# Intermediate Project Report

# Ai-sthetic: Generative model for aesthetic visualizations

Gordan Milovac, Marcus Winter, Jeffrey Mu, Arib Syed

Github: [https://github.com/gmilovac/AI-sthetic](https://github.com/gmilovac/AI-sthetic)

## Introduction

We were originally thinking about creating a tool that could rate images based on aesthetics, but we realized those tools already exist and the whole idea is really subjective anyway. So, we shifted to building a tool that can learn from already rated images and, hopefully, create its own visually aesthetic images too. This ended up being both a learning (classification/regression) and generative problem, which is why we decided to go with a Generative Adversarial Network (GAN); we need both a Discriminator (to rate and learn) and a Generator (to create).

## Related Work

Our project is built on the concept of Multi-Task Generative Adversarial Networks (MT-GANs). Instead of just a standard GAN, our Discriminator (the "Judge") has multiple output heads. The core architectural idea is drawn from similar aesthetic GAN models like ADGAN. We use a two-headed Discriminator: one head handles the typical adversarial task (Real vs. Fake), and the second head handles the auxiliary task of aesthetic classification (Aesthetic vs. Not Aesthetic).

Paper: NIMA: Neural Image Assessment (Talebi & Milanfar, 2018)

NIMA is highly relevant because it uses the same AVA dataset to learn human aesthetic perception. NIMA's main goal is to act as a quality meter by predicting the full distribution of human scores (1 to 10). Our Aesthetic Head is simpler: We only perform a binary classification (good or bad, based on the 5.5 threshold). But the key similarity is the shared goal: using the AVA scores to teach a model what humans find beautiful, which in turn guides our Generator's output.

Public Implementations (Living List):

- ADGAN (Aesthetic-Driven Generative Adversarial Network): Search for the paper title on GitHub.
- NIMA (Neural Image Assessment) Implementations: [https://github.com/master/nima](https://github.com/master/nima)

## Data

We are using the AVA (Aesthetic Visual Analysis) dataset. This is the standard, large-scale benchmark for this task, containing over 250,000 images with distributions of human-given scores.
Source: https://www.kaggle.com/datasets/nicolacarrassi/ava-aesthetic-visual-assessment

We built a custom TensorFlow data pipeline to handle file parsing, filtering missing images, resizing all images to 256x256, and normalizing the pixel values to the [-1, 1] range. This requires significant preprocessing, including using a file-based cache to manage the large dataset efficiently.

## Methodology

Our architecture is based on a Multi-Head GAN structure.

The Generator takes a noise vector and uses upsampling layers to create a 256x256 image (using hyperbolic tangent output). The Discriminator uses convolutional layers but splits into two heads after the shared convolutional body: the Realism Head (Real vs. Fake loss) and the Aesthetic Head (Aesthetic vs. Non-Aesthetic loss). The Discriminator's total loss is the sum of these two binary cross-entropy losses, and the Generator is trained to fool both heads.

The most challenging aspect will be achieving training stability and controlling mode collapse. Since we added a second loss function (aesthetics), getting the right weighting between the Realism loss and the Aesthetic loss will be critical to generate both realistic and beautiful images.

If the Generator consistently fails to produce anything but noise, our immediate backup plan is to halt generation and focus on implementing a really robust aesthetic classifier (essentially, a top-tier Discriminator). This is still a valuable tool that can analyze and predict the rating of real images based on certain details.

## Metrics

Ideally, a successful model should generate really aesthetic images. Seeing the GAN homework and the outputs on a simple MNIST dataset, we are not as confident as we were when we began planning the project (the outputs were horrible).

Quantification:

- Discriminator: We can evaluate how well our Discriminator does by comparing the output of the Aesthetic Head to already rated images from the dataset, targeting 65% accuracy.
- Generator (Qualitative): The generator output will be either good or terrible (we will be able to tell by just looking at it) and hopefully if good enough, we can also determine how it’s doing based on what the Discriminator thinks of it.

Goals:

- Base: Successfully train the Aesthetic Head to achieve 65% accuracy on binary aesthetic classification.
- Target: Successfully train the full MT-GAN to generate visually coherent images (not just noise) that score highly on the Discriminator's Aesthetic Head.
- Stretch: Generate high-quality, recognizable images that are visually appealing to humans. We will also experiment with downsizing the images if 256x256 proves too difficult.

## Ethics

There can be very obvious issues with image classification in general. The model can drive wrong conclusions, such as the importance of ethnicity, race, or gender on the aesthetics of an image instead of different driving factors like composition or color. This would cause it to give bad scores in the future and never generate that subset of images, which is extremely problematic because the model would be baking in and reinforcing the societal biases present in the training data.

The AVA dataset was collected by having thousands of individual users rate images. While this gives us a massive amount of data, the data is inherently subjective and potentially biased. The "average score" is reflective of the collective bias of the raters, who primarily come from specific cultural backgrounds. This means our model is not learning universal beauty; it is learning culturally specific preferences. If the model is deployed globally, its definition of "aesthetic" might exclude or undervalue art or photos from cultures not represented in the original labeling process.

## Division of Labor

This is not a finalized list yet, but so far we were meeting and working together on the papers and proposals, and Gordan did most of the initial data work. We will probably all work on this together, but a rough outline is:

- Gordan Milovac: Data pipeline design/implementation, documentation, and final presentation/poster
- Marcus Winter: Generator architecture design, implementation, and visualization tools
- Jeffrey Mu: Discriminator architecture design, implementation, and loss function tuning
- Arib Syed: Testing framework development, model tuning, and hyperparameter search