could be displayed.

This was our original intention for Albart, and even though we got the encoders working well, we were unable to complete the full pipeline due to time constraints and the difficulty of the problem. A lot of time was spent scraping the required data, and so we had to find a more feasible solution for the time given to us. Either way, the future direction of Albart would be very interesting to pursue, and it would produce much more relevant and possibly more interesting artefacts of album art.

Contributions

At the base of the contributions that Albart provides we feel that it continues the conversation on intent that was started by previous collage systems (Cook and Colton 2011). Albart is able to find the lyrics for a given song, analyze it (and eventually understand it more semantically), and find images that are relevant to those lyrics. Currently it is a rudimentary approach, simply counting the lemmatization of nouns and finding transparent clip art for each of the top nouns. Our desired system intends to go into much more depth with lyrics and understanding the themes in them, and will likely do more than just a simple Google search to find or create the desired images. This is the best source of what may be intent in the current system, but in the desired system, there are other points as well. The encoders themselves will learn what constitutes the general learning of the field (or of general images) and so aren't necessarily novel in what they learn and how to encode the information. The GAN, though, will learn from multiple inputs and discriminate its own output to produce hopefully better output than typical. It will also produce different outputs as the world changes (new images) as well as the ambient situation (from the temperature etc. to determine mood).

We hope that even if the current or future iterations of Albart don't necessarily output images that fool people into thinking they are human generated or even have some sort of clear connection (or deliberate disconnection) to the music that it will still present fun and interesting images to look at. The large variance in existing album art makes it easy to produce varying colors and gradients that make interesting backgrounds and aesthetics. We are pleased with the results that Albart produces and have been able to learn a lot about general generative models as well as what it might mean to have intent or to be creative, as a human or machine.

References

[Cook and Colton 2011] Cook, M., and Colton, S. 2011. Automated collage generation-with more intent. In *ICCC*, 1–3. Citeseer.

[Gen] Genius web api. `https://docs.genius.com/`.

[Goodfellow et al. 2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

[Krzeczkowska et al. 2010] Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clarky, S. 2010. Automated collage generation With Intent.

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

[Spa] spaCy - industrial-strength natural language processing in python. `https://spacy.io/`.

[Spo] Spotify web api. `https://developer.spotify.com/web-api/`.

[Zhu et al. 2017] Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.