Variational Autoencoders For Album Covers

Stuart Colianni[1], Amanpreet Kaur[1], Jay Pranger[1], Marcus Schwarting[1,2]
[1]Georgia Institute of Technology, [2]Argonne National Laboratory

Motivations

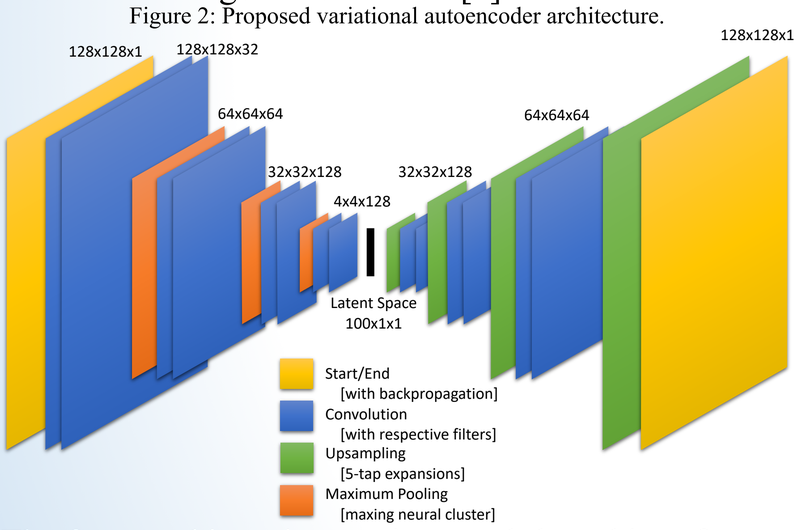
Presently, music recommender systems such as Spotify, Pandora, etc. function as a black box, and allow users minimal control over the discovery process. Additionally, these systems are album cover agnostic, and therefore are deprived of a potentially valuable data source. Our project improves upon existing recommender systems by visualizing the discovery process as a graph network over which the user has full autonomy to explore. Furthermore, we definitively demonstrate the predictive value of album cover art while simultaneously allowing users to generate new art of their own using deep learning.

Figure 1: The top 5 selling albums with their covers (left to right).

Approaches

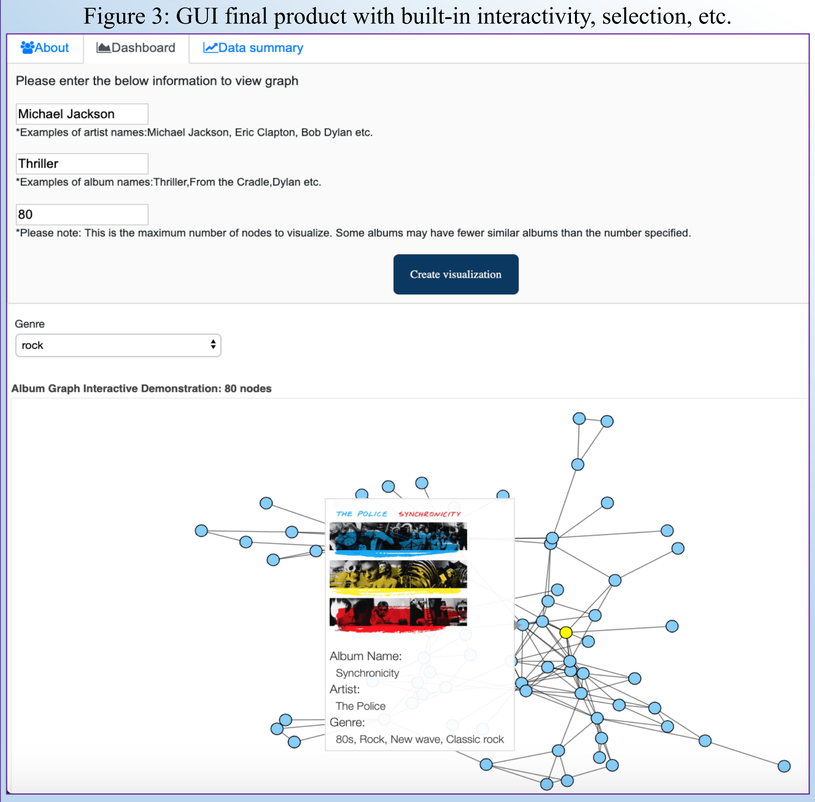
This project consists of a front- and back-end. The back-end utilizes a variational autoencoder (VAE) for generation of custom album covers. The VAE combines encoding and decoding layers to simulate members of the learned distribution. This makes VAEs ideal for the task at hand. While GANs have been used previously for this problem [1], we sought an original generative approach. Our architecture is based on ResNet for image classification [2].

Figure 2: Proposed variational autoencoder architecture.



The front-end is an interactive graph-based interface for easily exploring relations between albums. From a specified artist/album pair, the user can generate a graph of similarly related albums, thereby giving the user autonomy to discover albums they may enjoy. This novel approach operates as a fully user-controlled recommender system leveraging album cover art. The VAE integrates with the front-end based on user-selected inputs.

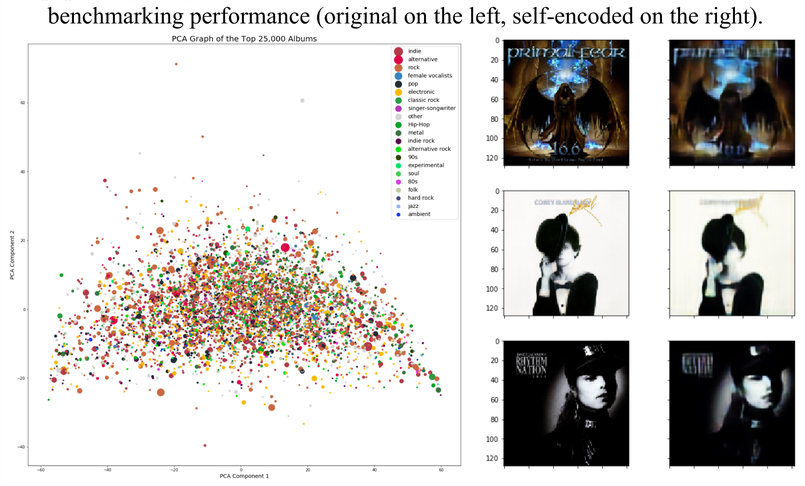
Figure 3: GUI final product with built-in interactivity, selection, etc.



Data Acquisition and Analysis

This project uses a dataset of 24,000 album covers and associated metadata collected via the Last.FM API. Unfortunately, the required API endpoints for bulk data collection went offline for an indeterminate number of days due to a server error. Each element of the dataset was therefore scraped on an image-by-image basis. Records were included in the dataset if they met a popularity threshold of 100,000 plays or 20,000 unique listeners. Once completed, the collection of cover art images required 15.8 GB of disk space. The images were therefore reduced in size from 600 x 600 to 128 x 128. The preprocessed dataset required only 720 MB of disk space – a memory reduction of over 95%.

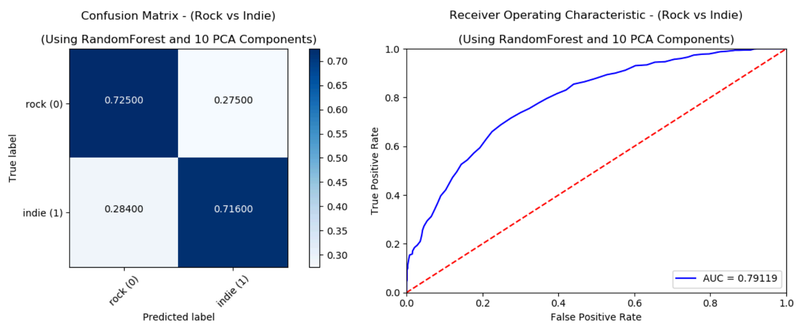
Figure 4: (Left) PCA of top all album covers, colored by genre and sized by popularity. (Right) Self-encoded results from the variational autoencoder for benchmarking performance (original on the left, self-encoded on the right).



Data Validation

This project seeks to answer whether album cover art has predictive value for recommender systems. We formulated a classification experiment that utilizes 10-fold stratified cross-validation. For each fold, the following occurs: All images in the training set are preprocessed to feature vectors via principal component analysis (PCA). A random forest classifier is trained on album cover data from two different genres. The trained classifier is evaluated to create a receiver operating characteristic curve (ROC) and confusion matrix.

Figure 5: (Left) Confusion matrix of two genres, (Right) corresponding ROC curve



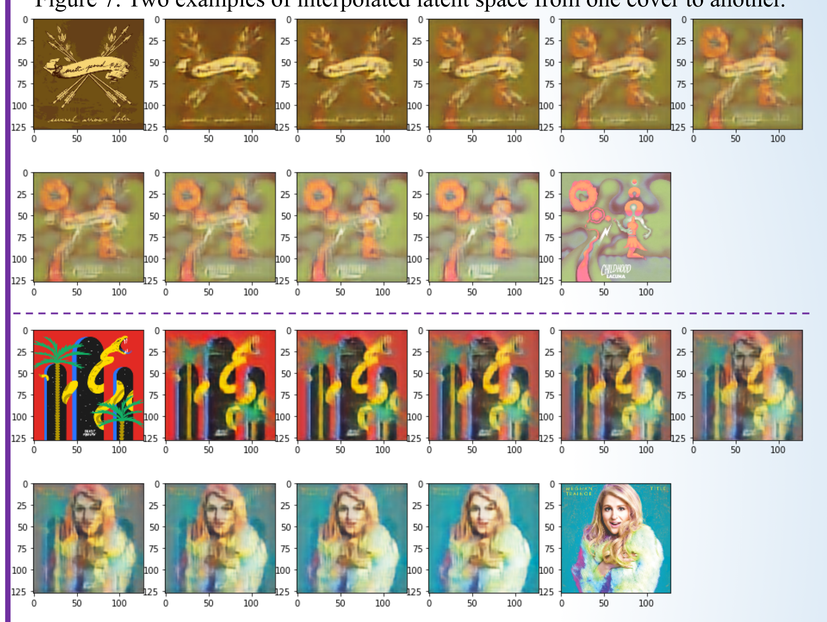
An ROC curve portrays the ability of a classifier to distinguish between two classes as decision threshold variable changes. The area under an ROC curve (ROC AUC) is therefore an indicator of the total ability for a classifier to discriminate between classes. ROC AUC values greater than 0.5 indicate a classifier makes predictions better than random chance, with larger values indicating superior ability. For every genre pair combination from rock, indie, electronic, and pop, the aforementioned experiment produced ROC AUC values between 0.71 and 0.79. In other words, album cover art contains a detectable signal that is significantly predictive of genre. This experiment therefore confirms the predictive value of cover art in augmenting recommender systems. Because the question of using cover art as an alternative data source is a previously unexplored topic, our experiments constitute the state of the art in this area.

Autoencoder Results

The quality of the VAE is evaluated by comparing the original image input with that returned by the VAE decoder. The test set accuracy was 98.7% when trained on 90% of the album data. Training took place over 400 epochs with a batch size of 128 images, and took roughly 4 hours (using TensorFlow on a GTX 1080). In Figure 6, the latent spaces of five album covers are averaged to produce a composite image (shown bottom right). This constitutes a new album album cover generated by the VAE. Additionally, album covers can be iteratively interpolated as shown in Figure 7.

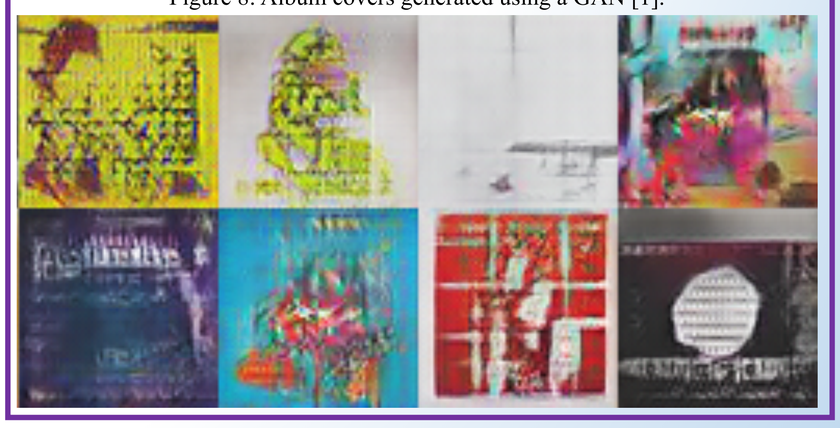
Figure 6: Two examples of a composite album cover made of five existing covers (composite in bottom right)

Figure 7: Two examples of interpolated latent space from one cover to another.



Compared to previous work using a generative adversarial network (GAN), the VAE shows similarly promising results for generating original album covers using both interpolation and averaging methods. The results presented in [1] are shown in Figure 8.

Figure 8: Album covers generated using a GAN [1].



User Feedback & Future Work

Several users gave respectable ratings (3.0 and 3.6 out of 5.0) regarding the VAE image quality and ease of music exploration. We hope to improve in the future by exploring different VAE architectures and also a GAN implementation.

References

[1] Hepburn, A., McConville, R., & Santos-Rodriguez, R. *Album Cover Generation from Genre Tags*.
[2] He, K., Zhang, X., Ren, S., Sun, J. *Deep Residual Learning for Image Recognition*.

Acknowledgements

The authors acknowledge Dr. “Polo” Chau for his instruction in CSE6242.