COMP3359 Artificial Intelligence Applications

Interior Home Design Style Hunter

Final Report

Group Member

Group 29

Liu Wei 3035556589

Huang Ziqian 3035329948

Abstract

Interior design has grown as an important way to show fashion taste and lifestyle. However, not all people could tell what is his favorite interior design style, which leads to communication barriers with interior designers. This paper provides an approach to infer the interior design the customer prefers from the color pictures he loves, such as paintings, by feature extractions and machine learning techniques. Combining color features and Convolutional Neural Network (CNN), the prediction reaches 50% accuracy of the test dataset. Lastly, this project displays the prediction to the user with the interior style and a corresponding interior picture.

Table of Contents

Table of figure

1. Introduction

This paper describes an approach for identifying favorite interior design styles for people with feature extractions and machine learning techniques. In the field of interior design, interior style found the communication base between customers and designers. One past paper mentioned classifying interior pictures with design style (Kim & Lee, 2020), while other studies on interior design focus on the 3D arrangement, such as InteriorNet (Li et al, 2018) and architecture style (Zhe Xu et al, 2014). This work started from image dataset labeling using raw images from Airbnb, a worldwide online lodge platform. In order to eliminate the impact of shapes and textures, this paper used color features extracted from interior images, the training set, and paintings, the test set, to configure the SGD classifier in sklearn and Convolutional Neural Network. Figure 1 shows the rough pipeline. In the following sections, this paper explains why this pipeline surpasses pre-trained CNN in this application, then states the process to configure the SGD model and analyzes the performance based on accuracies, AOC, and so on. Last but not least, the user interface is shown with file structure and user guide.

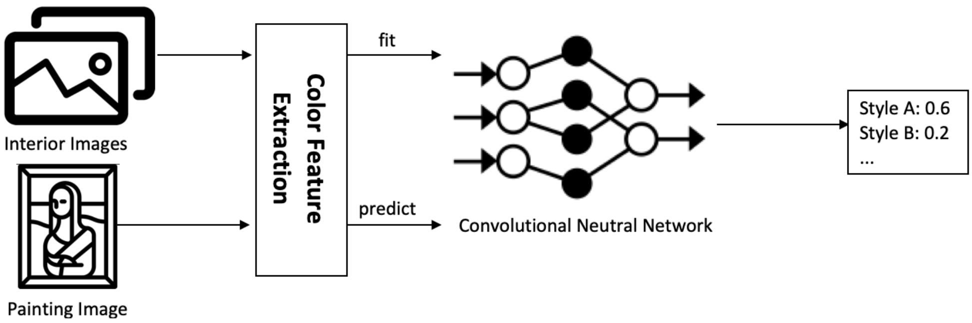

Figure 1 model usage brief

2.Methodology

2.1 Data Source

The training dataset in this project is the interior house pictures collected from Airbnb, an online marketplace for house rent, to classify interior house styles. The cleaned datasets (Inside Airbnb, 2021) contain a group of detailed information, including the available date, price, and picture URL but not any style labels. This project utilized the picture URL to obtain the image for training. Because the artwork style is similar to the house design style, this project took artwork painting as a test dataset for the trained model. Also, painting images is one of the choices that users input. This project chose the Painter By Number dataset in the Kaggle competition (Kaggle) containing 2319 artists' 103 thousand paintings with the genre, author, date, and type labels. However, these labels are different from interior styles so both the training and test datasets required data labeling.

2.2 Data Labelling

The Airbnb datasets of New York and London provide image URLs that required download to form a raw image dataset, while the Painter by Numbers dataset provided images directly. According to Simon,H.A.(1975), style in design contains various aspects, such as the common feature presented in the design work and the manner in designing. Style can also be determined by features, including shape, color, pattern, material and texture, and specific feelings, such as romantic, natural.

Research (Year)	Modern	Natural	Classic	Casual	Romantic	Elegant	Minimal
Kwak & Park (2000)	V	V	V	V	-	V	V
Kim and Lee (2004)	V	V	V	V	V	V	V
Kim (2006)	V	V	V	V	V	-	V
Cho & Kim (2007)	V	V	V	V	V	V	V
Lee & Park (2013)	V	V	V	V	V	-	-
Min & Choi (2014)	V	V	V	V	V	-	-
Jeong & Hwang (2018)	V	V	V	V	V	-	-

Figure 2 Categories of design style in some references research

The figure 2 (Kim. J & Lee.J, 2020) shows the categories of design style utilized in some referenced research. Modern, Natural, Romantic, Minimal, Classic (also called Country) are utilized in the high frequency. In this paper, we will focus on these five design styles and classify the images using below definition:

Style	Description
Romantic Style	Color: multiple colors, high saturation
Mordern Style	Color: bright colors, high contrast, gold
Natural Style	Color: green, yellow, wooden
Country Style	Color: brown, dark red
Minimalist Style	Color: white, black, low saturation

Table 1 Style description for data labeling

Furthermore, we utilize numerical representation for the categorical label (e.g. index 1 represents Romantic, index 2 represents Modern) to reduce the workload of image labeling.

However, during the training process the training label and the testing label process one-hot encoding, even though we utilize integer representations rather than category names in the labeling process. Because machines assume the integer values to have a natural relationship with each other (Lee, 2017), which is opposite to our expectations and influences our final result. Also, the machine processes much faster with 0 and 1 (Lee, 2017).

2.3 Feature Extraction

Each digital image of the house interior design is different in resolution and horizontal and vertical ratio. The larger size of the image is the more time consumed on preprocessing, while resizing images won't affect the features extraction results. In order to extract feature value from these house-style images quickly, the project resizes the pictures into the shape (224,224). Furthermore, the resolution of the images provided in the dataset was another important factor to influence the quality of features. Clear pictures can provide more detailed information, while low-definition pictures are difficult to extract the same detailed information as clear pictures, such as texture. Therefore, this project focused on extracting the features that are independent from the resolution, namely the color feature.

House interior design style can be determined through color, shape, and structure. This paper mainly concentrated on extracting color features from the images. When people look at the image, they tend to define the image from color distribution at first glance, and then the other detailed information. Also, for the target user, each of them has their own favorite colors, which helps to match with interior style.

Color Model

This section describes which color space to choose and how to extract the color feature through the statistical computation of the overall pixel of an image. Generally, images are inserted using an RGB model. Three channels Red, Green, Blue will all be dramatically influenced by the external lighting, such as shallow, pale, which is not similar to the human experience of color perception(Wikipedia). The grayscale measures the intensity of light at each pixel. The Hue Saturation Value (HSV) color space is much more robust with the external lighting changing. It means that Hue values vary less than RGB values when there is a minor change in the lightness (Thevenot, 2020).

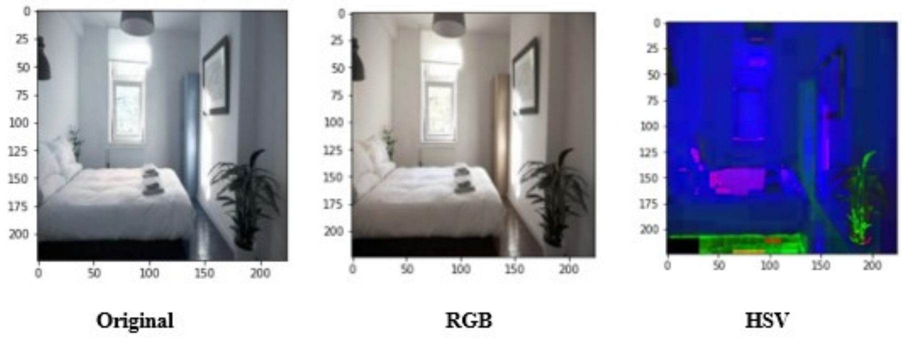

Figure 3 Picture in different color space

Color Histogram & Quantization

The color histogram represents the distribution of colors in an image. Using the color histogram, this project could ignore the shape and edge of the objects inside the image, and only concentrate on the color feature to ensure that the later training model is working as a style classifier, not the object detector (Sharma, 2019).

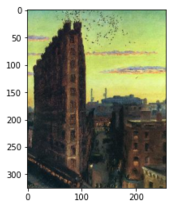

Figure 4 initial picture

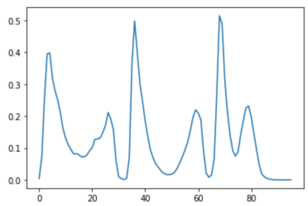

Figure 5 RGB histogram

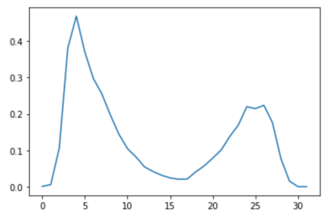

Figure 6 HS histogram

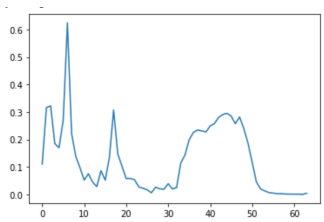

Figure 7 gray histogram

However, the dimension of the histogram vector is too high, because a single image consists of many colors. To deal with this problem, this project made use of color quantization, a process to reduce the number of distinct colors in an image, specifically the HSV color quantization shown in the following correspondence (shown in figure 8) :

$$H = \begin{cases} 0 & H \in [316, 20] \\ 1 & H \in [21, 40] \\ 2 & H \in [41, 75] \\ 3 & H \in [76, 155] \\ 4 & H \in [156, 190] \\ 5 & H \in [191, 270] \\ 6 & H \in [271, 295] \\ 7 & H \in [296, 315] \end{cases} \qquad S = \begin{cases} 0 & S \in [0, 0.2] \\ 1 & S \in [0.2, 0.7] \\ 2 & S \in [0.7, 1] \end{cases} \qquad V = \begin{cases} 0 & V \in [0, 0.2] \\ 1 & V \in [0.2, 0.7] \\ 2 & V \in [0.7, 1] \end{cases}$$

Figure 8 HSV color quantization table

Therefore, each color component would be synthesized into 72-dimensional one-dimensional vector according to this equation: G=9H+3S+V (Ma, 2018).

2.4 Classification

EfficientNet. Efficient-net achieves the highest ImageNet accuracy with the same amount of parameters and the accuracy steadily increases as the complexity increases in image classification (M.X.Tan & Q.V.Le, 2020). In this paper, we applied transfer learning with the EfficientNet B1 model because transfer learning helps to reduce the size of the required training dataset and achieve an acceptable prediction accuracy at the same time. Below is the configuration of the model (in figure 9) and the procedure for the transfer learning:

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
efficientnet-b1 (Functional) (None, 7, 7, 1280)        6575232
_________________________________________________________________
gap (Flatten)                (None, 62720)             0
_________________________________________________________________
dropout_out (Dropout)        (None, 62720)             0
_________________________________________________________________
fc_out (Dense)               (None, 7)                 439047
=================================================================
Total params: 7,014,279
Trainable params: 439,047
Non-trainable params: 6,575,232
_________________________________________________________________
```

Figure 9 configuration of pre trained-model EfficientNet

1. With backbone model (Efficient Net B1), pre-trained on ImageNet dataset.
2. Frozen the parameter of the backbone model.
3. Train the last layer by inserting the home interior dataset

CNN. Apart from using the pre-train model EfficientNet. We also build a simple CNN model to predict the design style. In the model training, the color feature extracted from the color histogram will be as the input data X, and the corresponding one-hot labelling as the learning output/predict output. The model configuration (shown in Figure 10)consists of two hidden layers with relu activation and one softmax layer at the end.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 72)                0
_________________________________________________________________
dense (Dense)                (None, 32)                2336
_________________________________________________________________
dense_1 (Dense)              (None, 16)                528
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 85
=================================================================
Total params: 2,949
Trainable params: 2,949
Non-trainable params: 0
_________________________________________________________________
```

Figure 10 configuration of CNN model

SVM Classifier. The Support Vector Machine is a supervised learning model with associated learning algorithms that analyze data for classification (Wikipedia). It is popular and has been applied extensively for pattern classification,regression, and density estimation. In this paper, we use a one-vs-all method to train linear classifiers with SGD training. Within SGD, the gradient of the loss is estimated each sample and the model will be updated along the way with a decreasing learning rate. Also, SGD supports mini-batch learning through the partial_fit method. In the SGD model, the loss function and penalty utilize hinge and l2, respectively, a typical linear SVM. Furthermore, the tolerance is 0.001, indicating training stops when the loss is larger than the best loss tolerance.

3. Result and Analysis

This chapter describes the training results of different types of machine learning models with various preprocessing techniques. The criteria of trained models include overall accuracy, precision, recall, and F1 score for each class of test dataset predictions. It also presented how the feature extraction and amount of data affect the models. Finally, this section demonstrates how to use the model in the application from the view of a customer hunting favorite interior style.

3.1 Criterie to use

This project urges to find out the corresponding interior style given any colorful picture. So the performance of classification models is measured based on the test dataset, 200 painting images randomly selected from Painter by Numbers dataset from Kaggle. Apart from accuracy, the F1 score, the harmonic average of precision and recall, is selected as a significant performance measurement of the classifier in this project. The definitions of precision, recall, and F1 score are shown as following equations.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1\ score = \frac{2 \times precision \times recall}{recall + precision}$$

For each target style to study, true positive means the classifier correctly classifies the painting image in the target style while false positive indicates the model misclassified the image in the target style, which should be classified as other styles. The false-negative shows the number of images in the target style misclassified to other styles.

3.2 Pre-trained EfficientNet with images

Considering the time limitation and available data size, a pre-trained model may be a good starting point. 1590 interior pictures were used to train the model, while the last 400 interior pictures fit as a validation set. After 10 epochs, the accuracy history is shown in figure 11.

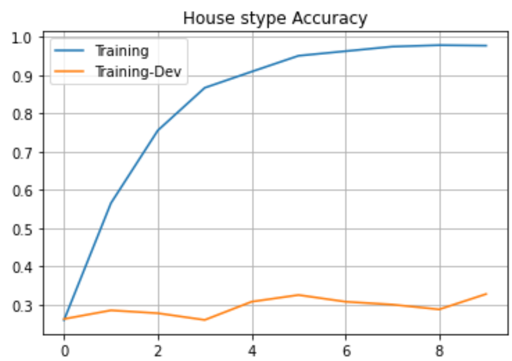

Figure 11 EfficientNet with 1590 images

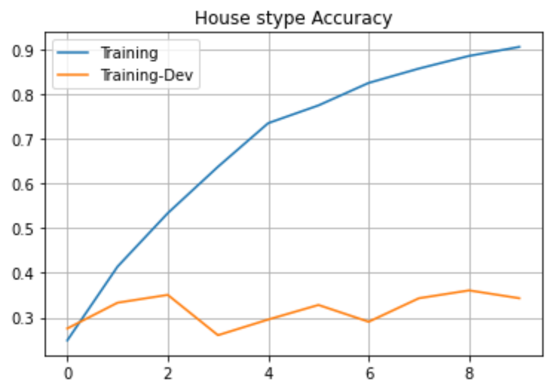

Figure 12 EfficientNet with Drop-out

The large difference between training accuracy and validation accuracy indicates that the model suffers from an overfitting problem. The model remembers each training data instead of the inner pattern. The possible causes usually come from high model complexity and no random elements when fitting data such as data augment or drop out. Mingxing Tan and Quoc V. Le (2020) experienced reinforcement learning with Efficient-Net B0 using CIFAR-10 (Krizhevsky & Hinton, 2009) and achieved 98.1% accuracy. The training data set has seven classes, similar to the number of classes in CIFAR-10 (Krizhevsky & Hinton, 2009) so the complexity of the model is not likely the cause.

After adding a drop-out layer with a 0.5 drop rate, the validation accuracy increased by 3% shown in figure 12. However, the gap between the training accuracy and validation accuracy remained large.

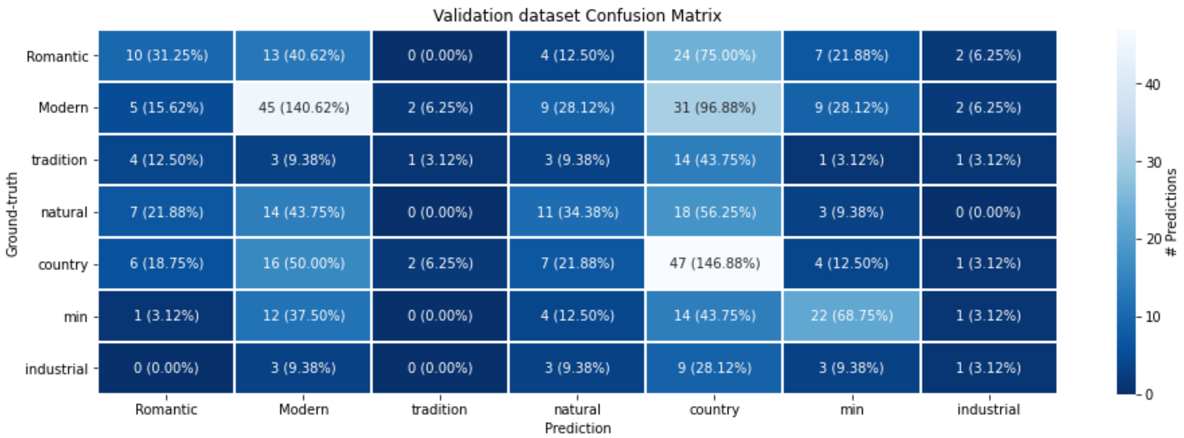

Figure 13 confusion matrix of EfficienNet with validation data

After a further study on the confusion matrix, the noticeable lack of data in the traditional and industrial style inspired us to remove these images from both the training dataset and test dataset. The study made by Kim and Lee in 2020 exposed the possible reason for overfitting, the weight of a pre-trained dataset. Moreover, EfficientNet was pre-trained by the ImageNet dataset, a more general images dataset compared with Place 365 which relates more closely with interior style.

Besides, the model takes all the information in the image such as colors, textures, shapes, and lightness. It may perform worse on the test dataset, the painting images since they have fewer patterns in common. To sum up, feature extraction is necessary to filter the noisy features, textures and shapes, and weigh the color features that are common in all colorful pictures. In the coming model training, this project dropped the two styles, traditional and industrial styles.

3.3 SVM model with feature extractions

This section describes the effect of different feature extractions on the performance of the same SGD. Table 2 shows the result with different features with 800 training images, 200 validation images, and 100 test images with 5 styles.

Features	input shape	training accuracy	validation accuracy	test accuracy
RGB histogram	(batch, 96)	32.75%	21.5%	30%
HS histogram	(batch, 64)	32.875%	26%	33%
gray histogram	(batch, 32)	25%	25%	24%
qualified hsv	(batch, 72)	24%	28.9%	34.5%
qualified hsv with one-hot coding	(batch, 72)	22.125%	30.87%	35.12%

Table 2 Accuracies of SVM with different features

The base accuracy, random guess, is 20% for 5 styles. With the SVM classifier, the test accuracies using color features only increased around 5%, much lower than expected. The character of SVM, linear boundaries, maybe the cause since the ground true boundaries may not be. Apart from the low accuracy, the F1-score of each style differs a lot. With gray histogram, f1-score for minimalist style is 0.42 while the one for natural style is 0. When looking into the details of SVM with qualified HSV features, the imbalanced performance of each class happens as well. One-hot coding was also applied with the qualified HSV feature to cure the imbalance, but it gets more serious with one-hot coding.

```
0:romantic 1:morden 2:natural 3:country 4:minimalist:
              precision    recall  f1-score   support

           0       0.50      0.20      0.29        55
           1       0.14      0.06      0.08        17
           2       0.00      0.00      0.00        30
           3       0.31      0.91      0.47        44
           4       0.58      0.32      0.41        22

    accuracy                           0.35       168
   macro avg       0.31      0.30      0.25       168
weighted avg       0.34      0.35      0.28       168
```

Figure 14 test report of SVM with one-hot coding

```
0:romantic 1:morden 2:natural 3:country 4:minimalist:
              precision    recall  f1-score   support

           0       0.43      0.78      0.55        55
           1       0.09      0.29      0.14        17
           2       0.00      0.00      0.00        30
           3       1.00      0.16      0.27        44
           4       0.60      0.14      0.22        22

    accuracy                           0.35       168
   macro avg       0.42      0.27      0.24       168
weighted avg       0.49      0.35      0.30       168
```

Figure 15 test report of SVM with qualified HSV

```
0:romantic 1:morden 2:natural 3:country 4:minimalist:
              precision    recall  f1-score   support

           0       0.51      0.78      0.61        55
           1       0.08      0.18      0.11        17
           2       0.75      0.20      0.32        30
           3       0.69      0.41      0.51        44
           4       0.64      0.32      0.42        22

    accuracy                           0.46       168
   macro avg       0.53      0.38      0.40       168
weighted avg       0.57      0.46      0.46       168
```

Figure 16 test report of SVM with 1320 training data

This imbalanced result may result from data labeling bias and training data size. After feeding more data,1320, with HS histogram which performed best, the test accuracy remained the same, 33%, while the training accuracy and validation accuracy increased 5%. Meanwhile, the test accuracy significantly increased, from 34.5% to 45.8% when feeding 1320 data with qualified HSV features.

To sum up, among all the color features, the SVM model taking qualified HSV features performs the best on the test dataset. Because the accuracy is limited by the character of SVM, namely the linear boundaries, this project turned to CNN to improve the accuracy using qualified HSV features.

3.4 Simple CNN with qualified hsv histogram

After applying feature extraction with SGD models, this project turned to CNN to see if any improvement compared to SGD. To avoid overfitting due to model complexity, the trial started from a simple CNN as shown in figure 10. Trained with 800 images, the simple CNN achieved 44% training accuracy and 41.7% validation accuracy, shown in figure 17.

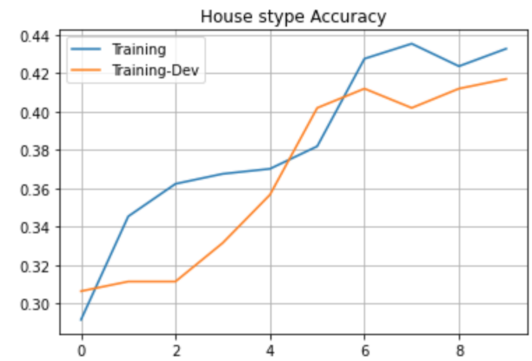

Figure 17 Accuracy history of simple CNN (8 epochs)

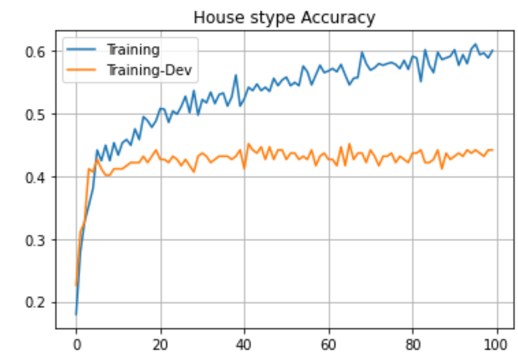

Figure 18 accuracy history of simple CNN (100 epochs)

Though the training accuracy increased when increasing the training epoch, the validation accuracy oscillated around 41% while the overfitting issue grew larger. Figure 19 demonstrates the performance of the simple CNN model with a test dataset, 160 paintings. The percentages along the diagonal line show the recall score for each style. Though the test accuracy was relatively high, 50%, there were few images classified as Modern or natural style. The same problem happens in the confusion matrix of training data and validation data so the image source is not like the cause.

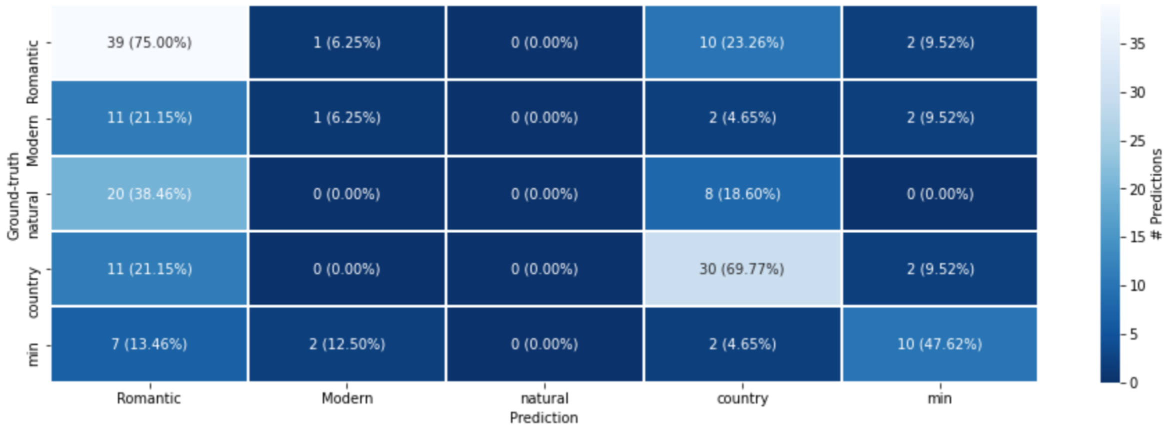

Figure 19 confusion matrix of test dataset in simple CNN

When this simple CNN was retrained by more images with the HSV feature, namely 1412 training data and 353 validation data, the training accuracy and validation accuracy did not change much while the test accuracy decreased 5%. However, the recall score of modern style on validation dataset increased significantly from 0% to 48.35% from figure 21.

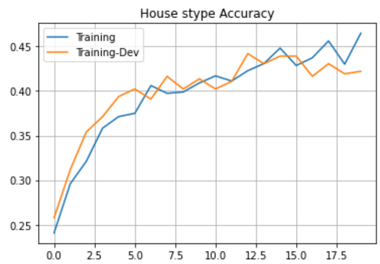

Figure 20 accuracy history of simple CNN with more data

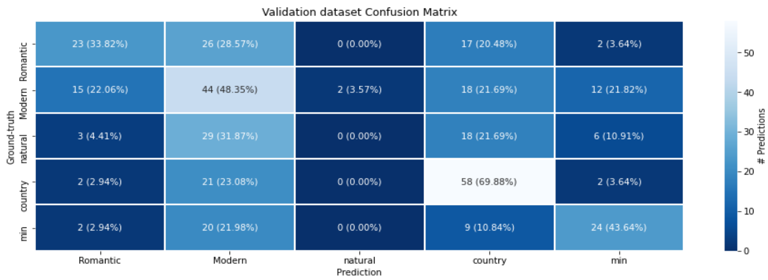

Figure 21 confusion matrix of validation data in simple CNN

This section describes the effect of data size and epochs on the performance of a simple CNN model. The model met trouble finding the pattern of modern and natural styles. The bias of data labeling and the size of data has a high probability to be the causes. Since this project targets testing accuracy, the simple CNN with 800 training data was chosen as the final model.

3.5 User Interface

In the user interface, the predicted style will be shown with a randomly selected corresponding interior image from the training dataset. So if the user thinks the style shown in the image is not his favorite one, he could choose another image or at least know the style is not the one he hunts for.

Figure 22 demonstrates the file structure that a user downloads. This project displays the result with the jupyter notebook running on Colab where all libraries required are default installed. Starting from mounting google drive, the classifier predicts the image user imported and the result is displayed with a corresponding image randomly selected from the training dataset.

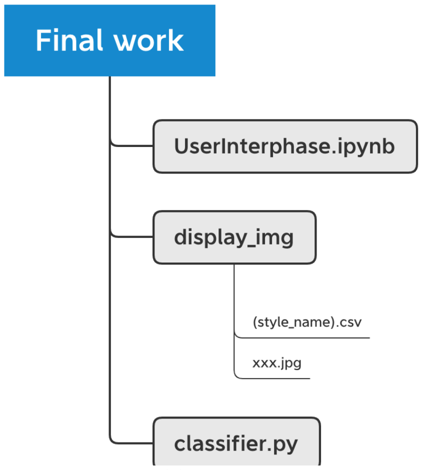

Figure 22 File structure

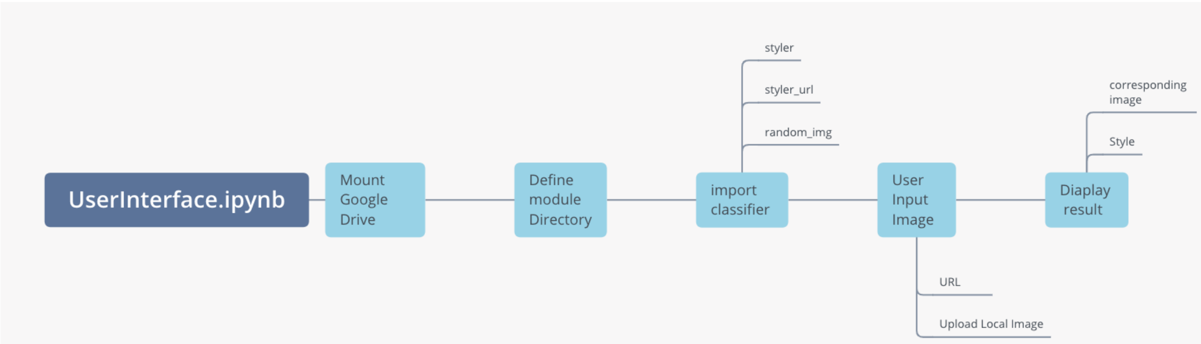

Figure 23 User Guide

4. Conclusion

This paper proposed an approach to deal with different sources of training data and test data. Extract common features of training data and test data before starting the model training process. Within all color features, qualified HSV histograms distinguish styles of image best. The goal of this project was to help people figure their favorite interior style by inferring interior style from any given colorful image. By combining color feature extraction with CNN, the model reaches 50% accuracy of the test dataset. However, the model had an imbalanced performance on styles, especially the natural style. The most likely cause was the quality of the datasets since they were labeled by non-professions in interior design. For further study, this project suggests importing more common features such as shape and texture. For example, both industrial and modern style has black color while the shape of black is usually in the block with industrial style and inline within the modern style. If a professional of interior design would like to label the raw images, the performance would improve significantly.

References

1. Mingxing Tan, Quoc V. Le. (2020, Sep 11). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. International Conference on Machine Learning. Retrieve from https://arxiv.org/abs/1905.11946
2. Jinsun Kim and Jin Kook Lee, 19 October 2020, Stochastic Detection of interior Design Styles Using a Deep-Learning Model for Reference Images
3. Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanekas, Qing Ye, Yuzhong Huang, Tui Tang, Stefan Leutenegger, 2918, InteriroNet: Mega-scale Multi-sensor Photo-realistic Indoor Scenes Dataset
4. Inside Airbnb.(2021, Feb). Retrieve from http://insideairbnb.com/get-the-data.html
5. Haldar, S. (2019, July 23). *DeepArtist : Identify Artist from Art*. Kaggle. Retrieved from https://www.kaggle.com/supratimhaldar/deepartist-identify-artist-from-art
6. Lee, J (2017, July 28). *Why One-Hot Encode Data in Machine Learning*. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/
7. Sought: https://www.programmersought.com/article/81694469853/
8. Sharma, M. (2019, Nov 1). *Histograms in Image Processing with skimage-Python*. Retrieved from Towards Data Science: https://towardsdatascience.com/histograms-in-image-processing-with-skimage-python-be5938962935
9. Thevenot, A. (2020, May 15). *Understand and Visualize Color Spaces to Improve Your Machine Learning and Deep Learning Models*. Retrieved from Towards Data Science: https://towardsdatascience.com/understand-and-visualize-color-spaces-to-improve-your-machine-learning-and-deep-learning-models-4ece80108526
10. Wikipedia. (n.d.). *RGB color model*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/RGB_color_model
11. Simon, H.A. Style in design. In Spatial Synthesis in Computer-Aided Building Design; Eastman, C.M., Ed.;Wiley: New York, NY, USA, 1975; pp. 287–309.
12. Jinsung Kim, J.-K. L. (2020, September 16). Stochastic Detection of Interior Design Styles Using a Deep-Learning Model for Reference Images. *Applied Science*.