Visualizing Music with Machine Learning

A Major Qualifying Project
submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfilment of the requirements for the
degree of Bachelor of Science

Submitted by: Samantha Crepeau
Advised by: Jacob Whitehill and Oren Mangoubi
Date: 17 March 2021

Abstract

The problem of generating art through the use of artificial intelligence is vital to our understanding of novel generation and creativity by machines. Inspired by this problem, our goal for this project was to develop a system that generates music visualizations; these are images that are meant to evoke the overall theme and mood of a song. Through the creation of these music visualizations, we aimed to further explore the idea of creative artificial intelligence through AI generated art and how it compares to human-created art in terms of the overall process, the impact on the viewer, effectiveness in conveying mood and theme, and other qualitative metrics. We successfully developed a system to generate abstract music visualizations comparable in aesthetic to those generated by a human artist. We used these music visualizations to examine their effectiveness in conveying mood and theme. We found that AI can be used to convey the mood and theme of a song with some success through abstract images, as participants in a survey used to determine the effectiveness of the visualizations were correct about 50% of the themes using one method and 50% of the moods correct using an alternative method.

Contents

1. Introduction

1.1. Music Visualizations

Our goal for this project is to develop a system that will generate music visualizations; these are images that are meant to invoke the overall theme and mood of a song. A human analogue can be found in abstract paintings inspired by songs, such as the paintings of Melissa McCracken, an artist who converts the songs she listens to into artwork. An example of such a piece can be found in Figure 1.1. By creating music visualizations of our own, we aim to explore the idea of AI generated art and how it compares to human-created art in terms of the overall process, the impact on the viewer, effectiveness in conveying mood and theme, and other qualitative metrics. We also will compare different methods of generating these visualizations.

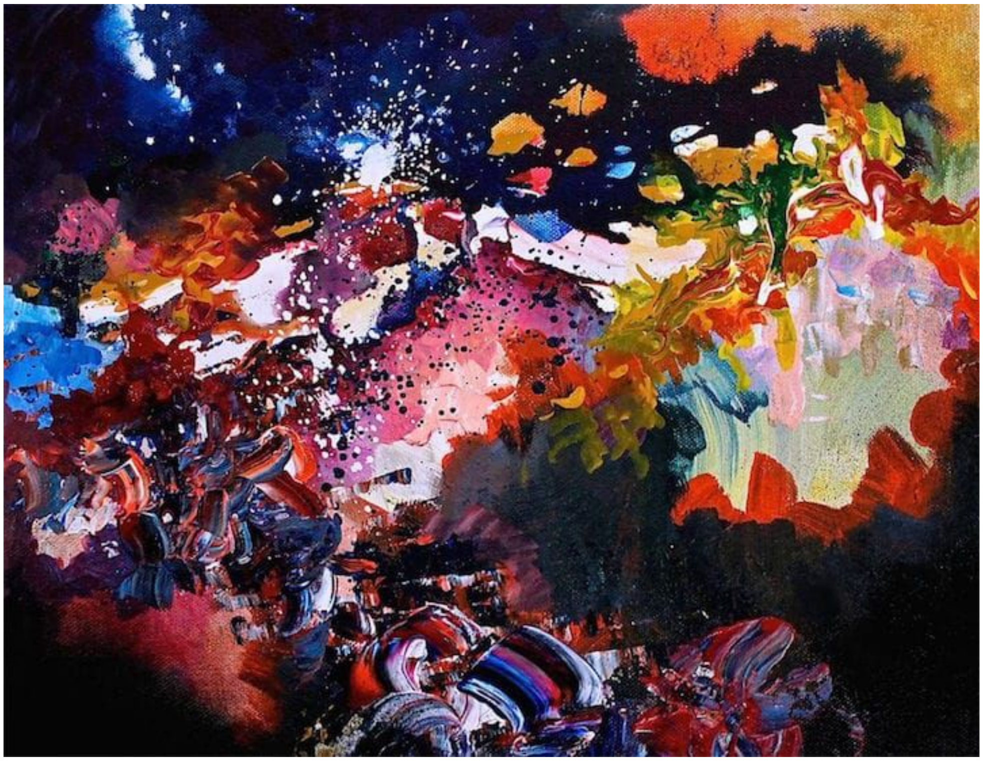
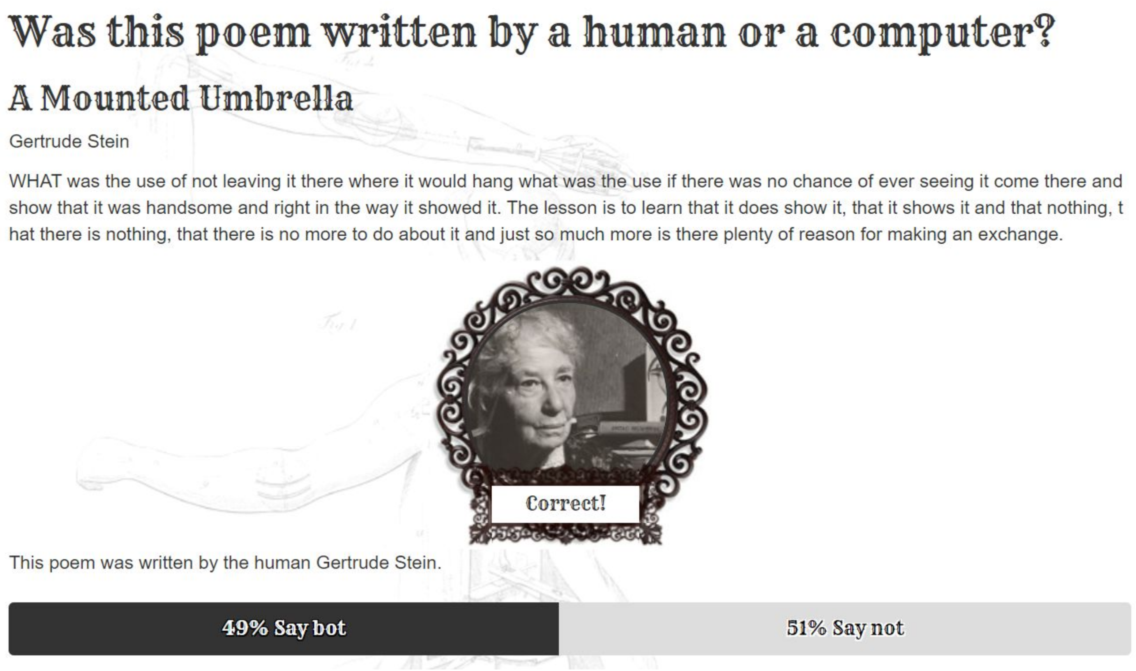


Figure 1.1: An abstract painting by Melissa McCracken, inspired by the song Karma Police by Radiohead. From https://mymodernmet.com/melissa-mccracken-synethesia-paintings/.

1.2. Creativity in Artificial Intelligence

Given the description “a cat sitting next to a bush with white flowers”, a human would likely have no trouble visualizing the corresponding image without ever having seen such a scene. For an artificial intelligence, however, this is no small task. Current algorithms are still far from being able to consistently generate accurate, realistic images from any given text description. Now if we ask a human to visualize a song in the same way as they might visualize text descriptions or a scene from a book, they would likely have more trouble with this exercise. It is a far more abstract task. A song is poetic, dynamic, open to interpretation; it may invoke certain moods or images or concepts in the mind of the listener. It cannot be easily condensed into a single image as a text description can be. However difficult this task may be for a human, the element of abstractness gives us more room for error when we ask an algorithm to generate a corresponding visualization for a song.

A similar example is if we were to ask an AI to generate a poem versus a coherent piece of text. It is far simpler to generate text that is open to interpretation and lacking any specific meaning than it is to generate text that is logical, meaningful, and consistent. The website Botpoet.com allows us to observe the success of an AI in creating poetry as a human does. The website asks users to be the judge in a Turing test and decide whether a random poem was written by an AI or a human poet. As evidenced by Figure 1.2, it can be difficult to determine whether the poem is AI- or human-generated (Kurt 2018).



Was this poem written by a human or a computer?

A Mounted Umbrella

Gertrude Stein

WHAT was the use of not leaving it there where it would hang what was the use if there was no chance of ever seeing it come there and show that it was handsome and right in the way it showed it. The lesson is to learn that it does show it, that it shows it and that nothing, that there is nothing, that there is no more to do about it and just so much more is there plenty of reason for making an exchange.

This poem was written by the human Gertrude Stein.

Figure 1.2: The Turing test result for the poem “A Mounted Umbrella”. From http://botpoet.com.

Not only can AI art be human-like in quality, but it is also becoming more widely acknowledged as an art form. In 2018, a portrait produced by an AI was auctioned and sold for $432,500 at Christie’s New York auction house. It hung amongst pieces produced by the likes of Andy Warhol and was the first AI-generated portrait to come up for auction, according to Christie’s. The piece was created by a Generative Adversarial Network, or GAN (Cohn 2018). Signed in the bottom right corner as an artist’s signature is the min-max function introduced by Goodfellow et al. in 2014 as the basis of the GAN model.

1.3. The Usefulness of Machine Learning in Art

Machine learning is an important tool in the creation of computer-generated art. Because machine learning algorithms improve through experience, they can learn how to mimic specific artists and genres of art or generate novel art pieces based on what they have learned. We will explore examples of algorithms that accomplish this in Section 2.6. We will also create our own system using machine learning algorithms in order to produce our own novel art pieces. These algorithms will aid us in determining song moods and transferring the style of one image to another image in order to generate music visualizations.

1.4. A System to Generate Music Visualizations

To create simple music visualizations, we will take a song's title and the musician as input, obtain a content image representing the theme of the song and a style image representing the mood of the song, then combine these two images using a machine learning algorithm to create the final output. To generate the content image, we will first use a simple API to access the lyrics website Azlyrics.com and find the lyrics of the desired song given the title and musician name. From these lyrics, we need to find images that are relevant to the song content. To achieve this, the lyrics are parsed for nouns and verbs that will be used as image search keywords; we will then use the Unsplash image search API along with these keywords to obtain relevant images. The keyword images will then be combined using edge detection to generate a content image for the visualization.

Next, we must obtain the style image. Using a neural network that classifies song moods based on audio features, the song will be classified into one of four different moods: happy, sad, calm, or energetic. The corresponding mood will be used to find a corresponding abstract image from a set of images that have been classified into one of the four different moods. For example, a "happy" abstract image is colorful and bright. We will also experiment with types of style images, such as actual paintings rather than abstract images. The goal of this is to see in what ways the style image affects the final output image. As the final step, we will use the Neural Algorithm of Artistic Style (NAAS) introduced by Gatys et al. to transfer the style of the style image to the content of the content image, giving us a complete music visualization (see Section 2.5 for more on NAAS). The whole process is depicted in the diagram found in Figure 1.3.

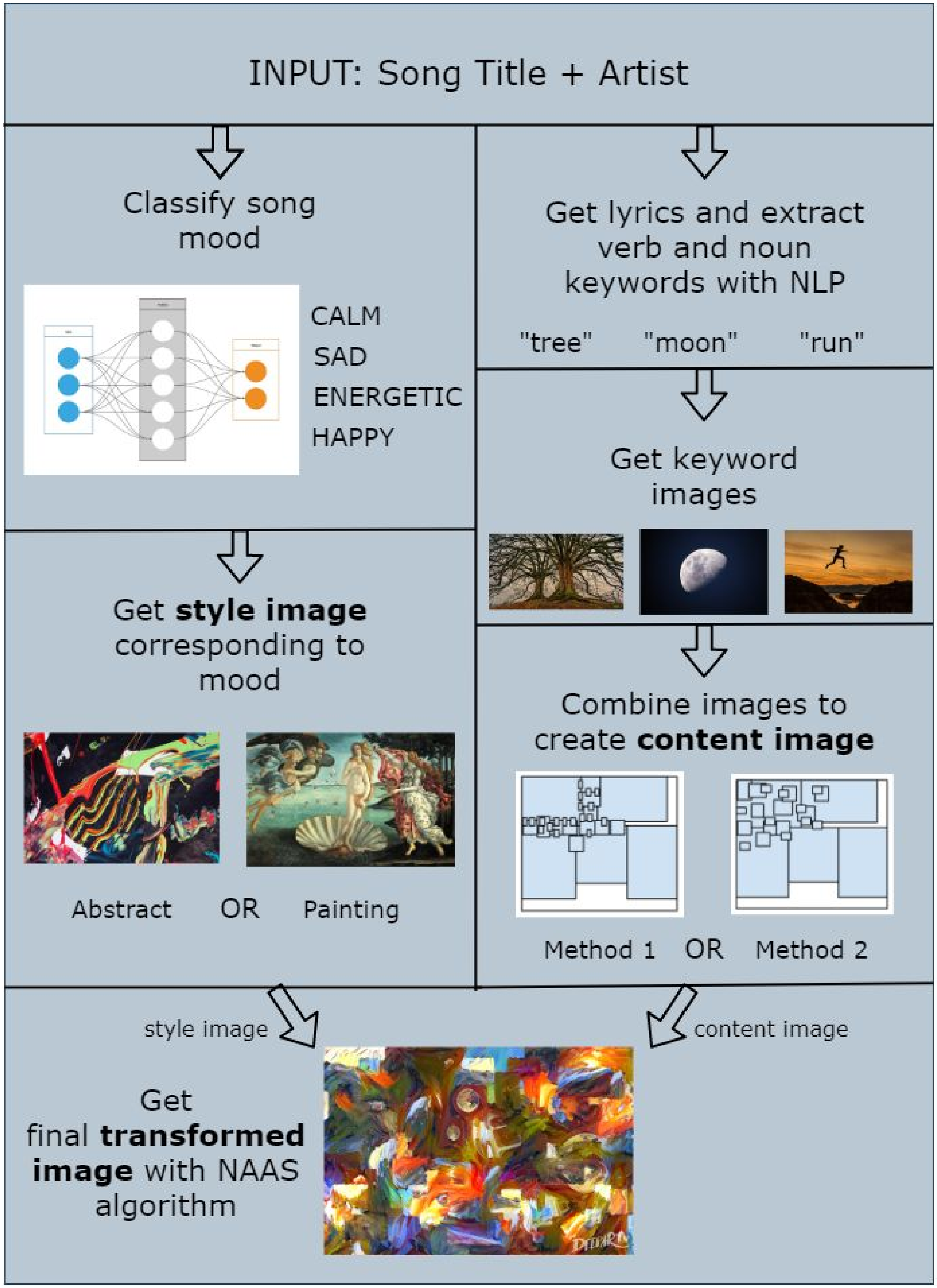

INPUT: Song Title + Artist
Classify song mood
CALM
SAD
ENERGETIC
HAPPY
Get lyrics and extract verb and noun keywords with NLP
"tree"
"moon"
"run"
Get keyword images
Get **style image** corresponding to mood
Abstract
OR
Painting
Combine images to create **content image**
Method 1
OR
Method 2
style image
content image
Get final **transformed image** with NAAS algorithm

Figure 1.3: A diagram depicting the system pipeline. The final image was created using deepart.io, which implements the NAAS algorithm to generate images.

2. Literature Review

2.1. Lyric Extraction and Processing

To obtain the lyrics of a given song, we can use a simple Python API wrapper that allows us to access data from a lyrics site. An API wrapper we can use is Azlyrics (Horrigan 2018), which will allow us to extract data from Azlyrics.com. We will also need to process the lyrics to extract keywords which will later be used as search terms for finding images. To do this, we can use a natural language processing library like SpaCy. This library allows us to determine which part of speech a word is, such as noun or verb. We can also use this library to find "noun chunks", which are phrases where a noun is attached to a word describing the noun, such as "autonomous car".

2.2. Music Mood Classification

We need to determine the mood of a given song in order to find a corresponding style image. In music mood classification, songs are typically categorized using the psychologist Robert Thayer's model of mood, which is visualized in Figure 2.1 (Thayer 2000). There are eight moods or categories in this model, though these can be simplified to four by ignoring the interactions or combinations of the moods of happy, sad, calm, and energetic.

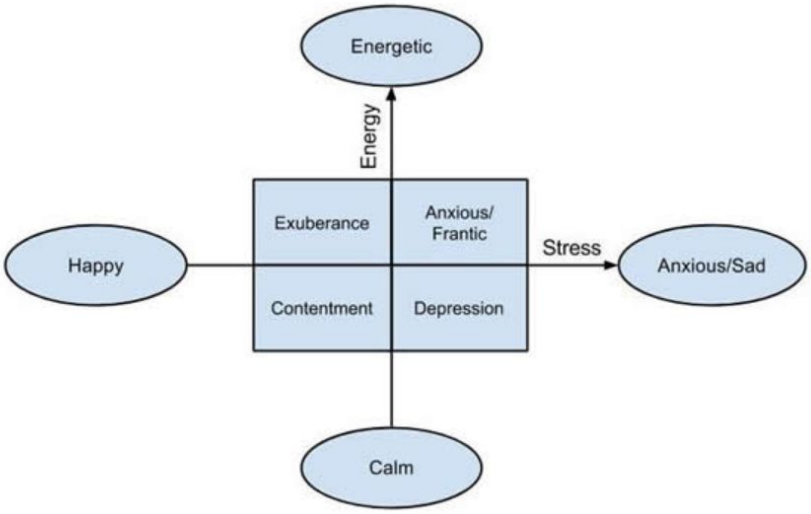


Figure 2.1: Robert Thayer's model of mood. From https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/.

To classify the mood of a song, we can train a model to learn which audio features and combinations thereof correspond to each mood. For example, songs with a faster tempo are more likely to be classified as energetic or happy. Audio features for a song can be obtained through the Spotify Web API; these features, as defined by Spotify, are:

- Acousticness: A confidence measure of whether the track is acoustic
- Danceability: How suitable a track is for dancing
- Energy: Represents a perceptual measure of intensity and activity
- Instrumentalness: Predicts whether a track contains no vocals
- Liveness: Detects the presence of an audience in the recording
- Loudness: The overall loudness of a track
- Speechiness: The presence of spoken words in a track
- Valence: A measure describing the musical positiveness conveyed by a track
- Tempo: The overall estimated tempo of a track

Of these features, all but loudness and tempo, which are measured in decibels and beats per minute respectively, are represented by a value between 0.0 and 1.0. For the model to classify song moods and learn which values of the audio features are related to each mood, a dataset of song audio features labeled with the corresponding mood is used to train the model. A deep neural network is suitable for this task.

2.3. Canny Edge Detection and Hough Transform

In combining images to create a smooth content image, we must identify a method to determine image edges. We consider the approach of Canny edge detection along with the Hough transform. Canny edge detection is an edge detection algorithm which can be implemented using the computer vision library OpenCV. The algorithm first reduces image noise using a Gaussian filter which causes a smoothing effect. Next, the edge gradient magnitude and direction for each pixel is determined. Using these values, pixels that are unlikely to belong to an edge are dropped to 0. The final stage involves two threshold values, a minimum and a maximum, which are given by the user and determine which of the final edges are kept or discarded based on the value of the intensity gradient as compared to the thresholds. The resulting image exhibits strong edges (Canny Edge Detection, n.d.).

To identify the vertical and horizontal image edges that make up the content image, we can use the Hough transform, which can be implemented using OpenCV. The Hough transform allows us to detect lines and shapes. Lines can be represented using ρ, which is the perpendicular distance from the origin to the line, and θ, the angle between this perpendicular line and the horizontal axis. The Hough transform uses a voting procedure which consists of incrementing the number of votes a possible line has when it detects a point on the line. If the candidate line accumulates enough votes, it is determined to be a line. The HoughLines() function in OpenCV returns the values of ρ and θ for each line. We can also use the probabilistic Hough transform to specify the minimum length of a line and the maximum space between line segments to treat them as one line (Hough Line Transform, n.d.). Based on the angle of each line returned by the function, we can easily determine which lines are vertical or horizontal.

2.4. Convolutional Neural Networks

The basis of the NAAS algorithm for transferring image styles is a convolutional neural network. Each layer in a CNN represents input data which has been manipulated by means of a mathematical operation called a convolution. This operation is a series of dot product operations between a matrix of input data, in this case an image, and a smaller matrix of weights, called a filter. The filter is three-dimensional in this case to account for the height, width, and color channels of the image. As the filter moves across the image, the pixel values are multiplied with the filter values and added together to produce a scalar output. This process is shown in Figure 2.2. The filter detects specific features in the image; a scalar output with high value corresponds to a strong correlation between the feature and the area of input data being filtered. The outputs are then stored in a feature map, which completes the process of a single convolution (Nolan 2019).

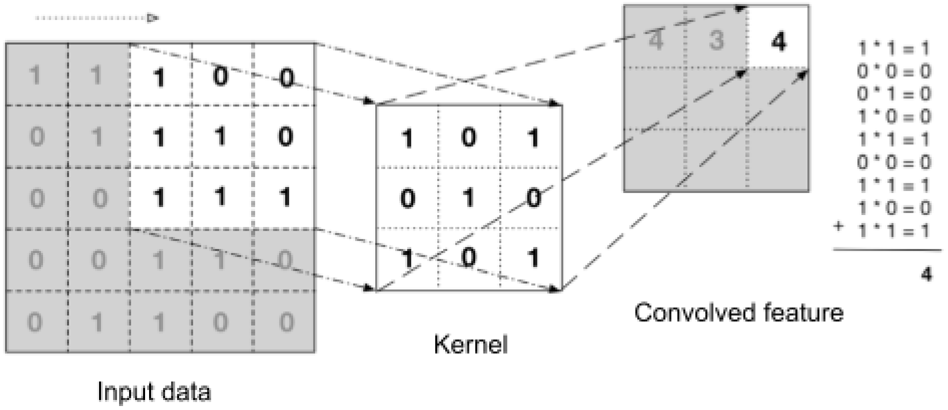


Figure 2.2: A representation of a single step in the operation between two-dimensional input data and the filter, or kernel. The filter is moved across the input data, centering around a different pixel with each step. The values of the input data and filter are multiplied using the dot product, and the resulting scalar product is stored in a feature map. From https://medium.com/machine-learning-for-li/different-convolutional-layers-43dc146f4d0e.

In a CNN consisting of multiple convolutional layers, each resulting feature map is used as the input in the next convolution. The values of the filters are learned over time by the model using backpropagation; the filters begin with random values and become increasingly useful in representing the desired features as new values are learned. Another component of the CNN is the activation function, in this case the rectified linear unit activation function (ReLU), which is applied after each convolutional layer. ReLU is defined as f(x) = max(0, x); the function forces every negative value to zero. This nonlinear transformation of the feature map is important to obtain a high accuracy for classification.

After a convolutional and activation layer, a feature map may be condensed using a pooling layer to reduce the necessary computational power and help prevent overfitting. The most common method of pooling is called max pooling, where the maximum value in each region of a certain size in the feature map is carried over and all other values are discarded. An example of this method is shown in Figure 2.3. This method helps the model focus on the most useful information while ignoring the rest (Nolan 2019).

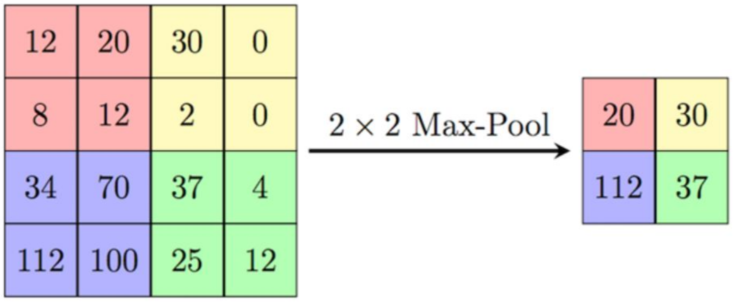


Figure 2.3: An example operation by a max pooling layer, with the original feature map on the left and the pooled matrix on the right. From https://computersciencewiki.org/index.php/Max-pooling_/_Pooling.

2.5. A Neural Algorithm of Artistic Style

The Neural Algorithm of Artistic Style (NAAS) introduced by Gatys et al. in 2015 is the algorithm central to generating our visualizations. It acts as the final step in our process and is responsible for converting each amalgamation of images into something more artistic. The algorithm uses two images, a style image and a content image, to create a single output image consisting of the style of the style image and the content of the content image.

The convolutional neural network that acts as the basis for NAAS is the VGG-16 network, which has been trained on the ImageNet dataset of over 14 million images each belonging to 1 of 1000 classes. However, only the convolutional layers are used in the NAAS network; the fully connected layers that enable classification are not used. The complete VGG-16 network is shown in Figure 2.4 (Nolan 2019).

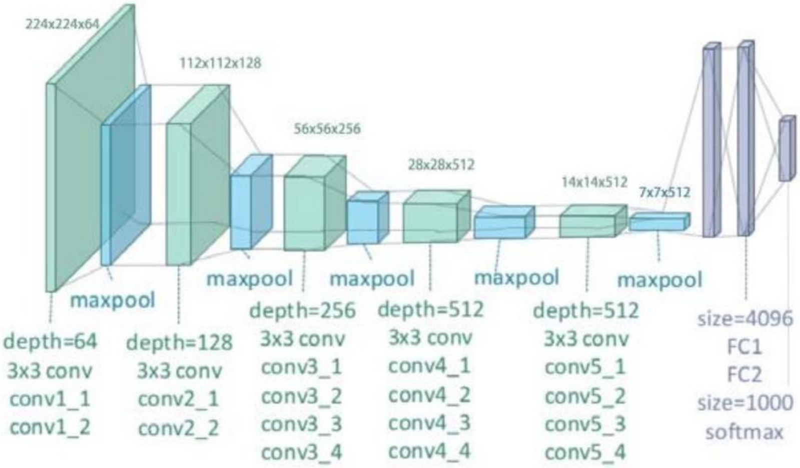


Figure 2.4: The complete VGG-16 network. The final three layers are not used in the NAAS. From https://towardsdatascience.com/a-neural-algorithm-of-artistic-style-a-modern-form-of-creation-d39a6ac7e715.

With VGG-16 as our basis, the step that defines the algorithm is the method in which the style of one of our chosen images is transferred to the content of our second image. More precisely, we transfer the content of the content image and the style of the style image to the optimization image. The optimization image is a white noise image, initially a set of random pixel values. The features of the content image are represented by the output after the content image has passed through VGG-16; the values used are from the layer conv4_2 after ReLU has been applied (Nolan 2019). In order to transfer the content to the optimization image, the optimization image is altered to minimize the loss function

$$L_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}\left(F^l_{ij} - P^l_{ij}\right)^2 \tag{1}$$

Where F is the matrix representation of the optimization image at layer l of a CNN, F_{ij} is the activation of the ith filter at position j in layer l, P is the matrix representation of the content image at layer l, and P_{ij} is the activation. By minimizing the loss function, we minimize the dissimilarity between the optimization image and content representation (Gatys et al., 2015).

To transfer the style of the style image to the optimization image, we find another image that matches the style representation of the original style image by minimizing the mean-squared distance between each value of two Gram matrices, one representing the features of the original style image and the other representing the features of the image being generated. The style loss for a single layer is given by

$$E_l = \frac{1}{4N_l^2M_l^2}\sum_{i,j}\left(G^l_{ij} - A^l_{ij}\right)^2 \tag{2}$$

Where G is the Gram matrix of the optimization image, A is the Gram matrix of the style image, N is the number of feature maps in layer l, and M is the number of elements in the N feature maps. To improve the results, the loss is computed at multiple layers, yielding a total style loss of

$$L_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l \tag{3}$$

where w_l are weights corresponding to the contribution of each layer to the total loss. By minimizing the style cost function, the optimization image becomes increasingly similar to the style image. To generate the final image consisting of the style of the style image and the content of the content image, the following final cost function is minimized:

$$L_{total}(\vec{p},\vec{a},\vec{x}) = \alpha L_{content}(\vec{p},\vec{x}) + \beta L_{style}(\vec{a},\vec{x}) \tag{4}$$

where α and β are constants for content and style reconstruction. These constants are weights which allow for adjustments of the content and style (Gatys et al., 2015). An example of the algorithm being used to transfer the style of one image to the content of another is shown in Figure 2.5. The images that we transform using this algorithm will exhibit a similar effect as in the figure.

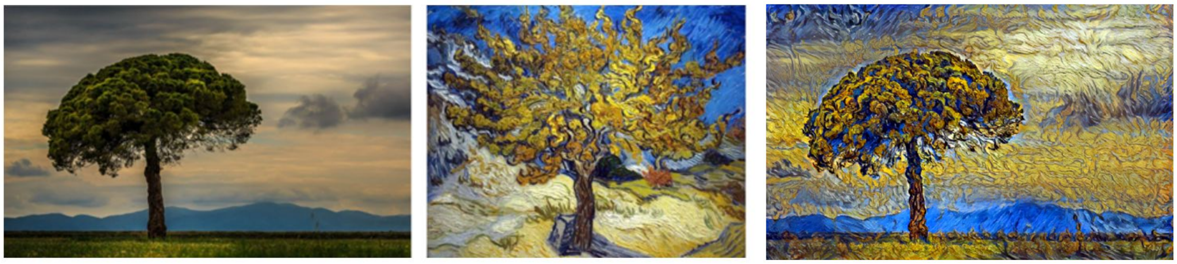


Figure 2.5: An example of usage for the NAAS network. From left to right, we have the content image, the style image, and the output of passing these images through the network. From https://towardsdatascience.com/a-neural-algorithm-of-artistic-style-a-modern-form-of-creation-d39a6ac7e715.

2.6. Generative Adversarial Networks

Generative algorithms are a powerful tool in creating art using AI. While we do not make use of GANs in this project, they are relevant to our review of methods of image generation. GANs use unsupervised learning to generate outputs that could have realistically come from the dataset used to train the GAN model. This model is composed of two sub-models, the generator and the discriminator. The generator learns to generate realistic samples, and the discriminator tries to determine whether the generated samples are real or fake, meaning from the original dataset or generated. The two models play a game defined by a min-max loss function: the generator tries to maximize the probability that the discriminator will classify a fake sample as real by generating increasingly realistic samples, and the discriminator tries to minimize this probability by improving its ability to determine a fake sample from a real one. For a well-trained model, this probability will reach 0.5, indicating that the discriminator can only guess whether a sample is real or fake (Goodfellow et al., 2014). Examples of images generated using GANs can be found in Figure 2.6.

Figure 2.6: Samples from the models trained by Goodfellow et al., 2014. The samples in the left image were produced by a model trained on the MNIST dataset. Those in the right image were produced by a model trained on the TFD dataset. The rightmost column in each image shows the most similar training example to the sample to its left. From https://arxiv.org/abs/1406.2661.

In the context of art, GANs can be used for tasks such as generating novel paintings of subjects such as flowers or landscapes, or for converting a photo into a painting in the style of a specific artist. GANGogh is a GAN-derived model that can combine components of many different pieces of art to create completely new art pieces. The model is trained on over 80,000 images labeled with the style and genre of the art piece. An image might be labelled with the style “Baroque” or “Gothic” and the genre “flower painting” or “cityscape”. The generator produces images for a specific genre, and from those images the discriminator chooses samples that it classifies as that genre with high confidence and also judges as being real (Jones 2017). An example of such images can be found in Figure 2.7.

Another model which can be used for artistic purposes is CycleGAN. CycleGAN accomplishes image-to-image translation by transferring the style of one domain of samples to the content of a different sample from another domain of samples. It achieves this by learning the mapping $G: X \rightarrow Y$ where X is the source domain and Y is the target domain. The inverse mapping $F: Y \rightarrow X$ is also learned, and together cycle consistency is enforced such that $F(G(X)) \approx X$ (Zhu et al., 2020). It is able to convert one image into another in the style of a specific artist, as can be seen in Figure 2.7.

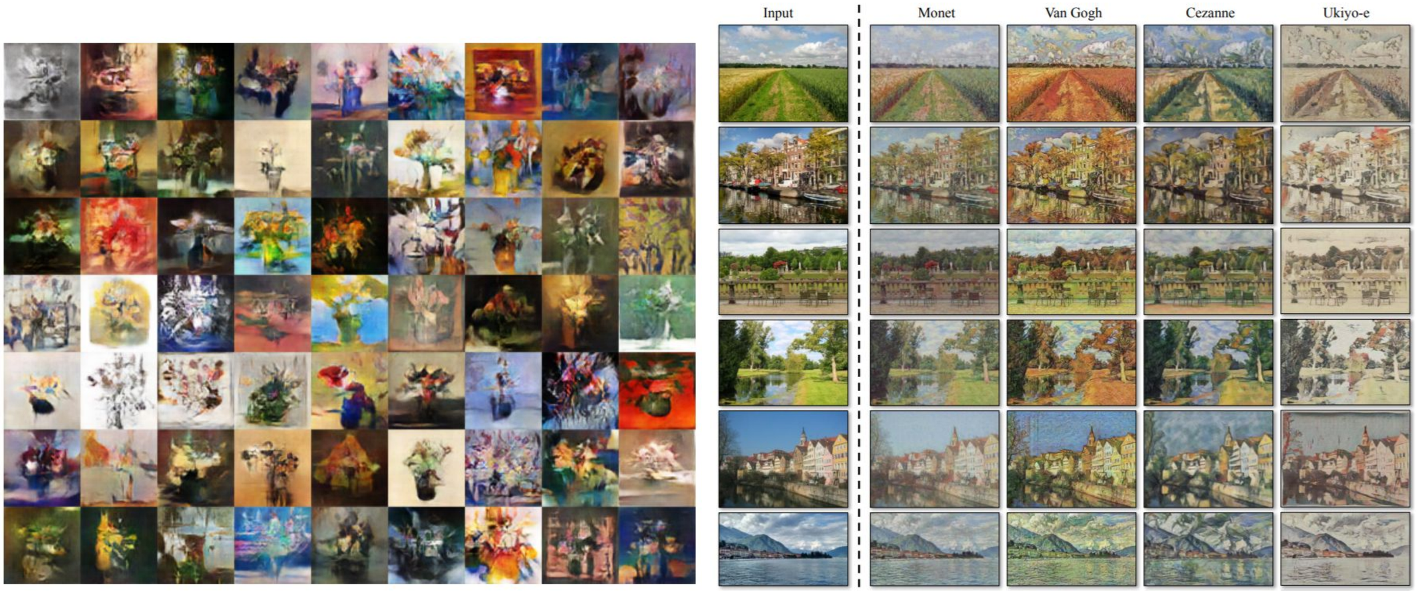


Figure 2.7: On the left, paintings of flowers generated by GANGogh, from https://towardsdatascience.com/gangogh-creating-art-with-gans-8d087d8f74a1. On the right, photos converted into paintings in the styles of various artists using CycleGAN, from https://arxiv.org/abs/1406.2661.

3. Methodology

Here we describe our system for generating music visualizations in detail. This includes our process for extracting keywords from song lyrics, finding corresponding images for these keywords, creating the content image, finding an appropriate style image, and the final transformation using the NAAS algorithm. We also describe our approach for evaluating the effectiveness of each method for creating the content image.

3.1. Lyric Extraction and Images

We begin with our method for obtaining and processing the lyrics for use as keywords in searching for images to combine for the content image. We start by using the API wrapper Azlyrics to extract the lyrics of a song from azlyrics.com. Given a song title and the name of the musician, we can extract the lyrics with the `lyrics()` method. Next, we use the SpaCy Python library to obtain the nouns and verbs from the lyrics. This can be done simply by searching through each word in each line of the lyrics and using SpaCy to check whether the part of speech is noun or verb. With these nouns and verbs as our keywords, we can find corresponding images for each keyword using the Unsplash Image API. We use this API to search for and download relevant images on Unsplash.com given each keyword. Figure 3.1 details an example of the use of Azlyrics, SpaCy, and Unsplash together.

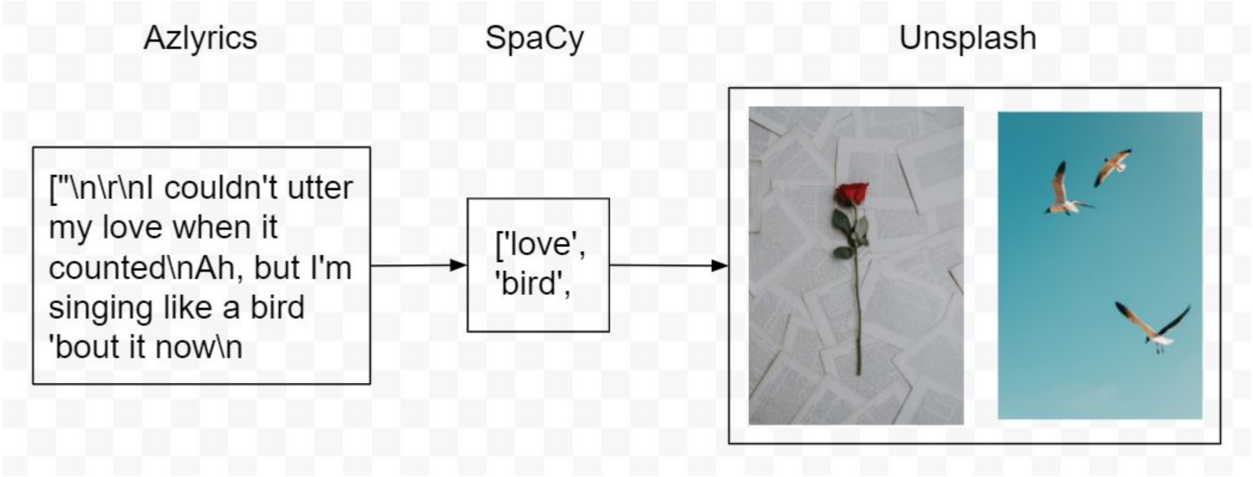

Azlyrics
SpaCy
Unsplash
["\n\r\nI couldn't utter my love when it counted\nAh, but I'm singing like a bird 'bout it now\n
['love', 'bird',

Figure 3.1: An example of the process for finding images relevant to the given song lyrics. We use Azlyrics to find the lyrics to “Shrike” by Hozier. The first two lines of the output are shown on the far left of the diagram. The nouns found in these lines using SpaCy are shown in the center and the images for these two nouns (found using Unsplash) are shown on the far right.

3.2. Creation of Content Image

The next step is to combine the images we found into a single content image. We use two alternative methods to combine these images.

3.2.1. Method 1

For the first method, the images are placed together over a large base image that acts as a sort of canvas, helping to fill in the gaps between the images. The base image is simply a random image selected from any of the images found using Unsplash. When placing the images on the base image, the location of each image is recorded; this is important for the next step as we do not want to obscure the subject of the image, but we do want to obscure the borders of each image in order to smooth the final content image. An example of very obvious image borders is shown in Figure 3.2. To obscure the image borders, we place smaller versions of the images on the original image borders, using these smaller images as “brushstrokes” to add variation to the content image. This process of recording image locations then shrinking and placing images is repeated using a for loop until the edges are sufficiently blurred, giving us our final content image. Three iterations through this loop after the initial placement of the images is sufficient.

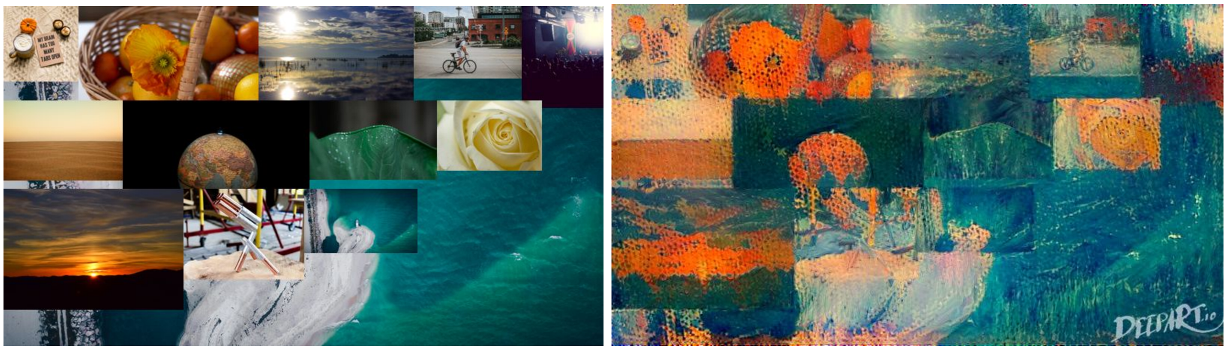


Figure 3.2: An example of a content image after the first iteration of image placement. On the left is the content image and on the right is the same image transformed with an abstract image using NAAS. The borders of the images even after the style transfer are very obvious.

3.2.2. Method 2

The second method uses a similar process to the first, except we place the smaller images randomly across the base image, obscuring both the content and the edges. After placing the initial set of images on the base image, we resize the images three times in total. At each size, we cycle through the folder of images 20 times and place each image. Cycling repeatedly through the images provides enough coverage to obscure the content. This method allows us to explore the impact that the actual content of the images has on the outcome of the final transformed image. A diagram of how the final content images are generated using the two methods is shown in Figure 3.3.

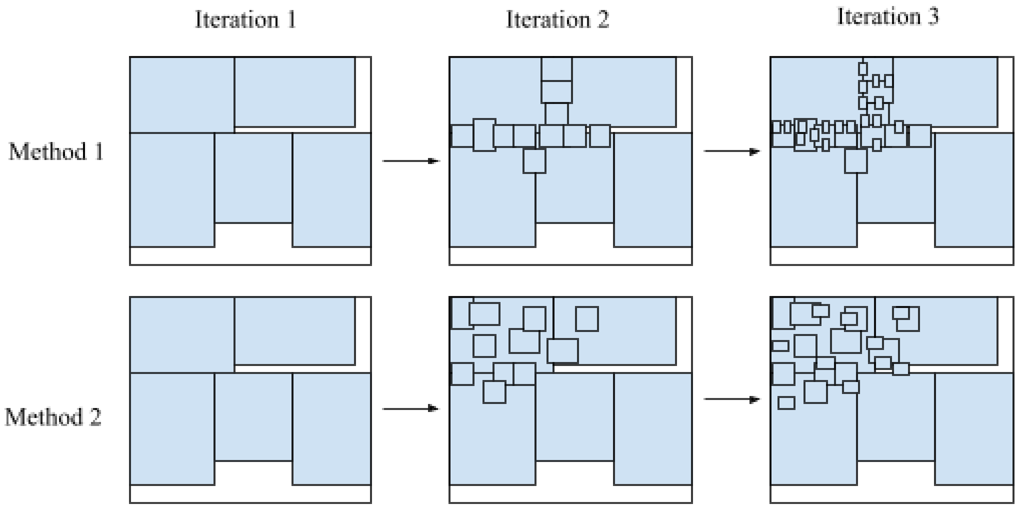


Figure 3.3: A diagram depicting the two methods used for combining images. Using Method 1, the subject is not obscured, whereas using Method 2, the subject is obscured. The diagram shows the process for only the top left quadrant, but the actual process is done for the whole image. The blue images are the images which are resized and placed on the white base image.

3.2.3. Hough Lines Method

Originally we intended to use the HoughLinesP function in OpenCV to find the image borders using a probabilistic Hough transform, but this method did not give good results as it did not find a sufficient number of the image boundaries. We experimented with a wide range of values for the arguments of minLineLength and maxLineGap, along with the values of the minVal and maxVal arguments used for Canny edge detection. Additionally, the algorithm identified too many lines within the individual images themselves, which was problematic for visualizations in which we did not want to obscure the subject. Figure 3.4 depicts an example of the results of the use of Canny edge detection along with the probabilistic Hough transform.

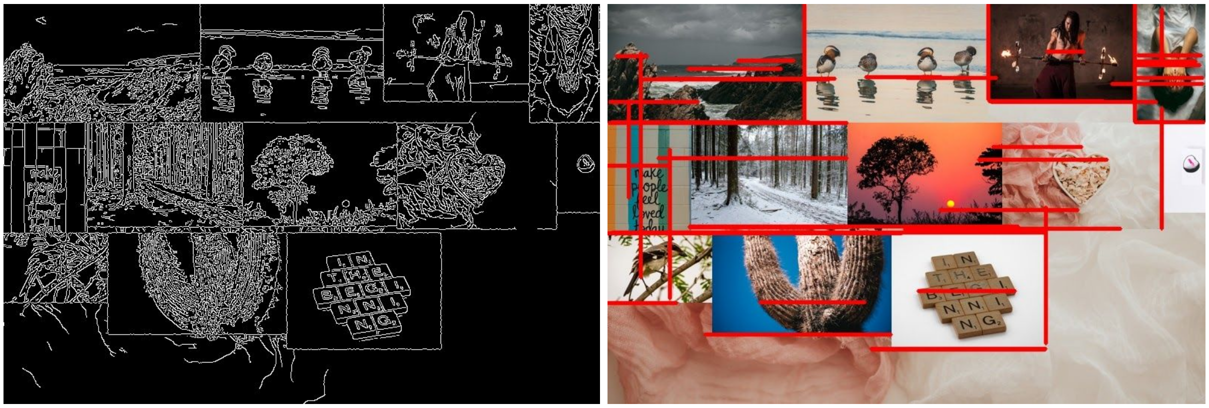
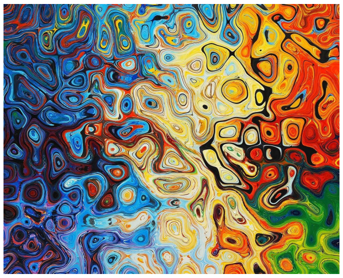
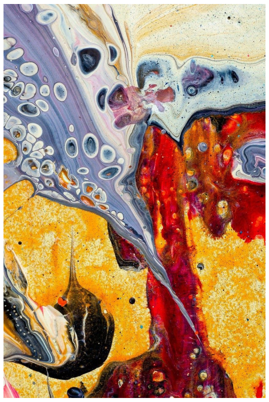


Figure 3.4: On the left is the result of OpenCV Canny edge detection on the image given. The lines determined using this image and the Hough transform are drawn in red on the original image on the right.

3.3. Obtaining the Style Image

Next, we must find a style image to transform with the content image. The first step for finding a style image is to classify the mood of the song for which we are creating a visualization. We achieve this through the use of Spotify audio features and a neural network, the code and data of which can be found here courtesy of Cristóbal Veas. The model is trained on a set of about 700 songs, with the predictors being the Spotify audio features of danceability, acousticness, energy, instrumentalness, liveness, valence, loudness, speechiness, and tempo, plus the length of the song. Each song is labelled with one of the four moods of happy, sad, energetic, or calm. The model itself consists of a 10 node input layer, an 8 node hidden layer with ReLu activation, and the final 4 node output layer with Softmax activation. This model reaches about 72% accuracy, most often confusing happy and energetic songs.

After determining the mood of the song, we next choose a random image from a small set of images corresponding to the mood, giving us the style image. There are 4 sets of images consisting of abstract images and paintings that invoke each of the 4 moods through visual qualities such as color or texture. These images are obtained through Unsplash and Pixabay. A description of the method for determining which mood corresponds to an image can be found in Figure 3.5.

Mood	Attributes	Example
Happy	● Colorful, textured ● More variation and brightness in color than energetic images	
Energetic	● Colorful, textured ● Less variation and brightness in color than happy images but more varied texture	

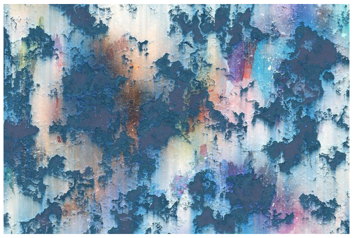
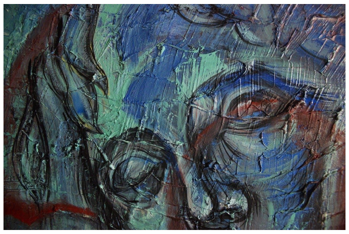
Calm	● Muted, less textured ● Smoother textures compared to sad images and brighter in color	
Sad	● Muted, less textured ● Lacking variation in color compared to calm images	

Figure 3.5: To determine the mood of an abstract image, we examine its color and texture. The reasoning behind the categorization of each image is given by the attributes of each mood. These attributes detail how some viewers, the author included, might ascribe a mood to a painting. Each example given exhibits the attributes of each mood.

3.4. Final Transformation Using NAAS

Our final step is to transform the content image with the style image using the NAAS algorithm. The code for this step is provided by Faris Nolan and can be found on Github. The process is rather straightforward, as we simply identify the content image and the style image and run the algorithm with these images, resulting in the final transformed image. The details of this algorithm can be reviewed in Section 2.5. As we did not have access to a more advanced GPU, we made use of Deepart.io in the creation of these transformed images to conserve resources. The site, originally created by Łukasz Kidziński and Michał Warchoł, uses the NAAS algorithm and generates these images in a fraction of the time.

3.5. Method Evaluation

In order to compare the two methods of content image generation, we developed a survey which asks the participant about the mood and content of 12 different music visualizations. Of these, 6 are generated using Method 1 and the other 6 are generated using Method 2. For each visualization, the participant is asked what the mood of the song might be given a choice of happy, sad, calm, or energetic. The participant is also asked to identify an object that they can see in the image given a choice of four different objects which change with each visualization. Only one mood and one object is correct. The full survey can be found in Appendix A.

The purpose of this survey is to examine which method of content image generation allows the mood and content to be more easily identified. While it is expected that the mood should be equally identifiable for each method, we predict that it should be nearly impossible to correctly identify an object in a visualization that was created using Method 2, as the goal of Method 2 is to create a completely abstract image. It should be easier to identify an object in Method 1 as this method is intended to keep the objects visible.

We also developed a second survey to compare the two methods. Two participants were asked to create a list of 10 songs that we then created music visualizations for. The participants were then asked to determine which visualization belonged to which song. The purpose of this survey is to determine if there is a correlation between which method was used to generate an image and how well the image evokes the theme and mood of the corresponding song. The survey can be found in Appendix B.

4. Results and Analysis

4.1. Content Image Method 1

Here we examine the first method of generating content images. This method takes the images found from keywords in a song's lyrics and aims to produce a content image where the subject of the images are less obscured. We use the song "Vogue" by Madonna as an example. Figure 4.1 depicts the first two iterations through the loop that places and resizes the images. The final two iterations are shown in Figure 4.2. As shown by the second image in Figure 4.2 which is the completed content image, placing resized images over the borders creates a more varied texture while preserving some of the content. Also note that the lines are not always placed directly over the image borders; this was initially not intentional, however we found the result to be visually pleasing due to the randomness it provides.

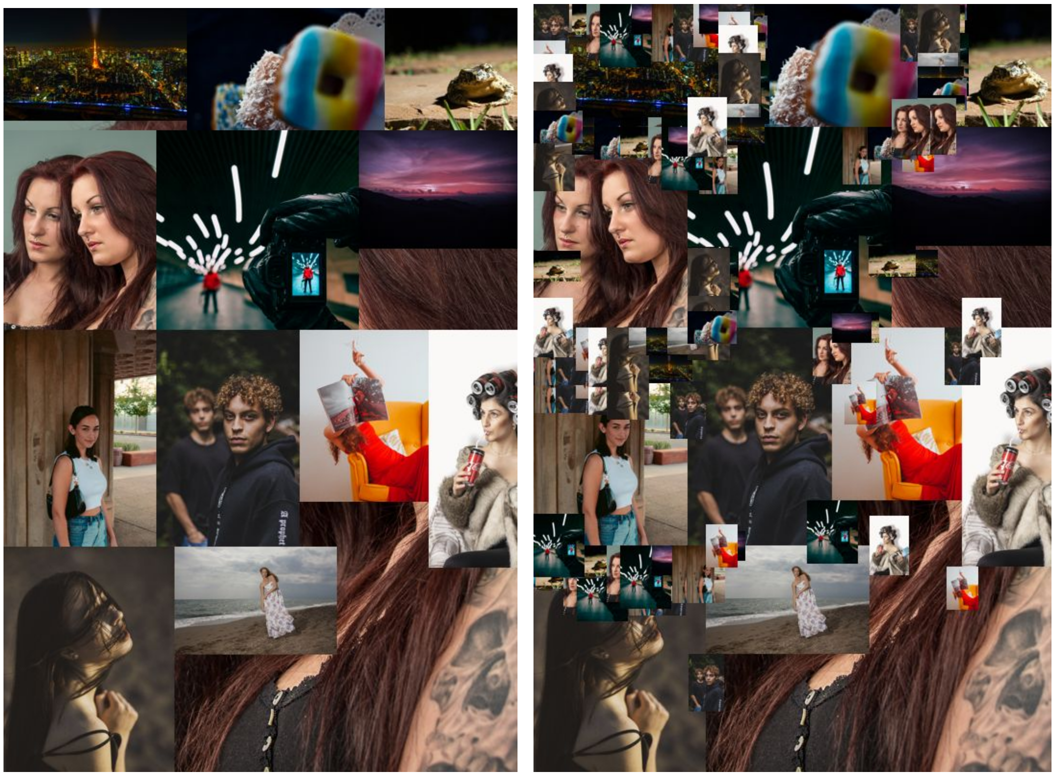


Figure 4.1: First and second iteration of content image.

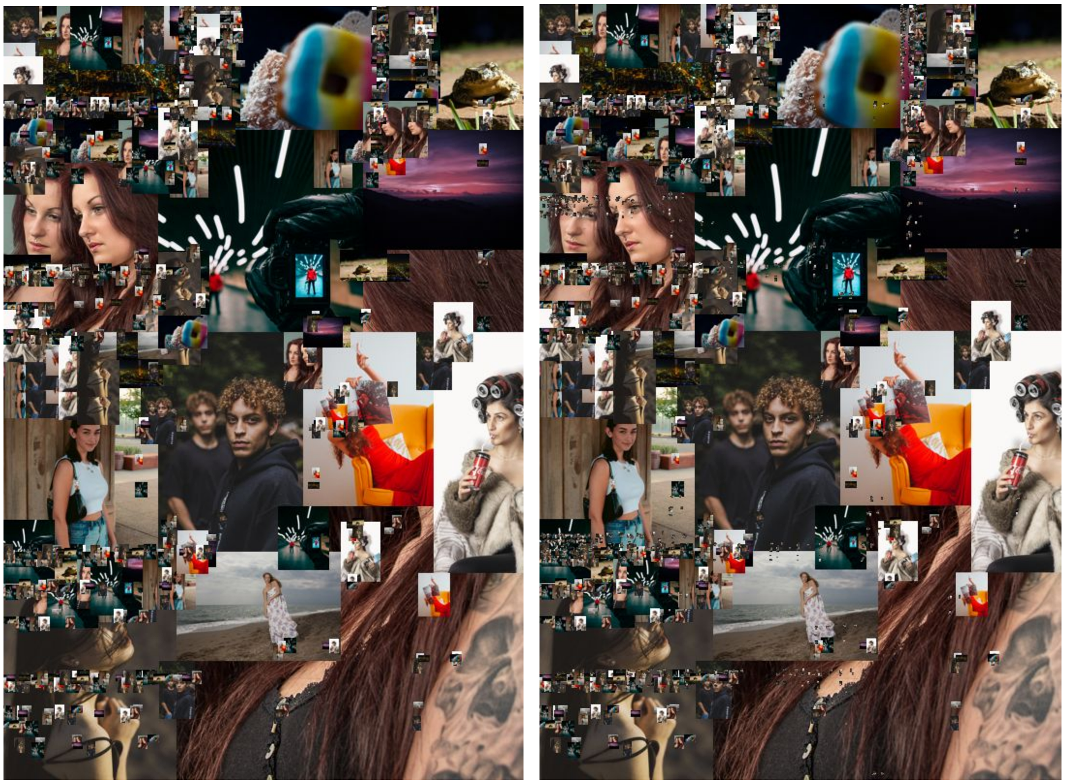


Figure 4.2: Third and fourth/final iteration of content image.

The images in Figure 4.3 depict the transformation using the NAAS algorithm with the final content image and a style image as input. The style image was obtained using the process described in Section 3.3. As shown by the images, much of the content is preserved while the colors and textures much more closely resemble those of the style image. Many of the smallest images that were present in the content image are not as clearly visible in the transformed image. The variance in color and texture provided by these smaller images creates the desired blending effect throughout the image and especially at image borders. Figure 4.4 contains 11 more example images generated using Method 1. These images are all featured in the two method evaluation quizzes found in Appendix A and Appendix B.

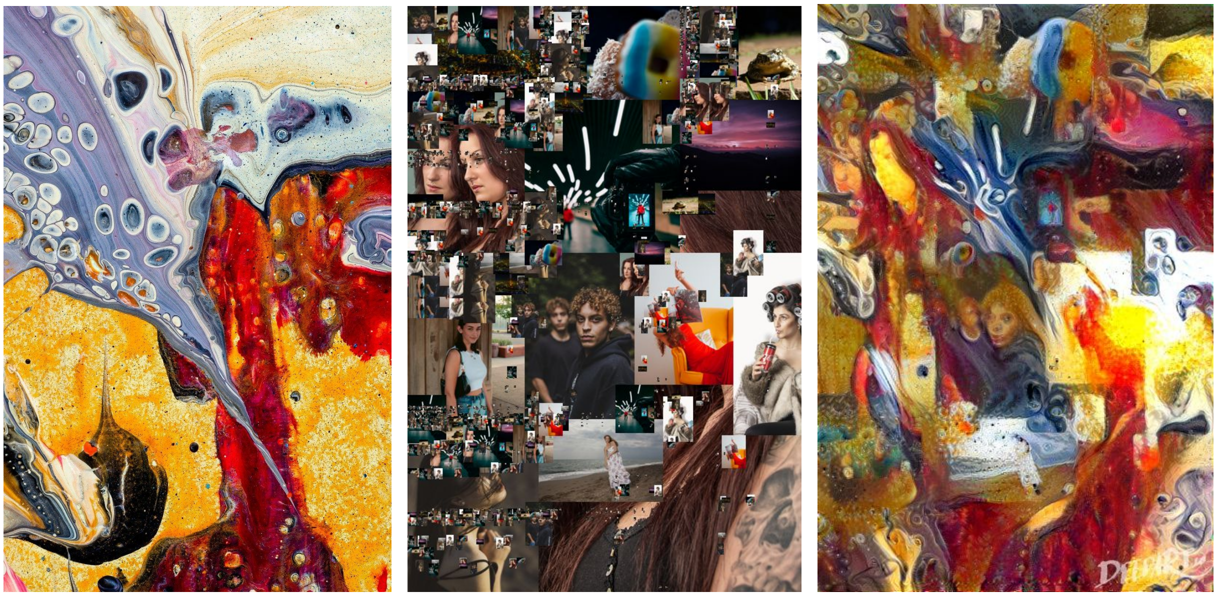


Figure 4.3: From left to right we have the style image, the content image, and the final image.

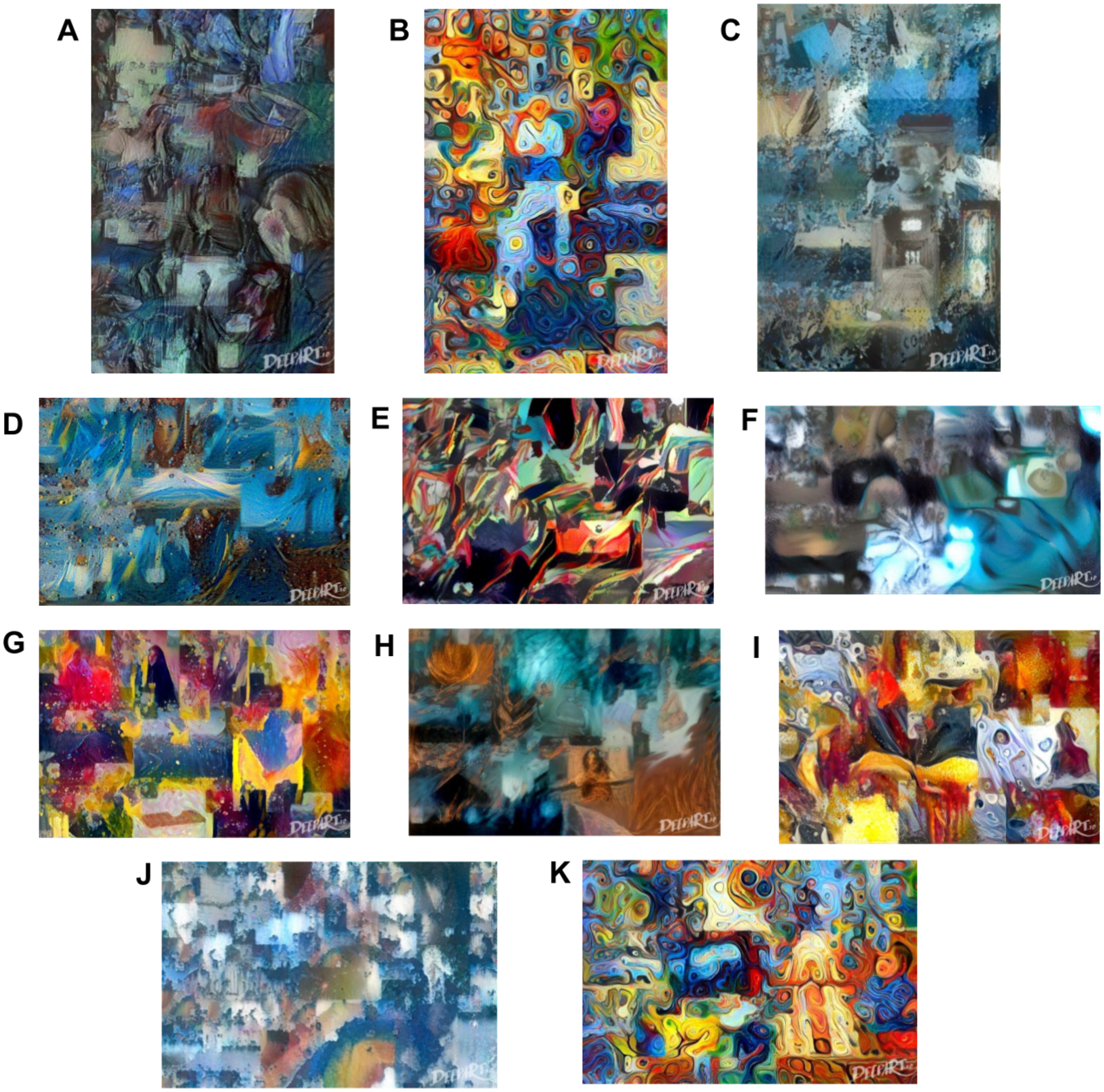


Figure 4.4: All music visualizations where the content image was generated using Method 1. The corresponding songs and song artists along with a link to the song are as follows: A. Shrike (Hozier); B. YMCA (The Village People); C. Take Me Home Country Roads (John Denver); D. Suspirium (Thom Yorke); E. I Know the End (Phoebe Bridgers); F. Mary (Big Thief); G. Panoramic Girl (Young the Giant); H. Stairway to Heaven (Led Zeppelin); I. Vogue (Madonna); J. Somewhere Over the Rainbow (Israel Kamakawiwo'ole); K. Youth (Glass Animals).

4.2. Content Image Method 2

Here we examine the second method of generating content images. This method takes the images found from keywords in a song's lyrics and aims to produce a content image where the subject of the images is completely obscured, resulting in a completely abstract final image. We again use the song “Vogue” by Madonna as an example. Figure 4.5 depicts the first two iterations through the loop that places and resizes the images. The final two iterations are shown in Figure 4.6, with the final content image on the right in the figure. As shown by these images, placing resized images at random obscures the content.

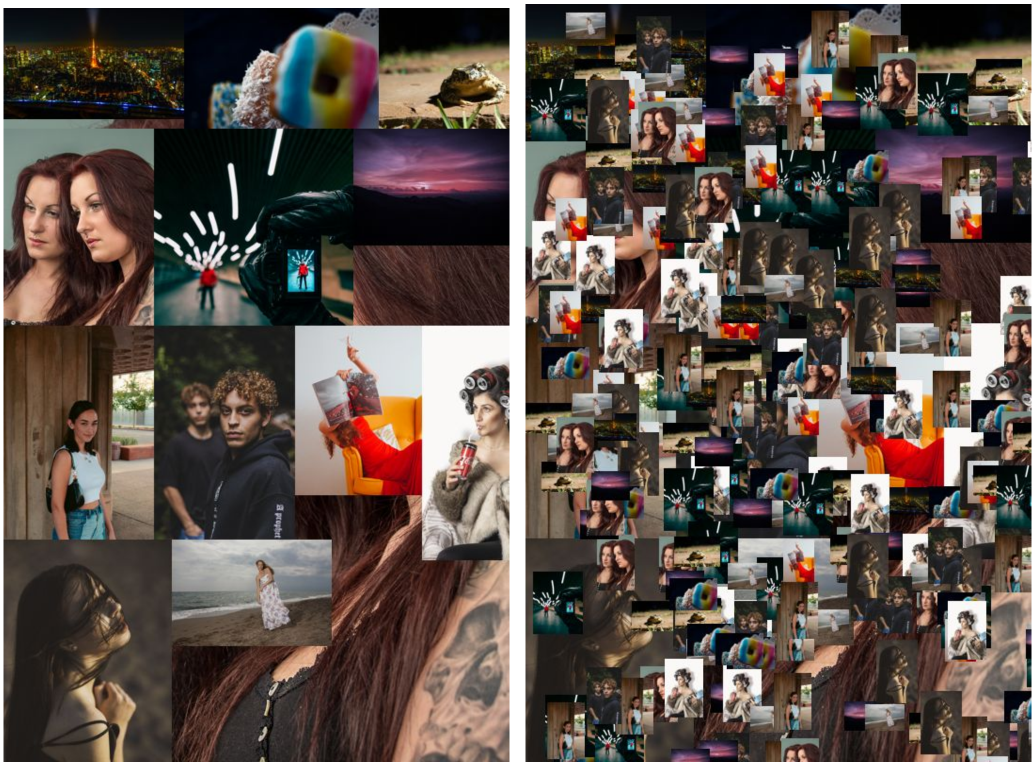


Figure 4.5: First and second iteration of content image.

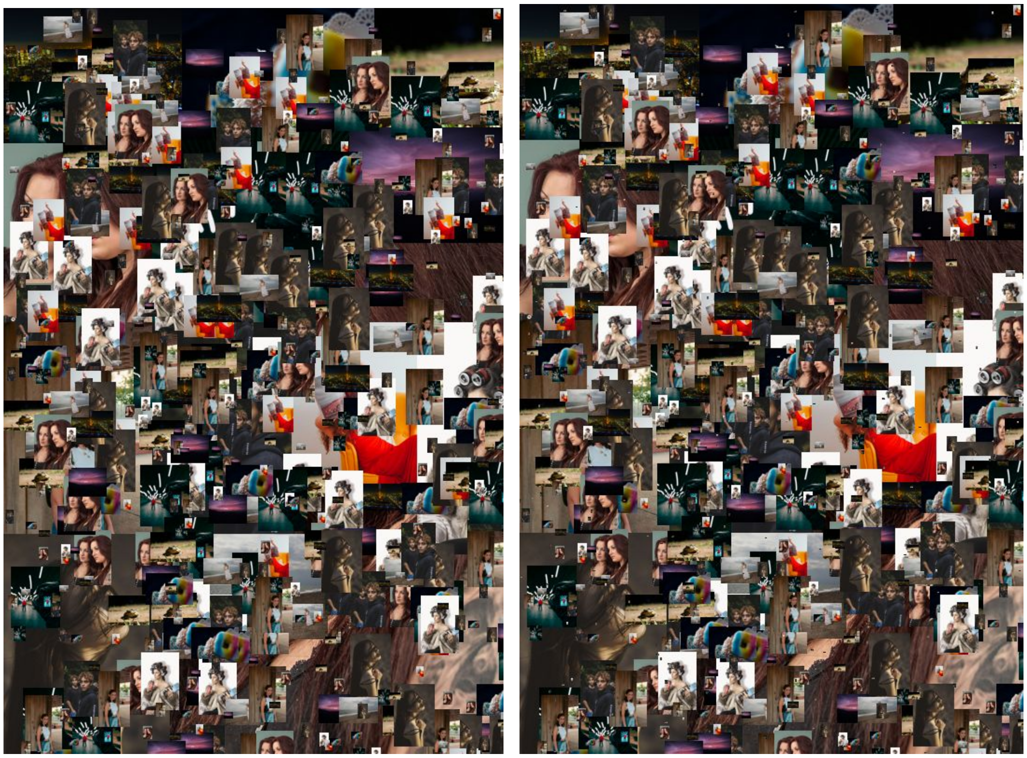


Figure 4.6: Third and fourth/final iteration of content image.

The images in Figure 4.7 depict the transformation using the NAAS algorithm with the final content image and a style image as input. As shown by the images, the content of the images is obscured and the colors and textures much more closely resemble those of the style image. The variance in color and texture provided by these images creates the desired blending effect throughout the image. Figure 4.8 contains 11 more images generated using Method 2. These images are all featured in the two method evaluation quizzes found in Appendix A and Appendix B.

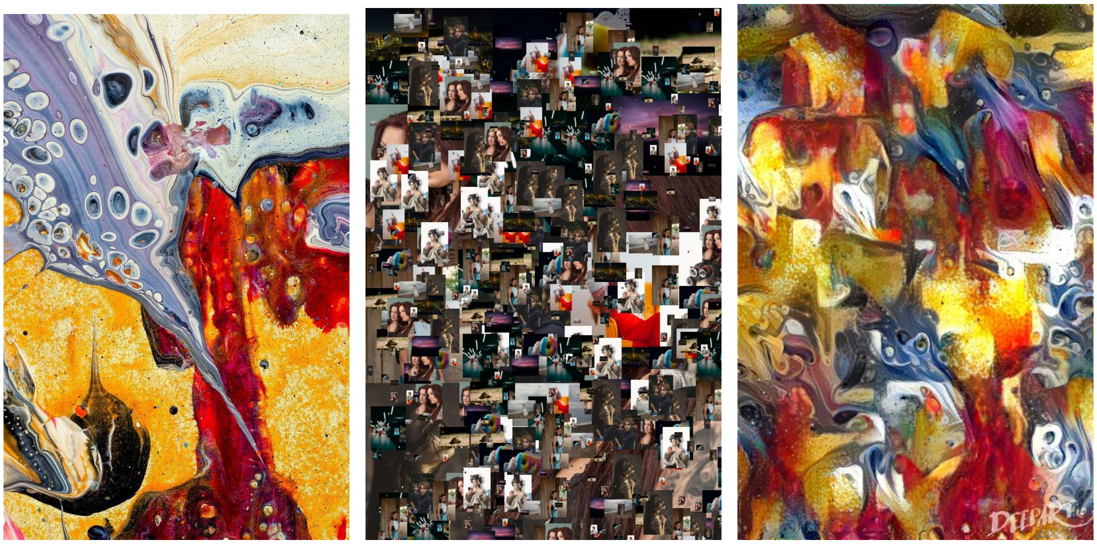


Figure 4.7: From left to right we have the style image, the content image, and the final image.

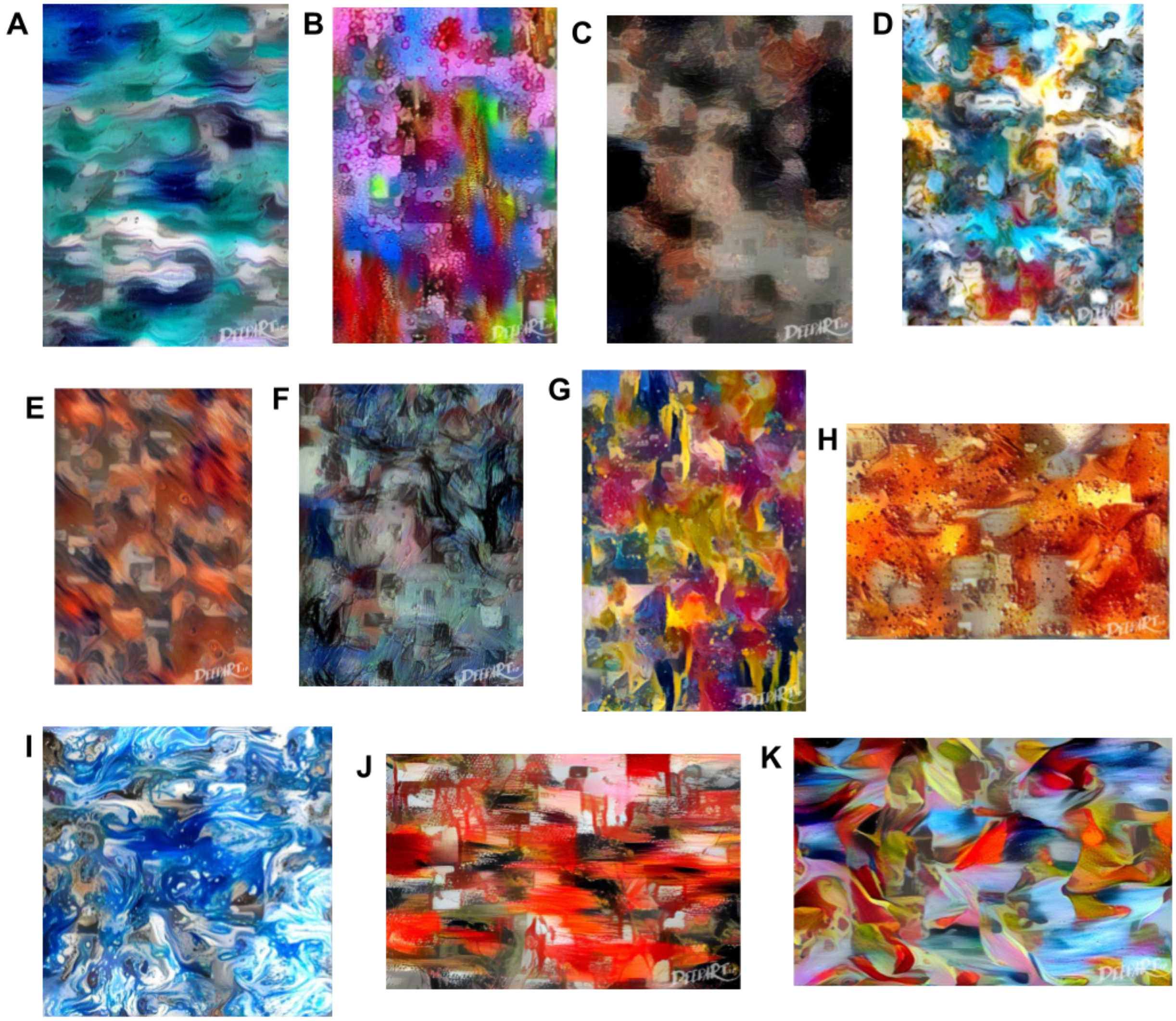


Figure 4.8: All music visualizations where the content image was generated using Method 2. The corresponding songs and song artists along with a link to the song are as follows: A. Weird Fishes/Arpeggi (Radiohead); B. 10000 Emerald Pools (Børns); C. Nica Libres at Dusk (Ben Howard); D. Phase (Pinegrove); E. Between the Bars (Elliott Smith); F. November Rain (Guns N' Roses); G. Sweet Caroline (Neil Diamond); H. Rivers of Babylon (Boney M.); I. Hello (Adele); J. Run Boy Run (Woodkid); K. Cheeseburger in Paradise (Jimmy Buffett).

4.3. Effect Of Style Image

Here we examine how differences in the style image, specifically the variation in color and texture, affects the output image. We achieve this by transforming a content image with increasingly textured style images as shown in Figure 4.9. It appears that with increasing variation in color and texture, the more closely the content of the output image matches that of the input content image.

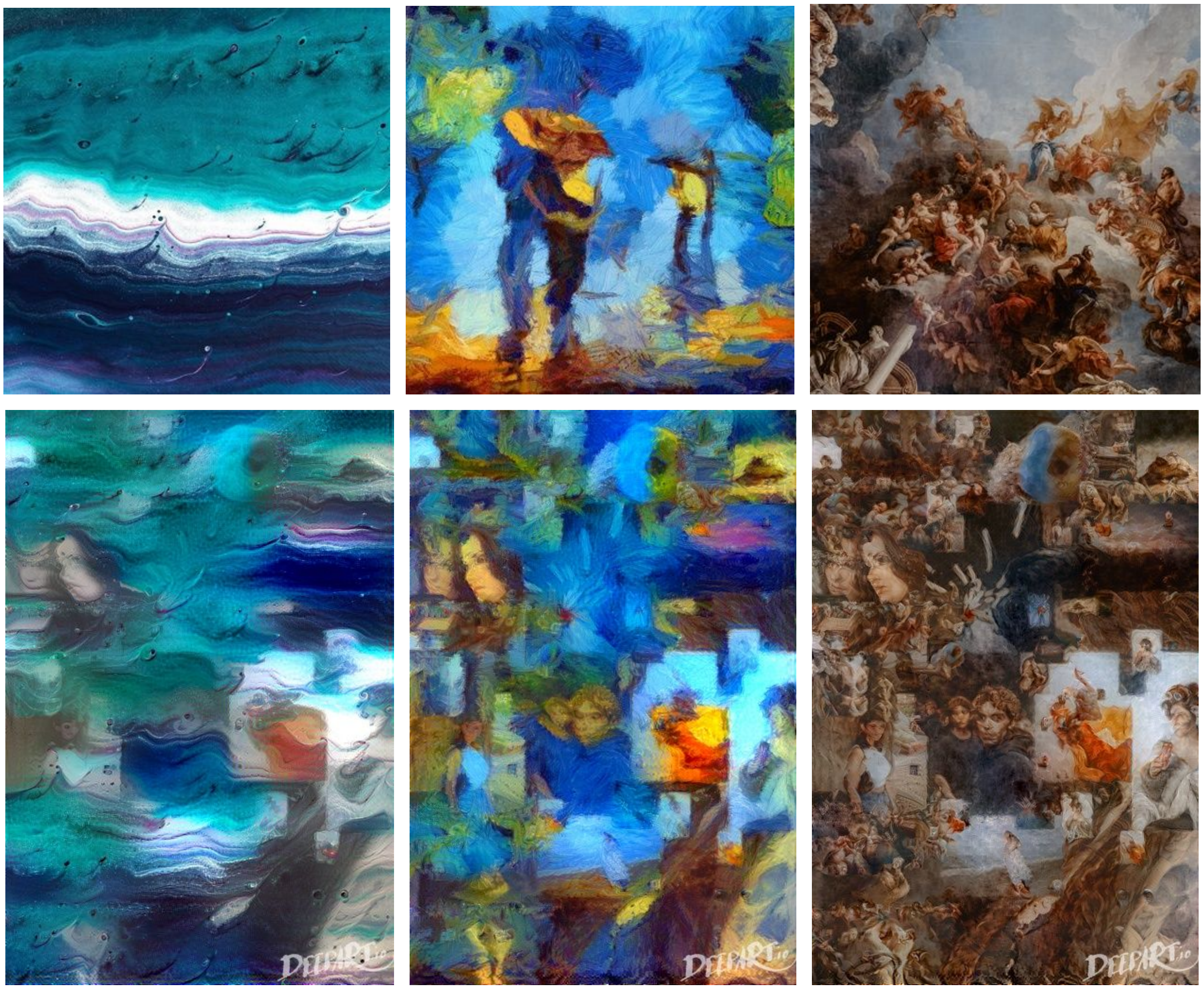


Figure 4.9: The top row depicts three style images with increasing variation in texture and color from left to right. The bottom row contains the corresponding output image after transformation with the style image and content image. The content image, though not shown here, is shown previously in Figure 4.3 and was generated using Method 1.

4.4. Method Evaluation Surveys

4.4.1. Survey A Description

We used two different surveys to compare the effectiveness of each method in conveying the mood and theme of a song. The first survey, which can be found in Appendix A, asked participants to view 12 different images and answer what they believe the mood of the corresponding song was from a choice of happy, sad, energetic, or calm. It also asked participants to decide which word appears in the song lyrics based on the image from a choice of four different words. Half of the 12 images were generated using Method 1 and the other half with Method 2. Additionally, there are three images for each of the four moods. Two subjects participated in the first survey. Their responses are recorded in Figure 4.10, along with the details and answers for each question. The qualitative observations of each participant about the survey can be found in Figure 4.11.

4.4.2. Survey A Results

Question	Song/Artist	Method	Answers	Response 1	Response 2
1	Shrike/Hozier	1	sad, bird	sad, bird	sad, bird
2	Weird Fishes Arpeggi/Radiohead	2	calm, fish	calm, river	calm, river
3	Youth/Glass Animals	1	happy, boy	sad, boy	energetic, boy
4	Suspirium/Thom Yorke	1	calm, body	sad, painting	happy, town
5	Run Boy Run/Woodkid	2	energetic, world	energetic, forest	sad, mountain
6	Mary/Big Thief	1	sad, planet	calm, dog	calm, stone
7	Phase/Pinegrove	2	energetic, horizon	happy, horizon	happy, horizon
8	10,000 Emerald Pools/Børns	2	happy, pool	happy, cloud	happy, grass
9	Between the Bars/Elliott Smith	2	calm, promise	energetic, love	energetic, ocean
10	Panoramic Girl/Young the Giant	1	happy, girl	happy, girl	energetic, girl
11	I Know the End/Phoebe Bridgers	1	energetic, promise	energetic, horse	energetic, sunset
12	Nica Libres at Dusk/Ben Howard	2	sad, dream	calm, love	sad, dream

Figure 4.10: A table depicting the results of the first method evaluation survey, along with the song and artist corresponding to the image, the method used to generate the image, the solutions to each question, and the responses of the participants.

Participant 1 Observations	Participant 2 Observations

● All images were visually pleasing, the task itself was enjoyable ● Unsure of the answer for many of them ● Unsure of what to look for in the image for words such as “dream”	● Images were visually pleasing ● Often difficult to identify an object

Figure 4.11: Qualitative observations by the participants about the survey.

Using Figure 4.10, we can examine which method more easily allowed the mood and theme of a song to be identified. Both participants were correct about 10/24 of the survey questions. The participants got a total of 5/12 moods correct and 6/12 themes correct for Method 1; both of these values are greater than the baseline of 3/12 corresponding to random guessing. For Method 2, the participants got a total of 6/12 moods correct and 3/12 themes correct. Only the number of correct moods is greater than the baseline. These results suggest that Method 1 more easily allows the theme of the song to be inferred whereas Method 2 is slightly better at conveying the mood. A larger experiment with more participants would be needed to confirm this result.

4.4.3. Survey B Description

The second survey asked the two participants to identify what they believed the corresponding music visualization for a song was given a list of ten songs that they had chosen, along with the ten music visualizations generated for those songs. The full survey can be found in Appendix B, and the responses of the participants are listed in Figure 4.12.

4.4.4. Survey B Results

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
P1	E	H	F	I	J	C	G	B	A	D
P2	B	H	I	G	D	C	J	E	A	F
A	B	H	C	G	D	A	E	F	J	I

Figure 4.12: A table showing the responses of the participants of the second survey, where the first row contains the question number, the second and third rows contain the responses of the two participants, and the fourth row contains the answers. The letters correspond to a music visualization, and the question numbers correspond to a song.

Using Figure 4.12, we can see that the first participant got 1/10 questions correct and the second participant got 4/10 questions correct. To determine if these values are significant, we can compare them to the probability that a participant might get a certain number of questions correct simply through random guessing. To determine these probabilities, we can run a simulation where we have 10 objects in a specific order; we then shuffle these objects randomly for many trials and count the number of times that X objects are in the correct order, where X is a number between 0 and 10. We do this for 10000 trials for each value of X. The graph depicted in Figure 4.13 shows the percentage of simulation trials where X number of objects were in the correct order.

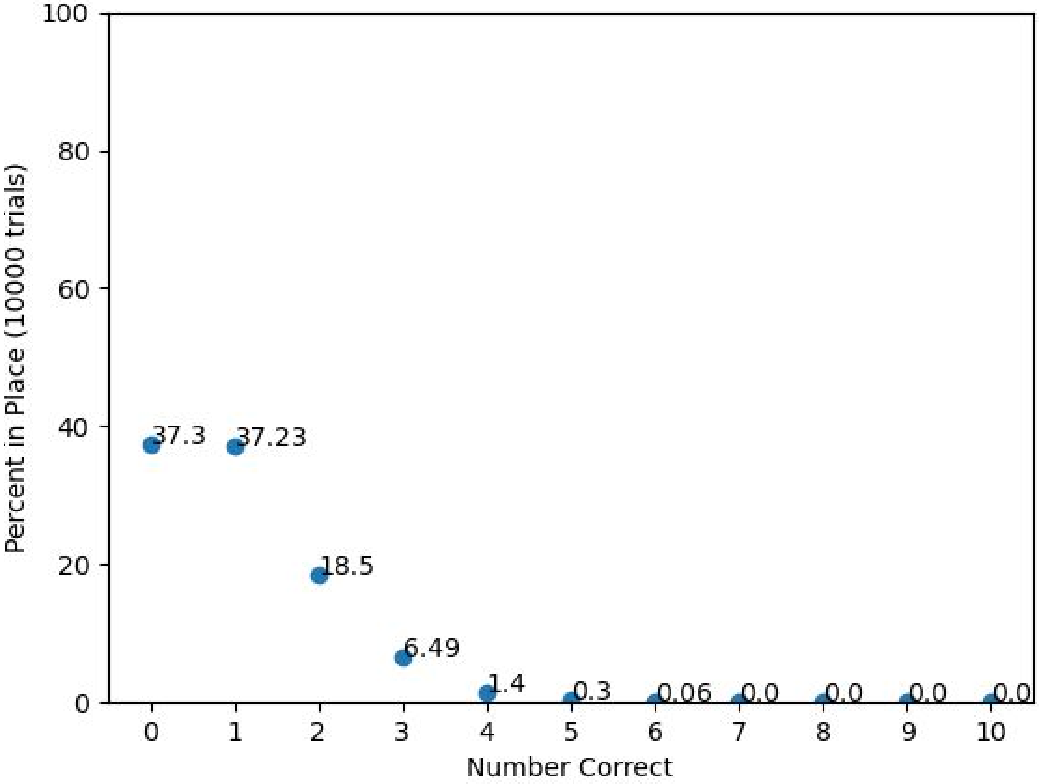


Figure 4.13: A graph depicting the percentage of simulation trials for which some number of objects were in the correct order.

As shown by Figure 4.13, there was a 37.3% chance that the first participant was able to get 1/10 questions correct simply by guessing. There was a 1.4% chance that the second participant was able to get4/10 questions correct by guessing. While the results for the first participant do not strongly indicate that a viewer can identify the connection between a song and a music visualization, the results for the second participant do seem to indicate this.

Some factors that may have affected the ability of the participants to succeed in linking the correct music visualization with a song include the fact that the participants may not have heard the song before and simply made a guess based on the title. An example of this may be found in question 6 of the survey, which featured the song “Rivers of Babylon” by Boney M. Though the mood of the song is energetic and therefore the corresponding music visualization is bright in color, both participants chose option C as what they believed the song’s visualization to be. Option C is blue and white in color and has a smooth texture. It corresponds to the song "Hello" by Adele, which is a sad song. It is possible that the participants chose the incorrect option C simply because the word "river" is a part of the title of the song "Rivers of Babylon", and figured that the corresponding visualization might be blue.

4.5. Runtime of Algorithm

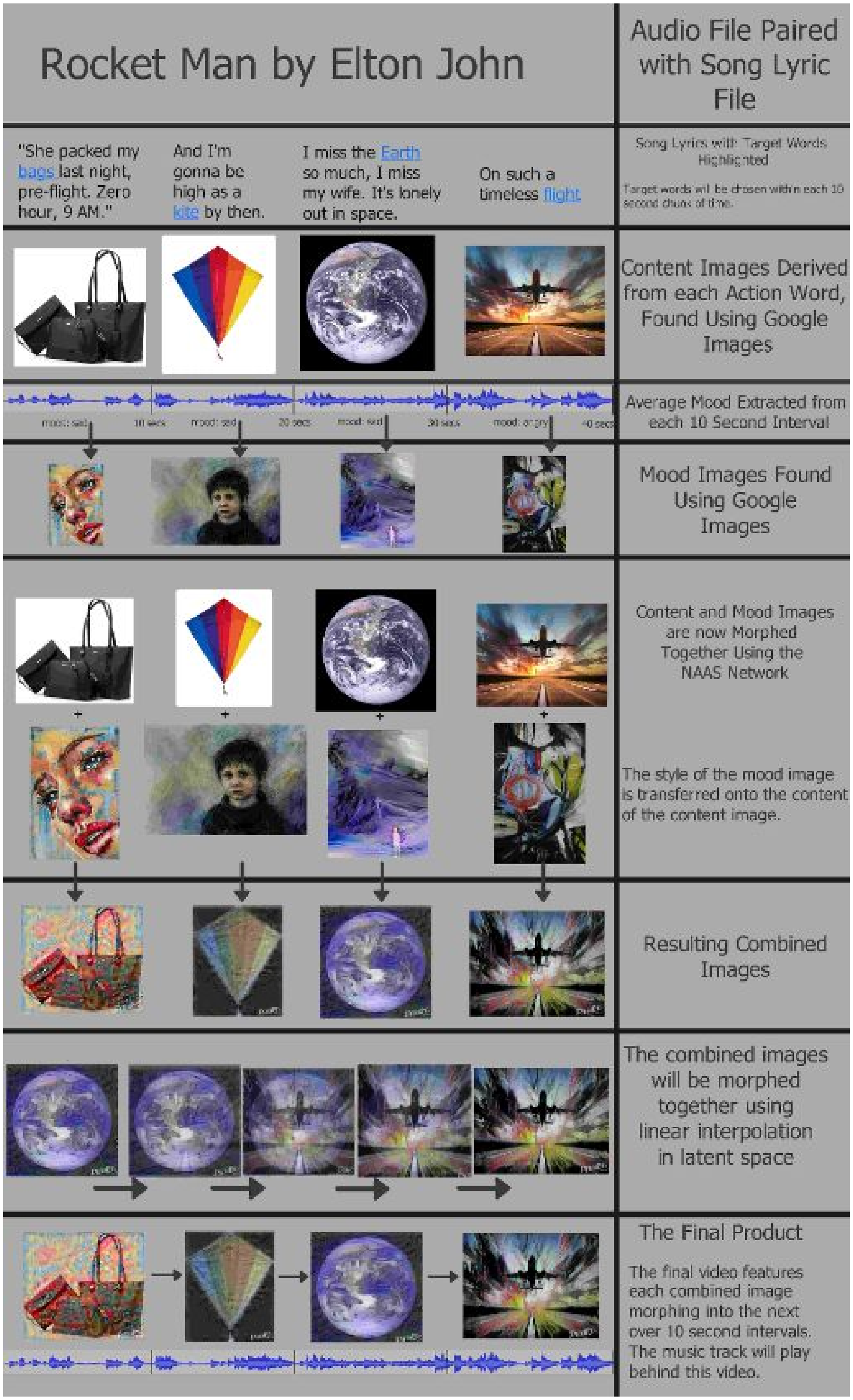
Due to the differences in the methods used to generate the content images, the runtimes of each method are quite different. Once the images that make up the content image are obtained using the Unsplash API, we time how long it takes for each method to generate the content image. While Method 1 takes about 8.5 seconds on average, Method 2 takes only about 1.5 seconds (using a machine with an AMD Ryzen™ 5 4500U processor, 2.3GHz base clock speed, 8GB RAM). This is because Method 2 does not keep track of the image boundaries; it simply resizes the images and places them at random. Method 1 must keep track of 4 lines for every image it places in order to represent the borders of each image. The number of lines it must keep track of increases exponentially as the image size decreases, as the number of images that must be placed increases due to the small size of the images being placed.

5. Conclusion

Our goal was to explore the idea of AI generated art by creating visualizations of music. We developed a system to generate abstract music visualizations comparable in aesthetic to those generated by a human artist. We used these music visualizations to compare AI art to human-created art and to determine the effectiveness in conveying mood and theme. We found that AI can be used to convey the mood and theme of a song with some success through abstract images.

Future extensions of this project would include further examination into the ways in which AI generated music visualizations compare to those generated by human artists such as the aforementioned Melissa McCracken. We might evaluate this by creating a site like Botpoet.com, which asks users to determine whether a poem is AI-generated or created by a human, though instead we would compare human and AI-generated art pieces. This would allow us to determine if humans can detect the difference between the two. Other questions that we could explore include how we might make the AI's process of generating art more similar to a human's in order to further examine the creative process of AI. One way we might do this would be to further develop the method of generating the style for a piece, such as increasing the number of possible moods or using audio features from a song's audio file itself to add variation and nuance to the visualizations. A project that uses this latter method to add randomness to the output is Xander Streenbrugge's recently developed Wzrd.ai, a site that uses AI to generate a music video given an audio file. The diagram for a similar technique of generating music videos can be found in Figure 5.1.

This project was a great starting point for my interest in AI and machine learning, and I was pleased to be able to include some of my other interests such as art and music. I was able to explore topics in machine learning that I might not otherwise have had the chance to. I do not regret that I decided to find a project topic on my own instead of joining another group, though this choice came with plenty of challenges. Completing this project without group members gave me the confidence that I could manage my time and my goals well enough to complete a large scale project like this. I also very much enjoyed being able to choose a topic that I was passionate about, research it exhaustively, and decide myself where I would like to take it.



Rocket Man by Elton John

Audio File Paired with Song Lyric File

"She packed my bags last night, pre-flight. Zero hour, 9 AM."

And I'm gonna be high as a kite by then.

I miss the Earth so much, I miss my wife. It's lonely out in space.

On such a timeless flight

Song Lyrics with Target Words Highlighted

Target words will be chosen within each 10 second chunk of time.

Content Images Derived from each Action Word, Found Using Google Images

Average Mood Extracted from each 10 Second Interval

Mood Images Found Using Google Images

Content and Mood Images are now Morphed Together Using the NAAS Network

The style of the mood image is transferred onto the content of the content image.

Resulting Combined Images

The combined images will be morphed together using linear interpolation in latent space

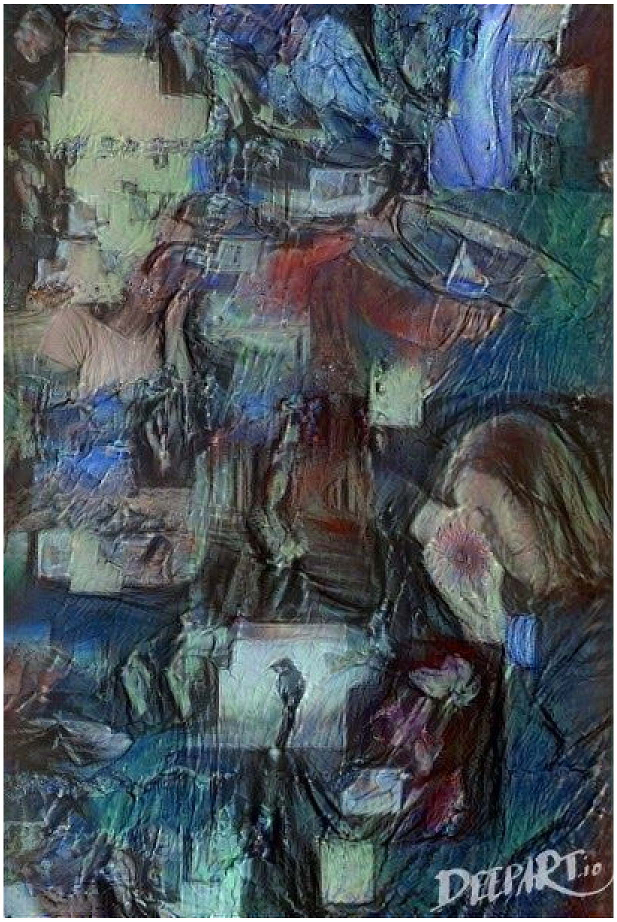
The Final Product

The final video features each combined image morphing into the next over 10 second intervals. The music track will play behind this video.

Figure 5.1: A technique for generating music videos from an audio file, using the example of "Rocket Man" by Elton John.

Appendix A: Method Evaluation Survey 1

Question Block 1



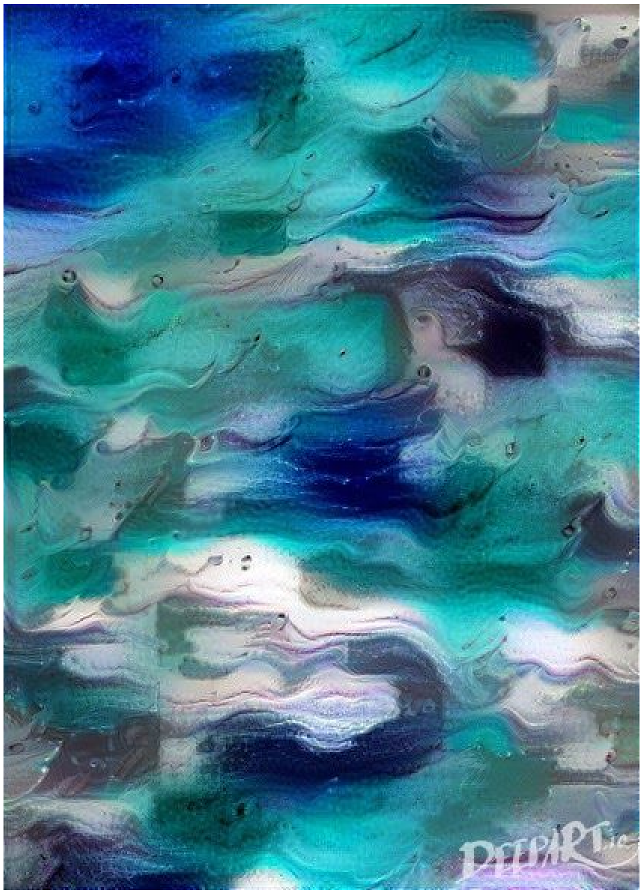
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. House
B. Ocean
C. Tree
D. Bird

Question Block 2



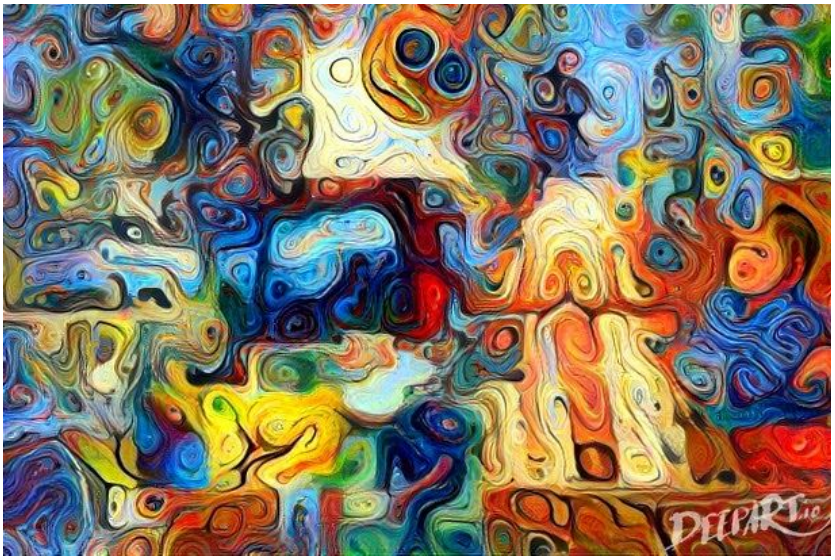
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Fish
B. Sun
C. Tree
D. River

Question Block 3



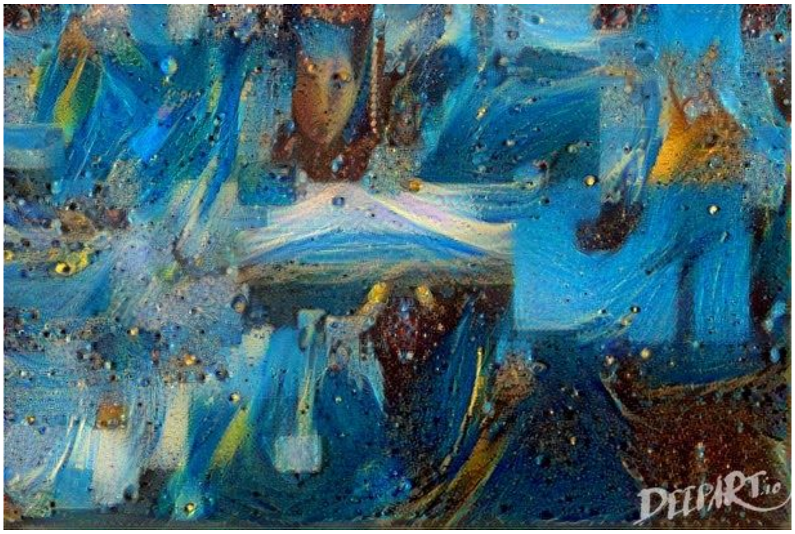
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Beach
B. Boy
C. City
D. Boat

Question Block 4



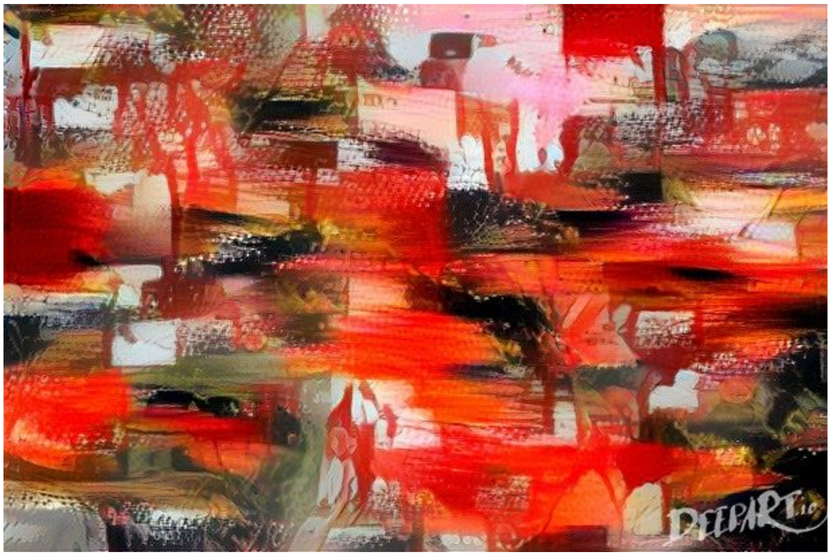
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Town
B. Painting
C. Body
D. Book

Question Block 5



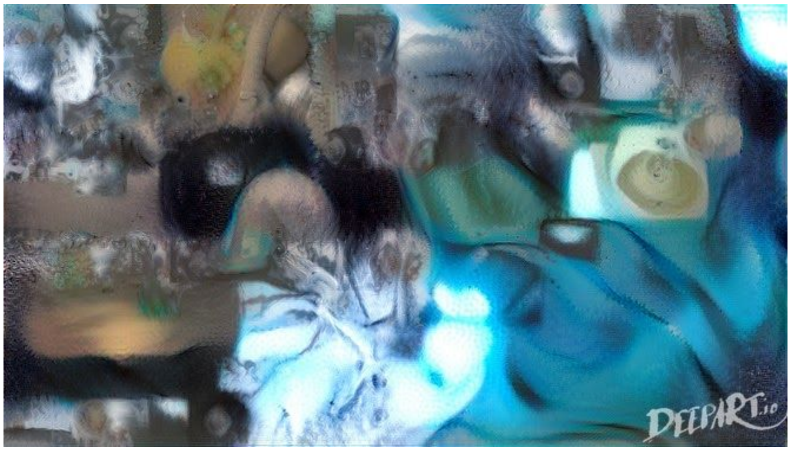
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. World
B. Forest
C. Mountain
D. People

Question Block 6



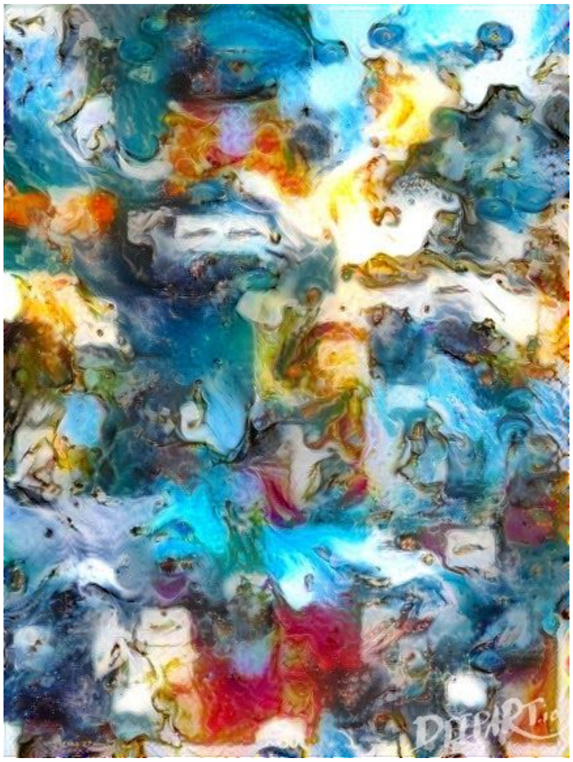
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Stone
B. Planet
C. Man
D. Dog

Question Block 7



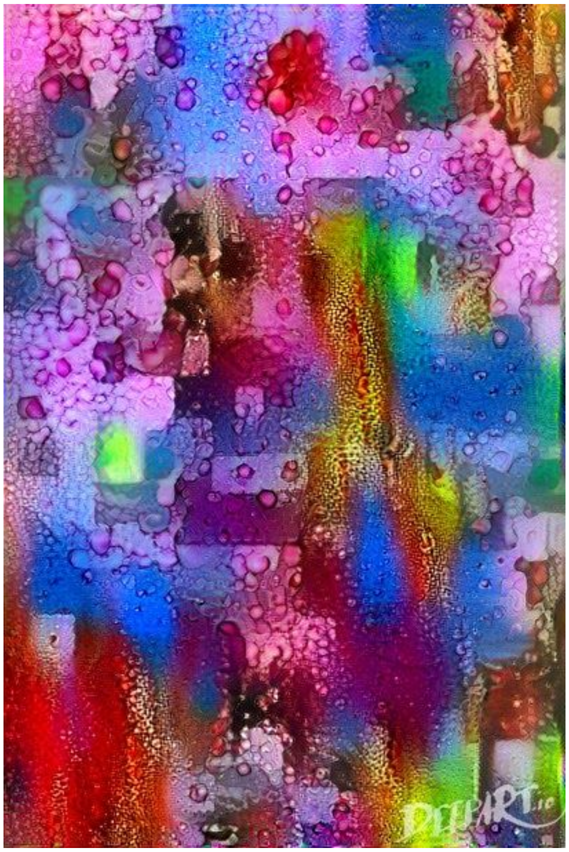
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Horse
B. Coast
C. Apartment
D. Horizon

Question Block 8



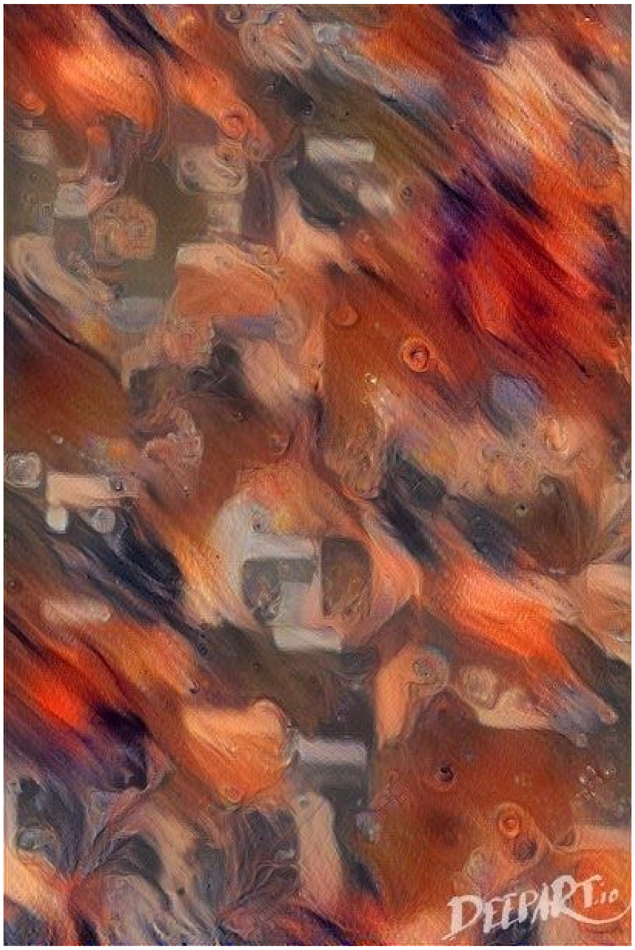
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Woman
B. Pool
C. Cloud
D. Grass

Question Block 9



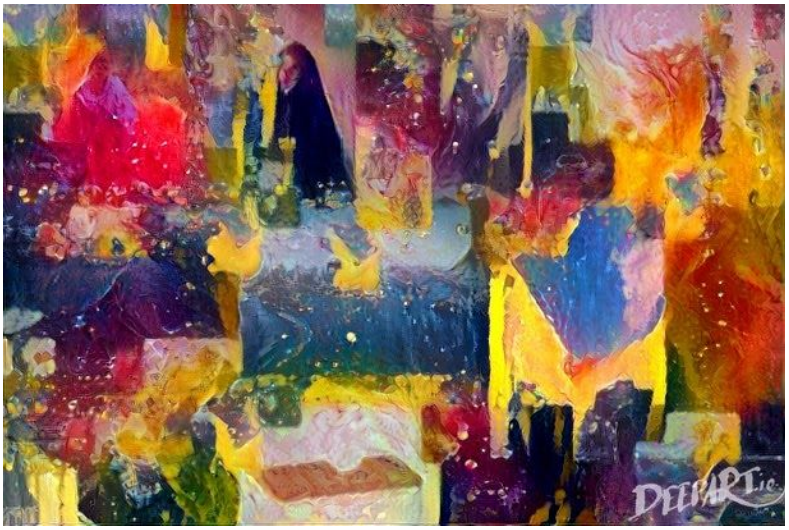
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. House
B. Ocean
C. Love
D. Promise

Question Block 10



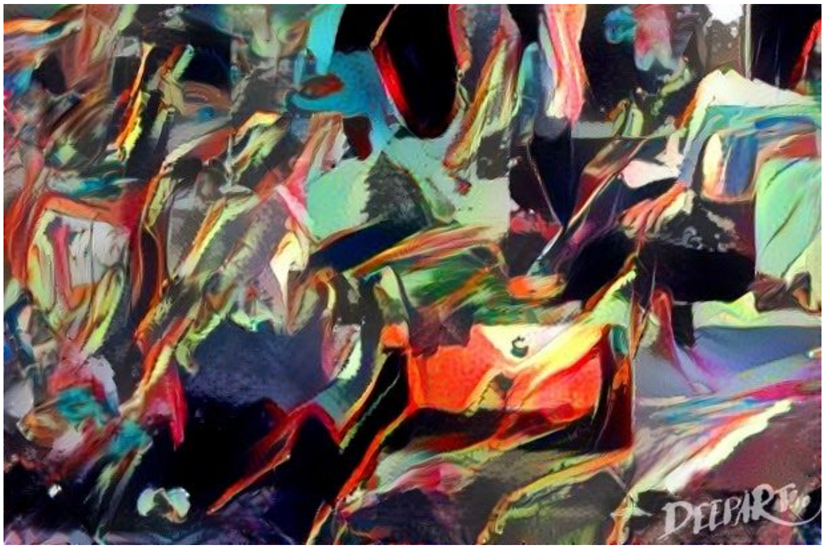
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Shirt
B. Girl
C. Plane
D. Sun

Question Block 11



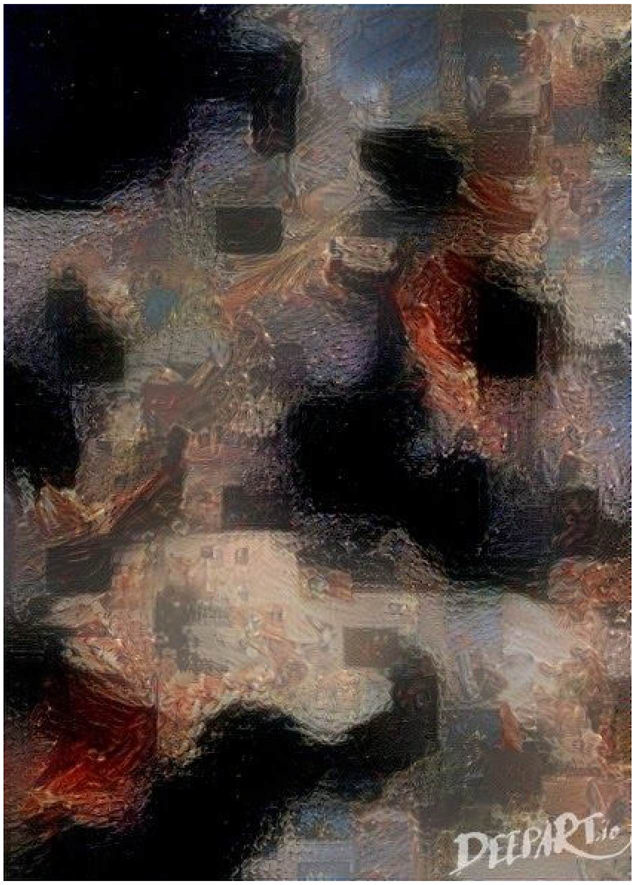
Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

A. Blanket
B. Water
C. Horse
D. Sunset

Question Block 12



Based on this image, what do you think the mood of the song is?

A. Happy
B. Sad
C. Energetic
D. Calm

Which of the following do you think appear in the song?

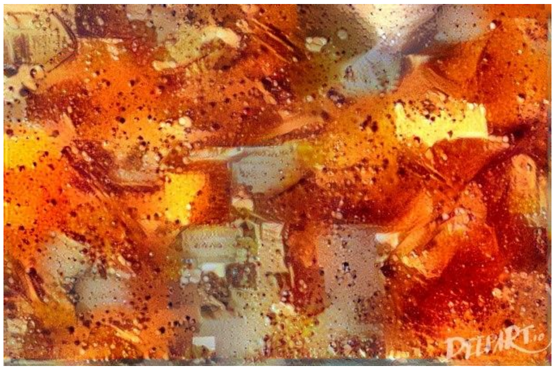
A. Town
B. Wall
C. Dream
D. Love

Appendix B: Method Evaluation Survey 2

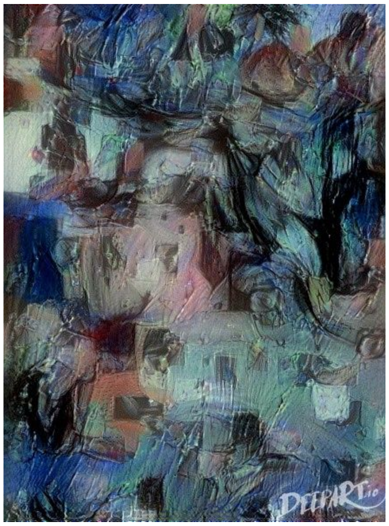
Try to figure out which image belongs to which song! Each image has a letter next to it to let you more easily indicate your choice.

Song list (random order):

1. November Rain (Guns N' Roses)
2. Take Me Home Country Roads (John Denver)
3. Hello (Adele)
4. Sweet Caroline (Neil Diamond)
5. YMCA (Village People)
6. Rivers of Babylon (Boney M.)
7. Somewhere Over the Rainbow (Israel Kamakawiwo'ole)
8. Stairway to Heaven (Led Zeppelin)
9. Cheeseburger in Paradise (Jimmy Buffet)
10. Vogue (Madonna)



A



B

C

D

E

F

G

H

I

J

References

Canny Edge Detection. (n.d.). Retrieved February 04, 2021, from https://docs.opencv.org/master/da/d22/tutorial_py_canny.html

Cohn, G. (2018, October 25). AI Art at Christie's Sells for $432,500. Retrieved November 25, 2020, from https://www.nytimes.com/2018/10/25/arts/design/ai-art-sold-christies.html

Gatys, L., Ecker, A., &; Bethge, M. (2015, September 02). A Neural Algorithm of Artistic Style. Retrieved December 05, 2020, from https://arxiv.org/abs/1508.06576

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014, June 10). Generative Adversarial Networks. Retrieved November 16, 2020, from https://arxiv.org/abs/1406.2661

Horrigan, A. (2018, March 19). Azlyrics. Retrieved March 17, 2021, from https://pypi.org/project/azlyrics/

Hough Line Transform. (n.d.). Retrieved February 04, 2021, from https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html

Jones, K. (2017, June 19). GANGogh: Creating Art with GANs. Retrieved November 17, 2020, from https://towardsdatascience.com/gangogh-creating-art-with-gans-8d087d8f74a1

Kurt, D. (2018, April 24). Artistic Creativity in Artificial Intelligence. Retrieved November 16, 2020, from https://theses.ubn.ru.nl/handle/123456789/5631

Nolan, F. (2019, January 24). A Neural Algorithm of Artistic Style: A Modern Form of Creation. Retrieved December 05, 2020, from https://towardsdatascience.com/a-neural-algorithm-of-artistic-style-a-modern-form-of-creation-d39a6ac7e715

Thayer, R. (2000). Mood Regulation and General Arousal Systems. *Psychological Inquiry, 11*(3), 202-204. Retrieved March 17, 2021, from http://www.jstor.org/stable/1449805

Zhu, J., Park, T., Isola, P., & Efros, A. (2020, August 24). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Retrieved November 17, 2020, from https://arxiv.org/abs/1703.10593