기초딥러닝 텀과제

"Color Palette Generator"

주제의 목적

Understand and utilize clustering algorithms and neural networks to identify most prevalent colors in an image / set of images and generate a color palette. Create a program that can potentially ease the burden of color-picking in the process of UI/UX Design.

주제의 해결방안

- Color Quantization via K-Means Clustering
  - for gathering center colors (most meaningful colors) from cluster of colors in an image
- Improved Accurracy and Added Features via Deep Neural Networks
  I. Feature Extraction via CNN (Convolutional Neural Network)
  - for gathering meaningful features (patterns, etc.) and transform raw pixel data into more informative representation
    II. Image Segmentation via FCNN (Fully Convolutional Network)
  - for dividing
    III. Harmonic Color Suggestions via LSTM (Long Short Term Memory)
  - for suggesting more colors that harmonize well in addition to the generated color palette

흐름 예제

[Color Quantization]

1. Read single input image
2. Cluster colors
3. Create and output palette

[Color Quantization + Image Segmentation]

1. Read single input image
2. Divide image into regions
3. Cluster colors based on regions
4. Create and output palette

[Color Quantization + Feature Extraction]

1. Read single input image
2. Extract meaningful features (patterns) from image without performing explicit segmentation

3. Cluster colors based on acquired features
4. Create and output palette

[Color Quantization + Image Segmentation + Harmonic Color Suggestion]

1. Read single input image
2. Divide image into regions
3. Cluster colors based on regions
4. Group together most prevalent or "harmonic" colors
    - Add more colors to palette as suggestions (or according to max palette size)
5. Create and output palette

개발 방식

Image Input

```
OpenCV(cv2), PIL Image
```

Palette Generation

```
Seaborn, Plotly Express
```

Color Quantization : K-Means Clustering

```
Extracting colors from images and compiling them via k-means clustering
```

- Concept
    - Reduce the amount of colors distinguished or classified (into a given specific amount of clusters) for simplicity
    - Limitations
        - globally optimal result may not be achieved ()
        - number of clusters must be selected beforehand (cannot learn clusters from data - "finds best clusters")
    - fundamental assumption: points are closer to their own cluster center than others
- Disadvantages
    - Inability to learn clusters based on data
        - Workaround: Give a specific number of clusters as value of k
    - Slow for large numbers of samples (each iteration of k-means must access every point in the dataset)
        - Workaround : Use subset of each data to update cluster centers at each step (via scikit.cluster.MiniBatchKMeans)

Harmonic Color Suggestions : LSTM (Long Short Term Memory)

```
Suggesting additional colors or most prevalent colors based on input images via neural network that utilizes Long Short Term Memory
```

- ***Color Harmony*** : Train a DNN model using LSTM to learn color harmonies and relationships from a large dataset of images or color schemes. Use model to evaluate the color combinations generated by K-means clustering and provide feedback on the overall harmony or suggest adjustments to create more aesthetically pleasing color palettes.

  "Read color relationships/overall theme. Suggest other colors accordingly."
- Possible Issues
  - LSTM may produce middling values (i.e. dull browns, greys --> dull colors)
- Suggestive Improvements
  - Modify Pre-processing + Post-processing techniques or Do Fine-Tuning :
    to avoid dull-toned (due to prevalent selection of middle colors) palettes

Image Segmentation : FCN (Fully Convolutional Network)

```
Divide images into segments (regions or objects) via Fully Convolutional Network and perform K-Means clustering on each segment for more accurate color palettes
```

- ***Image Segmentation*** : Use a FCN-based image segmentation model to separate the image into different regions or objects. By identifying distinct objects or areas, K-means clustering can be performed on each segment individually, allowing for more accurate and meaningful color palettes.

  "Do image segmentation. For more accurate color palettes."
- Concept
  - Calculate intensity values of each HSV (Hue, Saturation, Value) channel
  - Segment the image according to intensity values

Feature Extraction : CNN (Convolutional Neural Network)

```
Extract meaningful patterns, characteristics from images via Convolutional Neural Network and
perform K-Means clustering according to those extracted features for more accurate color
palettes
```

- ***Feature Extraction*** : Obtain relevant and informative representations or features from an input image. These features capture specific patterns, structures, or characteristics of the image that are useful for subsequent tasks such as classification, object detection, or image

retrieval. Feature extraction aims to transform the raw pixel data into a more compact and meaningful representation that can be easily understood by machine learning algorithms.

"Gather meaningful features. For more accurate color palettes."

주제의 활용 방안

1. Users will provide a set of images that represent the theme or ambience that they prefer
2. Set of images are prepared and pre-processed
3. (Upon implementation of Feature Extraction) Features from images will be extracted
4. (Upon implementation of Image Segmentation) Images will be segmented via FCNN
5. Prevalent colors are picked from the set of images via k-means clustering
6. Prevalent colors are visualized into a color palette (displaying their hex codes)
7. (Upon implementation of Harmonic Color Suggestion) Additional colors will be added to the color palette or suggested to the user
8. Users refer to the color palette to create designs that synergize with the images in their projects

성능 평가

General

- Visualize generated results as color palette via Seaborn and/or Plotly Express
- Compare generated R, G, B and HEX values with ground truth values

K-Means Clustering

- Compare results with actual colors included in color palette images
    - Extract colors from palette images via extcolor library

CNN

- Visualize features and compare results with other models
- Others : evaluate transfer learning performance, visualize activations, etc.

FCN

- Compare with ground truth segmentation mask

LSTM

- Compare results with generated color palette

개선사항

- Implement more optimized neural network models
  - Analyze proper training method for each DNN model
- Better image input options
  - Optimize single input and multiple input
  - Use GUI Interface (e.g. TKinter, or port to HTML)
- Try fine-tuning pre-trained models
  - Pix2Pix : https://www.tensorflow.org/tutorials/images/segmentation?hl=ko
  - DeepLab : https://github.com/tensorflow/models/tree/master/research/deeplab

참고자료

Inspirational Works

- Colormind.io - http://colormind.io/
  - color palette generator that uses neural networks (specifically, LSTM then GaN model)
- Coolors - https://coolors.co/image-picker
  - versatile color picker that extracts palettes from images
- PaletteGenerator.com - https://palettegenerator.com/
  - color picker that utilizes k-Means clustering, has good multiple-input feature
- Rickrack - https://github.com/eigenmiao/Rickrack
  - intricate software application for creating color palettes and a variety of other functions