AI-Powered Color Palette Generator

Ria Srivastava

Abstract:

This project explores the use of artificial intelligence to generate aesthetically coherent color palettes based on natural language descriptions. Using a dataset of image prompts and corresponding color palettes, a neural network model was trained to learn the relationship between text based inputs and their associated color schemes. The process involved extracting colors from a JSON-based dataset, converting them into numerical representations, and using sentence embeddings to represent text prompts. A feedforward neural network was then trained to map these embeddings to RGB color values. This resulting model is capable of generating new, creative color palettes from text inputs, such as “a rainy day in Tokyo” or “a sunset over the ocean,” allowing applications in areas of graphic design, user interface creation, and generating art. This project demonstrates how AI can help aid in language and visual aesthetics, allowing for intuitive and expressive color generation driven by only language.

Background:

Color plays an important role in visual communication. It not only influences emotions but also conveying messages and shaping aesthetic experiences. Designers, artists, and developers often rely on curated color palettes to create visual aesthetic in their work. Traditionally, creating these color palettes requires a blend of experience, creativity, and manual selection. However, with the rise of machine learning and generative models, it is not possible to automate parts of this creative processes.

Recent advances in natural language processing, also known as NLP, and computer visions have allowed AI systems to connect text and visual elements. Particularly, text-to-image models use embeddings of textual prompts to generate the corresponding visuals. Inspired by this idea, I created a project that explores how text descriptions, like “a summer night in Tokyo” or “warm, cozy cabin” can be used to generate color palettes using AI.

By leveraging a dataset of image prompts and corresponding color palettes, I aim to train a model that learns the underlying relationship between languages and color. This approach can offer a intuitive way to generate colors based on emotion, context, or style expressed in natural language processing, opening doors to AI-assisted design tools and creative applications.

Motivation:

Designing effective and visually appealing color palettes is critical in fields like user interface development, graphic design, and digital art. However, selecting the right colors often requires an understanding of color theory but also a clear vision of the emotional or thematic tone of a project. For beginners or even experienced designers, translating abstract ideas like a mood or scene, into a well-balanced color palette can be time-consuming and subjective.

The motivation behind this project is to explore how artificial intelligence can assist in the creative process through allowing users to create color palettes directly from natural language descriptions. By leveraging machine learning models trained on real-world associations between text prompt and color themes, I am to develop a tool that makes color generation more faster and intuitive driven by emotion, style, and context.

The approach empowers users to describe exactly what they envision to create in their art and instantly receive color suggestions that can align with that description. A tool like this could enhance creative workflows, support accessibility in design, and expand ways in which AI can support human design and creativity.

Problem Statement:

Despite the abundance of tools offered for creating color palettes, this process remains inherently subjective and dependent on designers' experience and personal preference. While softwares can provide color suggestions, there is a lack of intuitive, context-aware systems that can allow users to generate color palettes based on abstract descriptions and emotional cues. This creates a challenge for individuals without in-depth knowledge of color theory, as well as creative professionals who are looking for inspiration or a way to automate design workflows.

Moreover, existing methods for color palette generation are typically confined to presets or static rules that limit adaptability and their creativity. There is a need for dynamic, flexible solutions that can understand and generate color themes based on text input, allowing users to easily translate ideas into visually appealing and cohesive palettes.

This project's main focus is to develop an AI-powered model that can generate color palettes from textual prompts. By doing this, it will provide a more intuitive and creative way for users to create custom color schemes driven by descriptive language inputs, whether for design, digital art, or personal use.

Methodology:

To achieve the objective of generating color palettes from textual prompts, I created an approach by leveraging machine learning and deep learning techniques. The methodology consists of these steps:

1. Dataset Collection and Preprocessing
    a. The dataset this project utilizes a publicly available dataset on huggingface.co which contains color palettes associated with descriptive image prompts. The dataset contains a JSON record where each entry includes a text prompt (e.g. “a painting of a Ferrari by Claude Monet”) and corresponding color palettes derived from the images linked to that prompt. Each color consists of a list of RGB color codes.

    The dataset was loaded and preprocessed using Python’s Pandas library. The following preprocessing steps were applied:

- Extracting color palettes from nested JSON data and organizing it into a tabular format
- Cleaning the dataset by removing incomplete or irrelevant entries
- Tokenizing text descriptions into individual words and phrases to prepare for embeddings.

2. Text Embedding
    a. To connect the gaps between text and color data, I used text embeddings. Text embeddings are numerical representations of text which captures semantic relationships between words. Text descriptions were converted into embeddings using a previously trained model, like Term Frequency-inverse Document Frequency and Word2Vec. These embeddings provide a vector representation that my model can use to correlate text descriptions with the corresponding color palettes.
3. Model Design
    a. A feedforward neural network (FNN) was used. The network was designed to take the text embeddings as inputs and generated an outputted RGB color value that corresponds to the color palette for the prompt given. The model structure includes:
        i. An input layer that accepts the text embedding vector
        ii. Hidden layers with activation functions like ReLU to introduce non-linearity
        iii. An output layer that generates the color palette, with a size of 5 representing 5 colors.
4. Training the Model:
    a. The model was trained on the dataset using supervised learning. The input features were the text embeddings, and target output was the list of the RGB color values corresponding to the color palette. The following steps were taken during training:
        i. The dataset was split into training and validation sets, with 80% used for training and 20% for validation
        ii. The loss function used was mean squared error, as the output is continuous (color values)

iii. The optimizer I used was Adam due to its adaptive learning rate capabilities, which is effective in handling a complex dataset like color-palette generation.

5. Evaluation and Tuning
    a. The model's performance was evaluated using metrics like Mean Absolute Error and R^2 score to assess accuracy of the predicted color palettes. Hyperparameters like number of layers, neurons per layer, and learning rate was tuned to improve the performance of the model.
6. Generating Color Palettes
    a. Once trained, the model was tested on unseen text prompts to generate new color palettes. Given a new text prompt, the model converts it into a embedding and then predicts a set of 5 RGB values corresponding to the most likely color palette for that description. These colors are displayed using a visualization tool like matplotlib for better interpretability.
7. Post Processing and Visualization
    a. The generated color palettes were visualized using Python's matplotlib library, which allows for the representation of colors in a visual way. The generated palettes are displayed as horizontal bars with each color in the palette represented as a block of color, providing a clear view of the output.

Datasets and Evaluation

Dataset Description:

The dataset used in this project is sourced from the Hugging Face repository titled "Color Palettes from Stable Diffusion Prompts", a collection of over 2700 entries. Each entry contains

- A text prompt describing a scene, style, or concept.
- A list of images which includes
    - An image URL
    - A color palette extracted from the image, which consists of five hexadecimal color codes

These color palettes are curated representations of visual color themed that are inferred from AI generated images, The dataset then links natural language descriptions to corresponding visual aesthetics in the form of 5 colors.

Data Preprocessing:

To prepare the dataset for training:

- Color values (hex codes) were converted to normalized RGB triplets (e.g., #FF5733 is converted to [1.0, 0.34, 0.2])

- Each text prompt was tokenized and embedded using TF-IDF or pre trained sentence embeddings (using sentence-transformers or spaCy)
- Only the first color palette per entry was used in order to simplify training
- Missing or incorrect data entries were removed

Evaluation Metrics:

To assess the model's ability to generate accurate color palettes, the following metrics were used:

- Mean Squared Error
    - MSE measures the average squared difference between predicted RGB values and actual ones
- Mean Absolute Error
    - MAE provides a more interpretable metric by measuring average absolute difference between predicted and actual color values

Model Evaluation Procedure:

- The dataset was split into training and validation sets.
- The model was trained on the training data and evaluated on the validation set using the metrics above
- The training loss and validation loss were plotted across epochs to ensure the model was learning without overfitting.

Qualitative Evaluation:

In addition to numerical metrics, I performed qualitative evaluations by:

- Visualizing predicted color palettes from new or unseen prompts
- Comparing predicted palettes to actual palettes to see theme similarity
- Gathering subjective feedback on aesthetic quality and relevance of the generated palettes to input prompts.

Model Architecture and Training Results:

The core of this project relies on a feedforward neural network that is designed to learn and generate color palettes. This model consists of three fully connected layers:

1. Input layer: accepts a 15-dimensional vector representing 5 RGB colors.
2. Hidden Layers
    a. The first dense layer has 64 neurons with ReLU activation
    b. The second dense layer expands to 128 neurons to allow for richer feature extraction
3. Output layer: a 15-neuron dense layer that reconstructs the color palette, matching the format of the input.

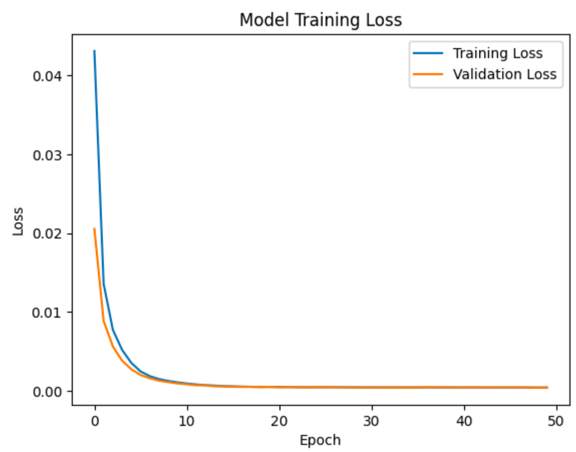
The total number of parameters in this model is 33,839 with 11279 being trainable. This is a relatively lightweight structure and is sufficient for modeling structured, low-dimensional color palette data.

Training Process:

The model was trained using mean squared error as the loss function with the Adam optimizer. The training was conducted over

50 epochs with a batch size of 32. A validation split of 20% was used to monitor the model's ability to generalize beyond the training data.

The figure below shows the loss values over the training epochs:




```
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 64)	1,024
dense_10 (Dense)	(None, 128)	8,320
dense_11 (Dense)	(None, 15)	1,935

```
Total params: 33,839 (132.19 KB)
Trainable params: 11,279 (44.06 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 22,560 (88.13 KB)
```

Summary and Discussions:

After training the neural network to reconstruct and understand the structure of color palettes, the next step was using the model creatively, by generating entirely new palettes from random input vectors.

To do this, I sampled five random 15 dimensional vectors from a uniform distribution. These vectors serve as latent representation, mimicking the format of the actual color palettes. Although these inputs do not correspond to any real palettes, the trained model interprets them based on its learned understanding and produces color combinations.

The generated outputs were reshaped into 5 RGB triplets per palette and visualized using a horizontal bar format. This format makes it easy to observe the color harmony and balance that the model learned.

These results demonstrate the model's ability to create coherent and aesthetic palettes from random noise which is a key trait in generative models.

To qualitatively assess the effectiveness of the model in generating real color palettes, I performed a side-by-side comparison between real palettes from the dataset and those generated by the model. Five real datasets were selected and visualized as well as five generated palettes. This visual comparison allows for an understanding of how well the model captured the structure and aesthetic of actual color combinations.

From the comparison, the generated palette shows a noticeable coherence in color blending and variation which resembles a balance and color diversity seen in the real palette. The outputs display an understanding of color theory which suggests the model successfully learned patterns during training.

Evaluation of Text-To-Color Mapping Using Sentence Embeddings:

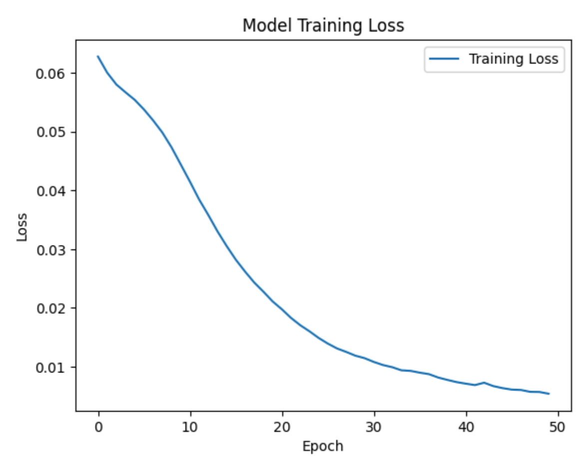
To extend the generative capabilities of this model beyond random latent inputs, I developed a system that learns to generate color palettes directly from natural language prompts. This allows users to input descriptive phrases and receive a corresponding color palette generated by the model.

I used the SentenceTransformers library to convert each text prompt into a dense, 384 dimensional vector using the all-MiniLM-L6-v2 pre-trained model. These embeddings capture semantic information from textual inputs and allows the model to learn the relationship between text and color. The dataset was processed to extract the prompt and the associated color palette in hexadecimal format from each entry. Each hex color was converted into RGB normalized values and I filtered the dataset to include only entries with exactly five colors to maintain consistent input-output dimensions.

The neural network model used is a feedforward fully connected network that accepts a 384 dimensional sentence embedding as input and outputs a flattened vector of 15 normalized RGB values. The architecture consists of:

- An input layer with 128 units and ReLU activation
- A hidden layer with 256 units and ReLU activation
- An output layer with 15 units and sigmoid activation to ensure values remain in the [0,1] range.

The model was trained using the Adam optimizer and mean squared error loss function for 50 epochs. This configuration allowed the model to learn the associations between text and color palette patterns.



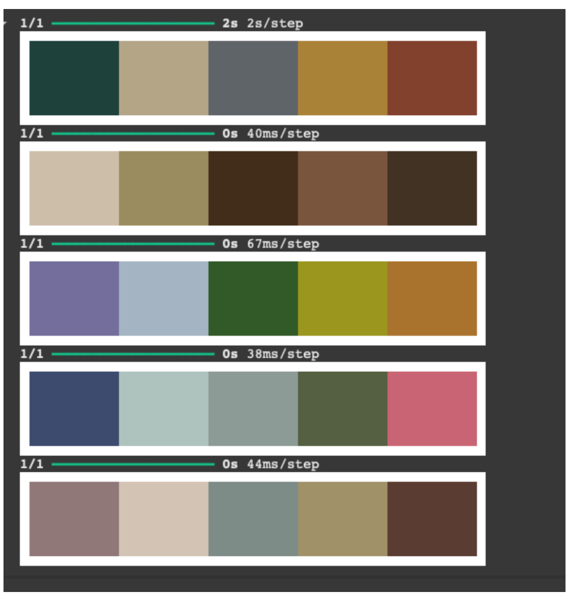
Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 128)	49,280
dense_16 (Dense)	(None, 256)	33,024
dense_17 (Dense)	(None, 15)	3,855

Total params: 258,479 (1009.69 KB)
Trainable params: 86,159 (336.56 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 172,320 (673.13 KB)

I then began testing this by adding my own textual prompts. Those textual prompts being:

```
palette = generate_palette("a rainy night in Tokyo")
show_palette(palette)

palette = generate_palette("a sunny day in New York City")
show_palette(palette)

palette = generate_palette("In a green rainforest")
show_palette(palette)

palette = generate_palette("Peaceful and calm")
show_palette(palette)

palette = generate_palette("A carnival at dusk")
show_palette(palette)
```

These prompts resulted in these color palettes:



Conclusions:

This project demonstrates the creative potential of neural networks in generating color palettes from natural language prompts. By combining sentence embedding models like all-MiniLM-L6-v2 with a simple feedforward neural network, I created and mapped text descriptions to coherent sets of RGB values. The model learns to interpret abstract and descriptive language like “a rainy night in Tokyo” or “peaceful and calm” and translate it into color palettes that reflect the moods and meaning of that text.

The results, while qualitative in nature, show that simple architectures can show the relationship between text and color. Visualizations of both generated and real palettes suggest the model’s outputs are aesthetically aligned with the input text prompt.

This project highlights the intersection of machine learning and design, opening pathways for further development by using more advanced generative models or creating interactive tools for artists or designers. Overall, this project shows how AI can help in creativity by allowing machines to interpret and visually express human language.