기초딥러닝 팀과제 제안서

" Prevalent Hues and Color Picker "

주제의 목적

- Goal : Ease the development of UI/UX design by creating a color picker that picks colors from a set of images based on one's preferred ambience
- How? Allow users to generate a theme based on the colors of the images they will provide as input.

*Build and train a deep neural network (DNN) that can identify colors in an image.

주제의 해결방안

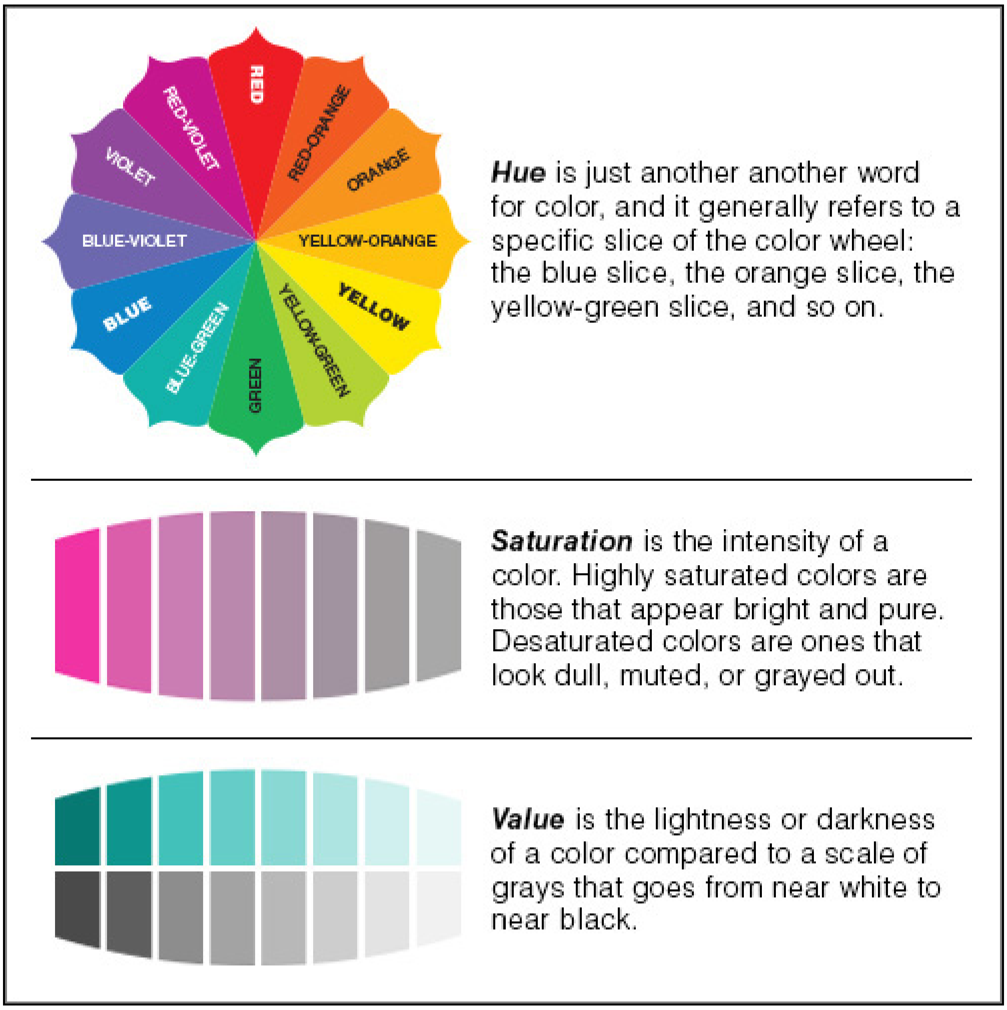
PREREQUISITIES

- Understanding basics of deep learning and image processing.
- Ability to use Python Libraries such as TensorFlow or Keras to build and train a neural network.
- Gather and preprocess a dataset of images for training the neural network.
    - Dataset consists of images with different types of colors and hues to ensure that the model can generalize well.
- Once dataset is collected, build and train a modl using appropriate algorithms and techniques e.g. Convolutional Neural Network (CNNs) and Transfer Learning.
- After training, use the model to predict the colors and generate the color palette.

COLOR 개념

```
Color concepts to study
```

- Hue
- Saturation
- Value
- Tint
- Tone
- Shade



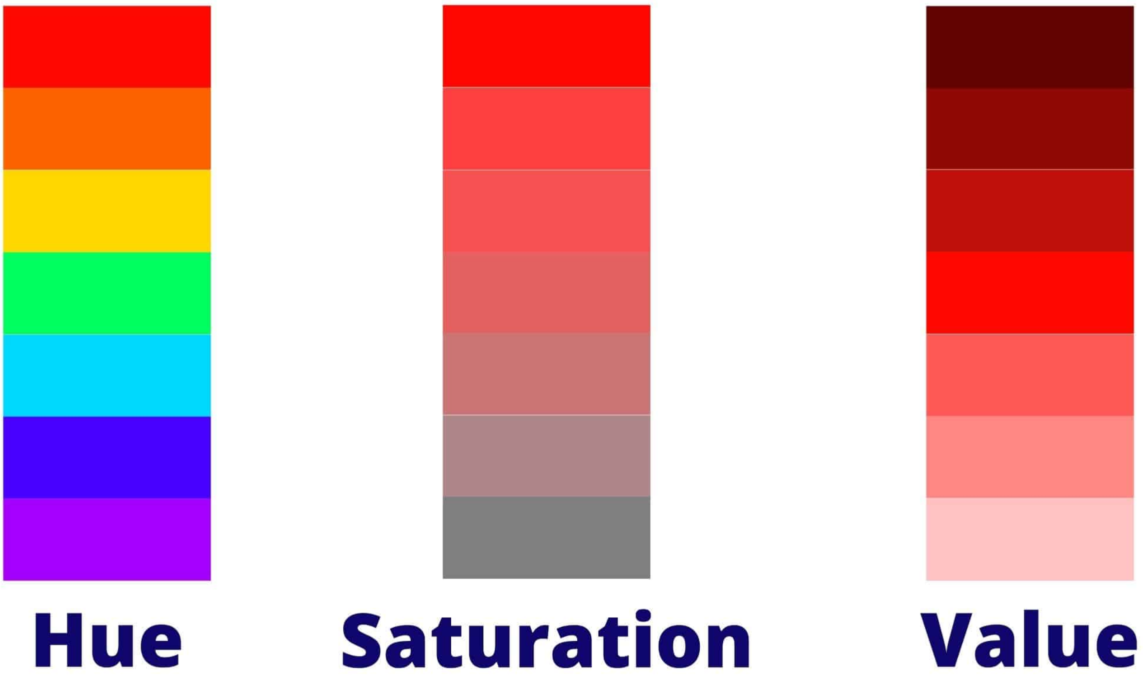
Hue is just another another word for color, and it generally refers to a specific slice of the color wheel: the blue slice, the orange slice, the yellow-green slice, and so on.

Saturation is the intensity of a color. Highly saturated colors are those that appear bright and pure. Desaturated colors are ones that look dull, muted, or grayed out.

Value is the lightness or darkness of a color compared to a scale of grays that goes from near white to near black.

COLOR PALETTE 형식

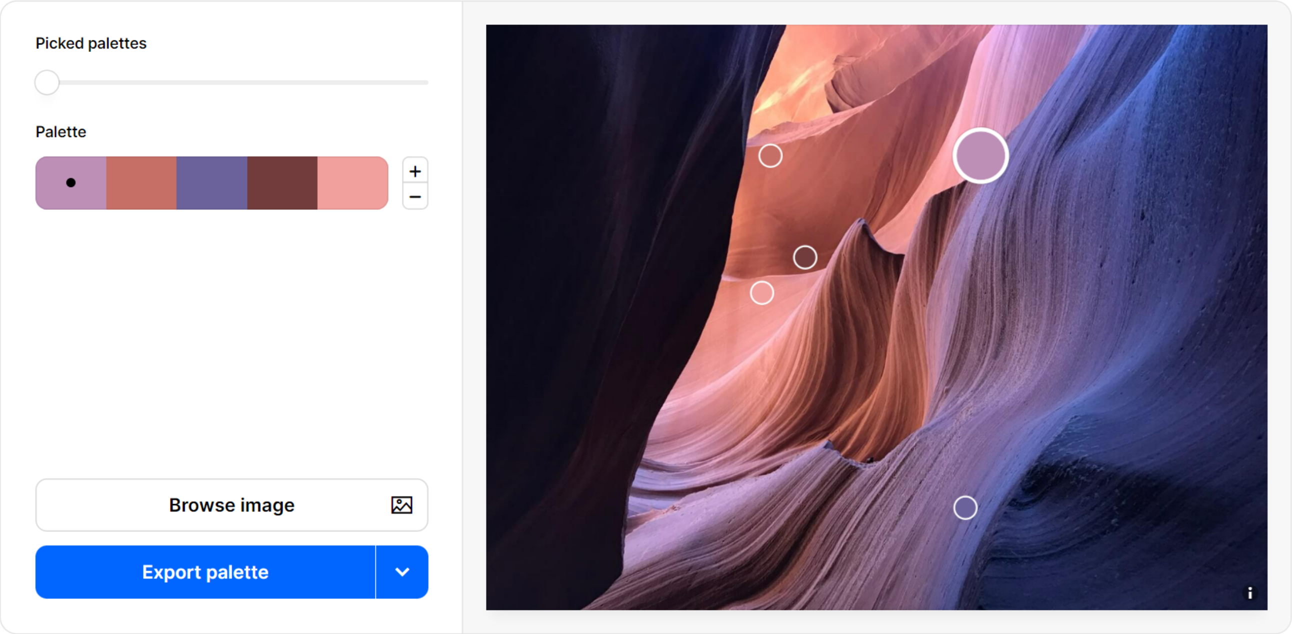
```
How color palette will be displayed
```



NOTE. Color picking values are in RGB (red, green, blue) format. HEX codes will be generated according to the RGB values.

Sample UI

```
Convert to multiple-image input
```



개발 방식

I. Pseudocode

1. Load images
2. Preprocess images
3. Define CNN Model
4. Extract features from images
5. Cluster feature vectors using k-means
6. Get cluster centers as color palette
7. Display palette

II. Implementation

DATA GATHERING

- METHOD I. User Input
  - Restrictions: specific image size, limited amount of images
    (will be decided based on the capabilities of the model)
    - Goal: Allow user to pick their favored images and extract color themes specifically from them; Allow customized theme collection
- METHOD II. CIFAR10 Dataset
  - Goal: Train model with widely used library of 32x32 images; Start small and set further restrictions based on results of training

```
NOTES.
- Image Size and Input Set will be calibrated according to in-between test results.
- User input includes gathering preprovided images of my own selection as input - will
be used for testing/training
```

LEARNING METHOD

1. ***Neural Network: CNN*** - foundation model that learns to gather colors from images, apply color quantization for more focused color extraction
2. ***Color Quantization: K-Means Clustering*** - get most prevalent colors, reduce set of prevalent colors to a smaller and more fixed amount

Concepts

CNN (Convolutional Neural Network)

```
- used to predict color palettes based on input images
  (deep learning to create color palette)
```

Color Quantization

```
- used to reduce color palette of an image to a fixed number of colors k
(color quantization to include only most prevalent colors in color palette)
```

***Basic Concept*

- Color quantization is a clustering problem
  - Clustering points in 3D-space, where points represent colors found in original image (x, y, z axes represent three color channels i.e. RGB)
  - After clusters are located, points in each cluster are averaged to obtain the representative color that all color are mapped to.

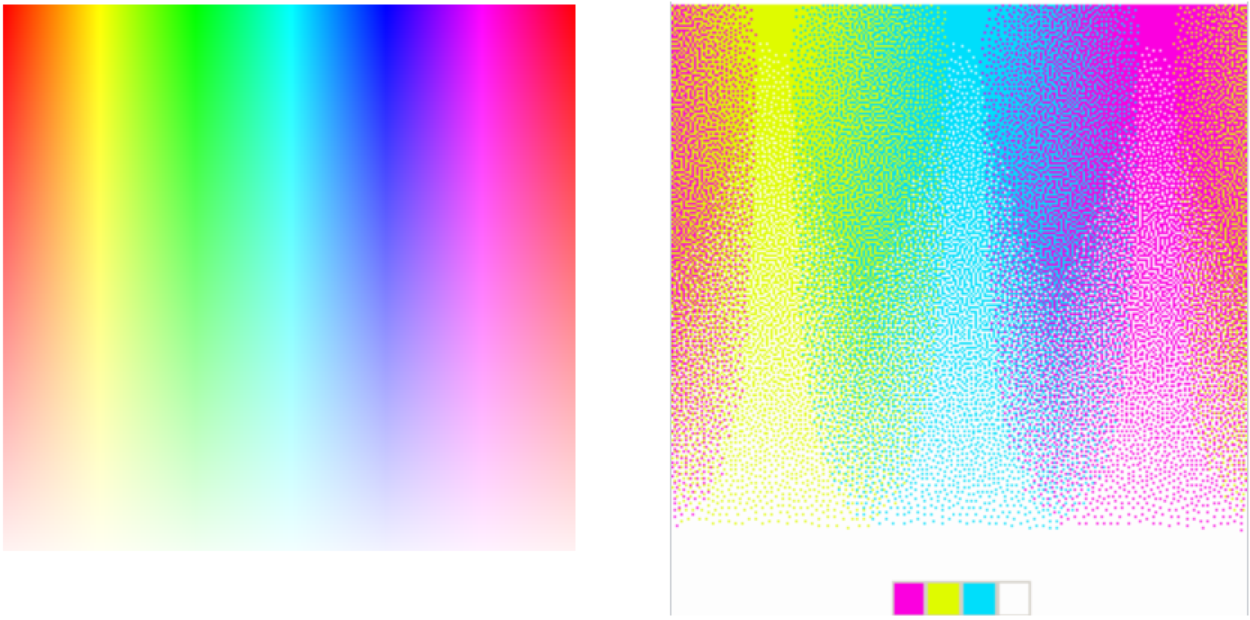


Image I. Spatial Color Quantization -- colorful image reduced to 4 colors

K-Means Clustering

- Method for vector quantization
- Partition n observations (from input data set) into k clusters (i.e. partitions)
  (each observation belongs to the cluster with the nearest mean -- cluster centers or cluster centroid)
- Minimizes within-cluster variances (squared Euclidean distances, not regular Euclidean distances)
- A variant of the "generalized" xpectation-maximization algorithm
  - assignment step == expectation step,
    update step == maximization step

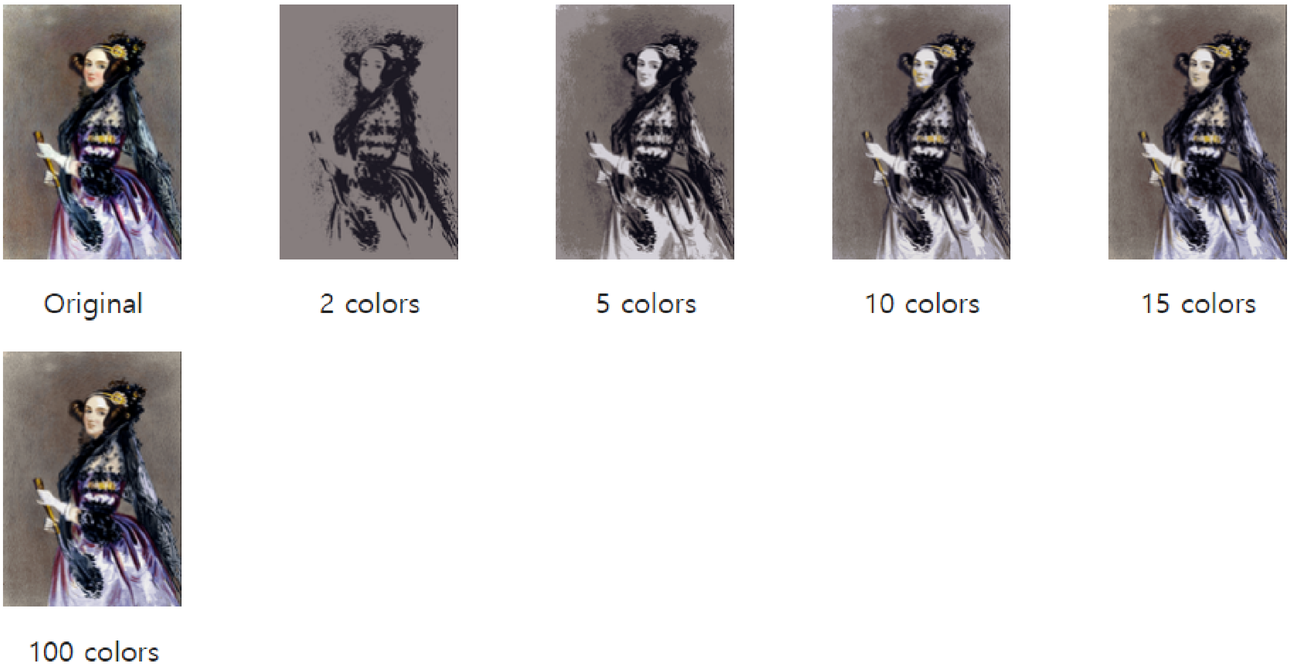
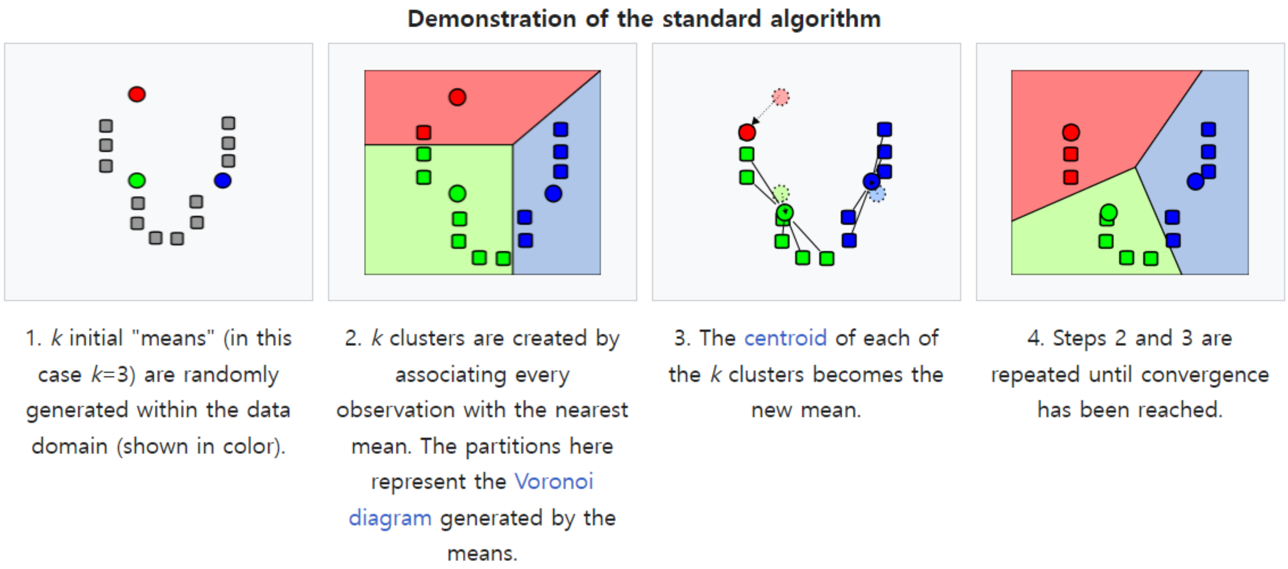


Image II. K-Means Color Quantization - several versions of color processing a portrait of Ada

Lovelace

####### k-Means Clustering Description



Demonstration of the standard algorithm

1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

Image III. Standard Demonstration of k-means Clustering Algorithm

Set of observations $(x_1, x_2, \ldots, x_n)$

- each observation is a d-dimensional real vector

Partition n observations into *k* (≤ *n*) sets **S** = {*S_1, _S_2, ..., _Sk*}

- to minimize the within-cluster sum of squares (WCSS) or variance.

Finaly, find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

where μ_i is the mean (centroid) of points in S_i

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x},$$

and $|S_i|$ is the size of S_i and $||*||$ is the usual L^2 norm.

This is equivalent to minimizing the pairwise squared deviations (variance) of points in the same cluster:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{|S_i|} \sum_{\mathbf{x},\mathbf{y} \in S_i} \left\| \mathbf{x} - \mathbf{y} \right\|^2$$

Deduced from:

$$|S_i| \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \frac{1}{2} \sum_{\mathbf{x}, \mathbf{y} \in S_i} \left\| \mathbf{x} - \mathbf{y} \right\|^2$$

i.e. maximizing the sum of squared deviations between points in 'different' clusters (between-cluster sum of squares, BCSS) because total variance is constant.

PERFORMANCE EVALUATION

- Tests will be done to verfiy the following expected output:
  - amount of colors shown in output is limited
  - gather only the most prevalent colors from the image
  - organize the colors into a simple color palette

주제의 활용 방안

SAMPLE DESCRIPTION

1. Users will gather and select a set of images that represent the theme or ambience that they prefer
2. Users will give them set of images as input to the neural network
3. Neural network then analyzes the set of images
4. Neural network generates a color palette consisting of the prevalent colors picked from the given set of images
5. Neural network displays color palette along with their RGB and hex codes
6. Users refer to the color palette to create designs that synergize with the images in their projects

Reference Material

COLOR PALETTE GENERATORS

- 7 Best AI-Powered Online Color Palette Generators
  https://www.makeuseof.com/ai-powered-online-color-palette-generators/
- Huemint
  www.huemint.com
- Khroma
  www.khroma.co
- Coolors
  https://coolors.co/image-picker
- PaletteGenerator
  https://palettegenerator.com/

Google Keywords
"ai that gets hues and colors from image"
"image and color picker ai"
"ai how to make image hue and color picker"

A.I. CONCEPTS

- Color Quantization
  https://en.wikipedia.org/wiki/Color_quantization
- K-Means Clustering
  https://en.wikipedia.org/wiki/K-means_clustering
- Clustering Algorithm (Cluster Analysis)
  https://en.wikipedia.org/wiki/Data_clustering

Google Keywords
"color quantization"
"k-means clustering"

RELATED LIBRARY FUNCTIONS

- Scikit Learn - KMeans Clustering functions (sklearn.cluster.KMeans)
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- Scikit Learn - Clustering Methods
  https://scikit-learn.org/stable/modules/clustering.html#k-means
- Python OpenCV2 - cv2.imread() Method
  https://www.geeksforgeeks.org/python-opencv-cv2-imread-method/