Fall 2019

# Predictive modeling of webpage aesthetics

Ang Chen



Department:

# PREDICTIVE MODELING OF WEBPAGE AESTHETICS

by

ANG CHEN

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN INFORMATION SCIENCE & TECHNOLOGY

2019

Approved by:

Dr. Fiona Fui-Hoon Nah, Advisor
Dr. Keng Siau
Dr. Langtao Chen

# ABSTRACT

Aesthetics plays a key role in web design. However, most websites have been developed based on designers' inspirations or preferences. While perceptions of aesthetics are intuitive abilities of humankind, the underlying principles for assessing aesthetics are not well understood. In recent years, machine learning methods have shown promising results in image aesthetic assessment. In this research, we used machine learning methods to study and explore the underlying principles of webpage aesthetics.

Keywords: Aesthetics, Machine Learning, Webpage Aesthetics

## ACKNOWLEDGMENTS

This thesis is completed under the guidance of my committee members: Dr. Fiona Fui-Hoon Nah, Dr. Keng Siau and Dr. Langtao Chen. They provided valuable guidance and helped me get over difficulties. Their profound professional knowledge and rigorous scientific attitude have a great impact on me. This thesis would not have been completed without these amazing scholars.

I would like to express my most sincere and heartfelt gratitude to my advisor, Dr. Fiona Fui-Hoon Nah. Dr. Nah contributed vast time and effort in helping me become a researcher. She has been providing me with guidance and encouragement throughout this thesis. Her knowledge, attitude and spirit has impacted me not only in research but also in life. It is a wonderful experience doing research under her guidance.

At last, I would like to sincerely thank all professors for their hard work in helping me finish this thesis. I am so fortunate to have the best family, professors and friends who have been encouraging and supporting me through my master`s program.

# TABLE OF CONTENTS

Page

# LIST OF ILLUSTRATIONS

Page

# LIST OF TABLES

Page

# 1. INTRODUCTION

Aesthetics has been shown to have a dominant influence in e-commerce, such as in influencing buyers' purchase decisions (Postrel, 2001). Aesthetics also affects users' evaluations (Tractinsky et al., 2000) and preferences (Schenkman and Jonsson, 2000). Webpage design is an area that warrants attention to the assessments and dimensions of aesthetics. A good webpage design can bring many benefits, such as increased click rates, registration rates, subscription rates, volume of downloads, and conversions rates. From a business point of view, these factors are critical to revenue. When a group of websites offers similar services, users tend to choose sites that are visually more attractive than others (Touch et al., 2012). Research shows that aesthetics in web design is a major determinant of perceived credibility and trustworthiness (Fogg et al., 2003; McKnight et al., 2002). Aesthetics has also been shown to positively influence behavior, such as user performance and purchase intention (Reinecke et al., 2014; Moshagen et al., 2009; Bloch, 1995).

However, research on aesthetics is limited or lacking in the Human-Computer Interaction (HCI) area. The traditional HCI field has been mostly concerned with usability and functionality (Reinecke et al., 2014). Although designers are aware of the importance of aesthetics, design decisions are made mainly based on "inspiration" and "educated guesses" (Liu, 2003). It has also been shown that demographic differences, such as personality, gender, and age, can have an impact on aesthetic impressions (Moss & Gunn, 2009; Cyr et al., 2010; Reinecke & Bernstein, 2011; Wang, 2014).

Evaluating aesthetics is a highly subjective task. Different people can have different views on aesthetics. A beautiful webpage in one's eyes can be unpleasant in the eyes of another. Sometimes, designers may develop websites which they believe to be aesthetic but are not welcomed by users.

Most of the earlier research on aesthetics focus on design principles. The use of a quantitative modeling approach to study aesthetics is fairly scarce in the literature. Although the use of qualitative and interpretive approaches can provide deep insights into aesthetics, it is often not easy to apply their findings in practice. From the perspective of application, the use of quantitative methods could be easier to implement and test, and hence, can generate greater practical value. Also, it has been argued that general design principles may not apply in all contexts, especially from the perspective of aesthetics. On the one hand, the design criteria proposed by qualitative researchers often cannot be quantified or assessed quantitatively. On the other hand, the applicability of these criteria is based on the context. Besides, people's demographic differences and personal tastes can also have an influence on aesthetic impressions (Lindgaard et al., 2006; Martindale et al., 1990; Reinecke & Gajos, 2014).

With recent advances in machine learning and the boom and availability of data, machine learning techniques have become a powerful and efficient tool in computer vision. In the image recognition area, the convolutional neural network has been regarded as the biggest advancement. It has helped researchers in understanding complex phenomena and can potentially help us automatically assess aesthetics in HCI. In recent years, deep learning techniques have been shown to be better in assessing aesthetic quality of images than traditional methods that utilize webpage features associated with

aesthetics (Karayev et al., 2013; Lu et al., 2014, 2015a, 2015b; Dong & Tian, 2015; Kao et al., 2016; Kong et al., 2016; Mai et al., 2016; Jin et al., 2016; Wang et al., 2016a, 2016b).

However, there are differences between image aesthetics and web aesthetics. Image aesthetics are mostly used on photos and may not apply well to the area of web aesthetics. Although a webpage can be turned into an image file with a screenshot, there are many differences between the aesthetics of a webpage and an image. The composition of a webpage is generally more complicated and sophisticated than a photo. It contains many complex elements, such as text, borders, pictures, and even animations. Deep learning models recognize and learn objects based on the edges of them. There are far more edges between webpage elements in a screenshot than of a flower in a photograph. Despite many differences between webpage aesthetics and image aesthetics, there could be commonalities between them. According to a previous study (Lindgaard et al., 2006), the aesthetic impression of a webpage can be formed within 50ms. A webpage can be viewed as a whole image, and hence, we are interested to assess the performance of deep learning models that have been used in image aesthetics and apply them in the context of screenshots of webpages.

Traditional assessments of webpage aesthetics draw on specific measures of a webpage to predict aesthetics. The advantage of this approach is that they can be quantified and used directly in webpage design to guide web designers when a comprehensive set of such measures is established in the literature. The disadvantage of this approach is that identifying and determining these measures are often difficult, require deep domain knowledge, and tend to miss other important measures. Further, the

extraction method of some measures is very complex, which is often difficult to apply directly in rapid web evaluation applications.

In the deep learning area, Convolutional Neural Network (CNN) directly extracts abstract measures from image input. The advantage of this method is that it can make out key aspects of a picture. This approach effectively compensates for the shortcomings of traditional aesthetic assessment methods by treating webpages as images in the web assessment applications. While this approach sounds promising, it has its drawbacks. On the one hand, the neural network is a black box, which means we have no way of knowing how it rates a web page as beautiful or ugly. On the other hand, this method can potentially only train a model to predict webpage aesthetics, and often fails to come up with principles of aesthetics that the traditional approaches do. Whether one approach is superior to another is still debatable. We can only say that each approach has its advantages. However, attempts can be made to compare these approaches in assessing and understanding webpage aesthetics.

In this thesis, we propose to use machine learning methods to study aesthetics. First, we will review related work, which includes fundamentals about aesthetics and machine learning methods used in this study. Second, we will describe the methodology for the study. Third, we will present the data analysis and the results. Next, we will present the shortcomings of the study and directions for future research. Finally, we will summarize the results and provide conclusions for the thesis.

# 2. RELATED WORK

In this section, we will briefly introduce related work in the literature, which includes webpage aesthetics, aesthetics and user preferences, computational interface aesthetics, traditional webpage aesthetic assessment, and webpage aesthetic assessment using deep learning methods.

## 2.1. WEBPAGE AESTHETICS

Aesthetics or beauty is one of the three basic requirements of architecture according to Vitruvius, the first systematic theoretician of architecture (Kruft, 1994). The three requirements are:

- Firmatis (Durability) – Architecture should be robust and remain in a good condition.
- Utilitas (Utility) – Architecture should be useful and functioning well when people are using it.
- Venustatis (Beauty) – Architecture should be delightful for people and their spirit.

Among the three requirements, the theory of Venustatis (or beauty) is more complicated. Vitruvius believed that human's "beauty" is the "truth of nature", and "nature's designs" are harmonic and symmetric.

The terms, harmonic and symmetric, are closely associated with a term in aesthetics--balance. Balance is an important principle in aesthetics. Balance means that the visual weights of the page elements are evenly distributed throughout the page. When

a webpage is designed to be well balanced, users perceive equilibrium psychologically (Lindgaard, 1999).

## 2.2. AESTHETICS AND USER PREFERENCES

Preferences are strongly influenced by aesthetics (Tractinsky, 2004). First impressions are important. According to research, aesthetic impressions – the spontaneous emotional responses based on visual preferences – can seriously influence whether we perceive a product as useful or not (Sonderegger & Sauer, 2010). An aesthetic impression is typically formed within 50 to 500ms during the first contact, and it is persistent once it has been formed (Lindgaard et al., 2006; Tractinsky et al., 2006). Because of these characteristics of aesthetic impressions, researchers often use static screenshots of webpages to test whether users like those webpages (Reinecke et al., 2013).

User preferences, as measured by evaluations of various aspects of a system or by expressing attitudes towards the system, may not necessarily correspond to actual decisions to use or buy one system over another (Ben-Bassat et al., 2006). Although a user's final choice is influenced by many other factors (e.g., economics, environment, culture), the user’s preference based on its aesthetic impression is weighed heavily and is hard to overcome in the decision process (Russo et al., 1998). If people develop an initial preference for more attractive designs, judgments of objective measure information may shift in the direction of more attractive products (Hoegg et al., 2010).

Although scholars have been trying to come up with universally applicable principles of aesthetics, many factors can significantly impact these principles. For

example, due to individual and cultural differences, aesthetics could be perceived differently by different people. Some people prefer concise designs while others prefer designs involving artistic or special effects, which makes aesthetic assessments somewhat subjective. Besides, websites frequently update their design styles to keep up with the latest trends in website design. Websites today are very different from those in the 90s. However, we do believe that it is useful to develop a better understanding of the principles of webpage aesthetics. In addition, machine learning is powerful in learning patterns based on data, which matches the characteristics and requirements of this research.

## 2.3. COMPUTATIONAL INTERFACE AESTHETICS

Computational interface aesthetics is a field of study aimed at developing a computation model for the aesthetic quality of interfaces. Past HCI studies have focused on coming up with universal aesthetics principles, and considerations of computational aesthetics have been largely ignored. Although aesthetic principles are useful in guiding designers in their work, they do have limitations. For example, individuals can have different aesthetic impressions when perceiving aesthetics (Lindgaard et al., 2006; Martindale et al., 1990; Reinecke & Gajos, 2014). Thus, it is appropriate to personalize design when targeting a specific group of users. More boldly speaking, if the technology is mature enough, the design should modify and improve itself with each user.

It can be difficult to assess aesthetics objectively since aesthetics evaluation is expected to be subjective. Different people may have different views on what aesthetics is, which suggests that there will be large intra-class differences in aesthetic perceptions

(Jin, 2016). This problem is also challenging for machine learning algorithms since a large intra-class variance will significantly bring down the performance of regression or classification results.

The main challenge in the traditional aesthetics quantifying field is to evaluate aesthetics from all aspects. Although more and more measures related to webpage aesthetics are found, the significance of each measure is not the same. Researchers have focused on two of the most striking measures: colorfulness and visual complexity (Reinecke & Gajos, 2014).

Another challenge in computational interface aesthetics is that it requires knowledge from multidisciplinary areas such as mathematics, computer science, human-computer-interaction, art, design, psychology, and so on. Besides, researchers from different knowledge domains prefer to explain aesthetics based on their respective expertise, which makes it difficult to reach a consensus in aesthetic assessments.

## 2.4. TRADITIONAL WEBPAGE AESTHETIC ASSESSMENT

Using aesthetic measures has been the main approach in the area of computational aesthetics. Researchers have been focusing on using aesthetic measures (i.e., aesthetic rule-based features) to assess and explain aesthetics. According to Jin et al. (2016), scholars in this area usually research using the following three steps:

1. Collect or design interfaces according to the needs of the research, then conduct psychological experiments by having subjects assess the aesthetic quality of interfaces. The results of the assessment are generally presented in the form of an aesthetic score or class, such as “low” or “high”.

2. Craft measures or design principles such as rule of thirds, visual balance, and rule of simplicity. Some researchers use generic image features such as low-level image features, Fisher vectors, and bag of visual words to predict image aesthetics. The source of these papers can be found in Jin et al.'s (2016) reference section.
3. Build a statistical model or use machine learning models such as Support Vector Machine and Random Forest to predict the aesthetic quality. The steps include training the model based on the objective measures crafted and the subjective aesthetic assessment collected, then use the model to predict the aesthetic quality or help explore the relationship between the measures and aesthetic quality. These methods are usually regarded as "white boxes" since they can explain or show the relationship between the independent and dependent variables.

Measures of aesthetic dimensions or features are used as independent variables to assess the dependent variable, aesthetic quality. Although aesthetics is a fairly subjective concept, quantitative aesthetic measures do exist to quantify various aesthetic parameters and measures of screen layouts.

Ngo's theory (Ngo et al., 2003) is widely used as the baseline for measures of a layout. There are 14 measures in Ngo's work (2003): Balance (BM), Equilibrium (EM), Symmetry (SYM), Sequence (SQM), Cohesion (CM), Unity (UM), Proportion (PM), Simplicity (SMM), Density (DM), Regularity (RM), Economy (ECM), Homogeneity (HM), Rhythm (RHM), as well as Order and Complexity (OM). The details of their formulas and computations are described in their original work (Ngo et al., 2003).

Aesthetic measurement application (AMA), that was developed by Zain et al. (2008) to automatically assess aesthetics, uses 6 measures, i.e., BM, EM, SYM, SQM, RHM, and OM, that are based on Ngo's work. The aesthetic scores which were given by their model closely matched the rankings provided by the users. Altaboli and Lin (2011) found that three measures, BM, UM, SQM, have significant effects on perceived interface aesthetics. Maity and Bhattacharya (2017) found 9 measures, i.e., BM, CM, EM, HM, PM, RM, SQM, SYM, and UM, to be statistically significant in predicting aesthetics of webpages. However, they used a classification model instead of a linear regression model. Thus, the weights of these measures are not known and numerical evaluations cannot be offered by their study.

In our research, we will carry out a study of predictive modeling, linear and non-linear regression analysis, and exploratory analysis. Since we will be using a number of machine learning techniques and it is customary to refer to independent variables as features in the machine learning terminology, we will refer to measures of aesthetic dimensions as aesthetic features, or simply as features.

## 2.5. WEBPAGE AESTHETIC ASSESSMENT USING DEEP LEARNING

Boosted by the huge amount of data generated and vast improvement of computation capability, deep learning techniques have been greatly improved. In recent years, deep learning has gained tremendous success in computer vision, such as object recognition, object detection, and image classification (Jin et al., 2016; Szegedy et al., 2015; He et al., 2016).

However, deep learning techniques have rarely been used in the area of webpage aesthetics. One related work is by Dou et al. (2019) who used a deep neural network composed of 5 convolutional layers, 2 max-pooling layers, 2 fully connected layers, and a regression layer to predict the aesthetic score of images of webpages. In addition, the transfer learning technique was applied to increase the performance of the aesthetic scoring model.

## 2.6. MODEL DICTIONARY

This thesis applies machine learning methods to assess the aesthetics of webpages. Many scholars have applied these methods to natural photographs (Datta et al., 2006; Wong & Low, 2009; Wu et al., 2010; Faria et al., 2013).

**2.6.1. Ordinary Least Squares Model.** The Ordinary Least Squares (OLS) model is a statistical model of linear regression. It is used to estimate the relationship between one or more independent variables and the dependent variable. The basic idea is to minimize the sum of squares of the difference between the actual value and the model predicted value of the dependent variable. The OLS model can search the best matched parameters by minimizing the sum of squared errors. For a detailed introduction, please refer to the work of Hutcheson (2011).

**2.6.2. Decision Tree Model.** The decision tree model is a tree-like model. The ‘tree’ consists of nodes and branches. There are two types of nodes: internal nodes and leaf nodes. Each internal node represents a feature, and each leaf node represents a label or target. Branches represent the output of that feature attribute on a range of values. For details of decision trees, please refer to the paper by Quinlan (1986).

The decision-making process of a decision tree starts from the top-most internal node which is also called the root node. An observation is fed to the decision tree model, after which the observation will be directed to a node by a branch according to the specific feature value. This process continues until the leaf node is reached and the decision is made.

Compared with other machine learning algorithms, decision trees have advantages in the following aspects:

1. It is a white-box model that is easy to understand, which means we can understand and interpret the logic and meaning behind the decision trees.
2. Data preparation is often simple or unnecessary for the decision trees. Other techniques often require steps such as changing data types or removing redundant or blank attributes.
3. The computational complexity is not high; the output results are easy to understand and visualize.

It also has some disadvantages:

1. Easy to overfit. Sometimes decision trees can become so complex that they could not generalize to real-world data. Setting a minimum number of samples for the leave nodes or limiting the maximum tree depth can help to minimize overfitting.
2. Decision trees give higher preferences to classes with high sample quantity. Thus, balancing the dataset is recommended if some classes dominate in quantity.

**2.6.3. Random Forest Model.** A random forest model is a ‘forest’ which ensembles many decision trees. There should be no correlation or a very weak correlation between decision trees in the random forest. The output is determined by the prediction

which is voted by most of the decision trees. A detailed explanation of the random forest model can be found in the paper of Breiman (2001).

Each decision tree in the random forest may be weak. However, the combination of them makes the random forest strong. Each decision tree can be regarded as an "expert" in their field (m features out of M total features) and there are numerous "experts" in the random forest. When a new problem (new data) is given to the random forest, these "experts" will vote for answers based on their perspectives. These characteristics of the random forest are also very suitable for the problem we want to solve in this thesis because random forest models can handle non-linear relationships, which may exist between the aesthetic features and the aesthetic ratings. In the field of computational aesthetics, experts and scholars often put forward a variety of measurement methods and often fail to reach a unified understanding of a universal method. By including all the measurements, it is appropriate for experts to vote on the best solution.

Here is how a random forest is generated:

1. N represents the number of training samples. M represents the number of total features. m represents the number of features that are randomly selected from M features.
2. For each decision tree, a specific training set will be formed by sampling N cases from the original training set. Each case sampled will be put back to the sample pool, which means the same case could be sampled several times to form a specific training set. The cases that are not sampled will be used as the testing set

for evaluation and prediction. This sampling method is also called bootstrap sampling.

3. For a specific decision tree in the random forest, some random features are selected from M features for a non-leaf node. An optimal feature will be selected from the features that are used for the node.
4. Repeat steps 2&3 to form numerous decision trees, then the random forest is generated. A training set (bootstrap sampling) is formed by sampling N training cases (samples) with the method of sampling back, and the unsampled use cases (samples) are used to make predictions and evaluate the error.

   The random forest model has the following advantages:

   - Not easy to overfit data due to the randomness brought by the random forest.
   - It can handle data with many features without feature selection.
   - It can give estimates of the importance of features.
   - It can fit non-linear relationships.

These advantages make the random forest a very suitable model for our research. We do not have to be concerned about non-linear relationships (as the model will take care of them), which is often the biggest concern with regression models. Further, the random forest model can pick features that are important, which means the random forest model is able to identify the important features.

**2.6.4. Gradient Boosting.** Gradient Boosting is a machine learning technique that integrates weak models and iteratively makes a stronger model. These weak models are

typically decision trees. To investigate specific details of gradient boosting, please refer to the original paper by Friedman (2001).

We discussed the random forest model above. Gradient boosting has much similarity with the random forest. Both models involve integrating weak models into a strong model. However, the random forest ensembles all the weak models in a parallel way, while gradient boosting does it iteratively. For example, we use decision trees as weak models to make a strong gradient boosting model. When we built the first decision tree, it may not work very well. When we try to build the second decision tree, we selectively build a better one than the first we built, which is the idea behind gradient boosting.

**2.6.5. Artificial Neural Network (ANN).** Artificial neural network (ANN) is a machine learning technique that simulates the human brain to realize artificial intelligence. It requires setting up the structure of a neural network for learning. An early attempt at ANN was to make them perform tasks that were difficult for traditional machine learning algorithms. It can be applied to various tasks such as speech recognition, computer vision, natural language processing, board games and video games, medical diagnosis and so on.

Since artificial neural networks are based on the imitation of biological neural networks, this approach is closely related to cognitive science and neuroscience. Computational aesthetics falls well into the category of this field.

**2.6.5.1. Architecture of artificial neural network (ANN).** To better understand the ANN, we will first start with a classic architecture of it. Figure 2.1 shows a basic three-layer neural network. The yellow nodes (i.e., left-most layer) is the input layer, the

red node (i.e., right-most layer) is the output layer, and the green node is the middle layer, which is also known as the hidden layer. In Figure 2.1, the input layer has three input units, the hidden layer has one hidden unit (also called a neuron or a perceptron), and the output layer has one output unit.

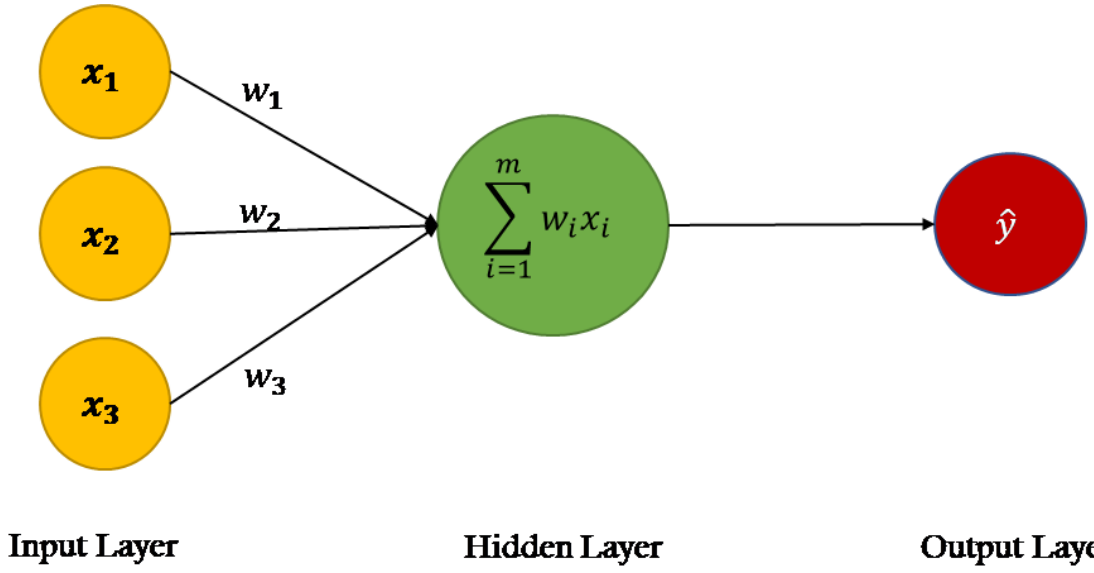



Figure 2.1. Basic Unit of an Artificial Neural Network — Artificial Neuron (Vaibhav, 2018).

**2.6.5.2. Input layer.** The input layer is designed to connect input information with the neural network for processing. Each circle in the input layer represents a feature or a channel of the input information. The input data for the neural network can come in various forms. For example, it can be a feature of whether your house comes with a garage in a property prices prediction problem. In computer vision, a circle in the input layer can represent the value of a pixel on a specific location of an image. You do not need to manually extract features from data as the neural network will try to automatically learn the patterns from the input information. In other words, these neural

networks do not need to be fed with inputs of higher-level features, such as complexity and colorfulness, but can still make predictions on aesthetic quality.

**2.6.5.3. Hidden layer.** In Figure 2.2, the working of a biological neuron is shown. The purpose of showing the working of a biological neuron is to demonstrate how an ANN is modeled after the working of biological neurons.

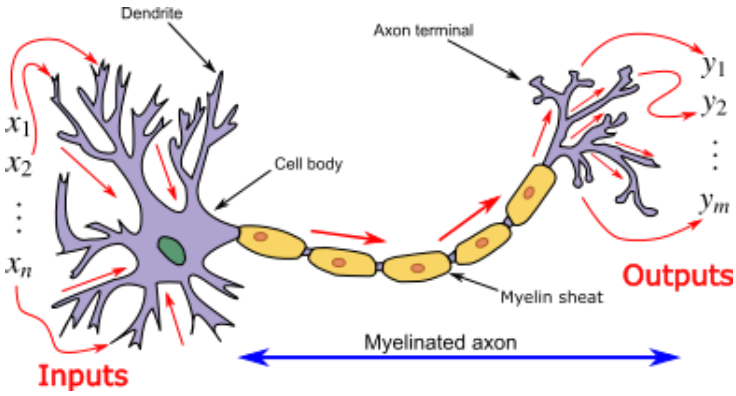



Figure 2.2. How a Biological Neuron Works (Wikipedia Contributors, 2019).

Neurons have been studied and known by biologists since 1904. A neuron usually has multiple dendrites, which are mainly used to receive incoming information. There is only one axon but there are many axon terminals at the end of the axon that can send messages to many other neurons. The axon ends make connections with dendrites of other neurons and transmit signals.

In Figure 2.3, the working of an artificial neuron is shown. An artificial neuron is made to simulate the biological neuron. The directed arc is the simulation of dendrite-synapse-axon. From Figure 2.3, we can see that the inputs ($x_1$, $x_2$ … $x_m$) are multiplied by corresponding weights ($w_1$, $w_2$ …$w_m$) and then sent to the neuron. This process simulates

the signal interaction between biological neurons. The weight of the directed arc indicates the strength of the neural signal interaction between interconnected artificial neurons (Feng, 2015). When the neuron receives the weighted signals, it sums them up and adds a bias factor, then the summed signal will be sent to an activation function $\varphi(\cdot)$ for processing and an output y will be generated. This y value will be transmitted to the next neuron as input or directly used as output if it is in the output layer.

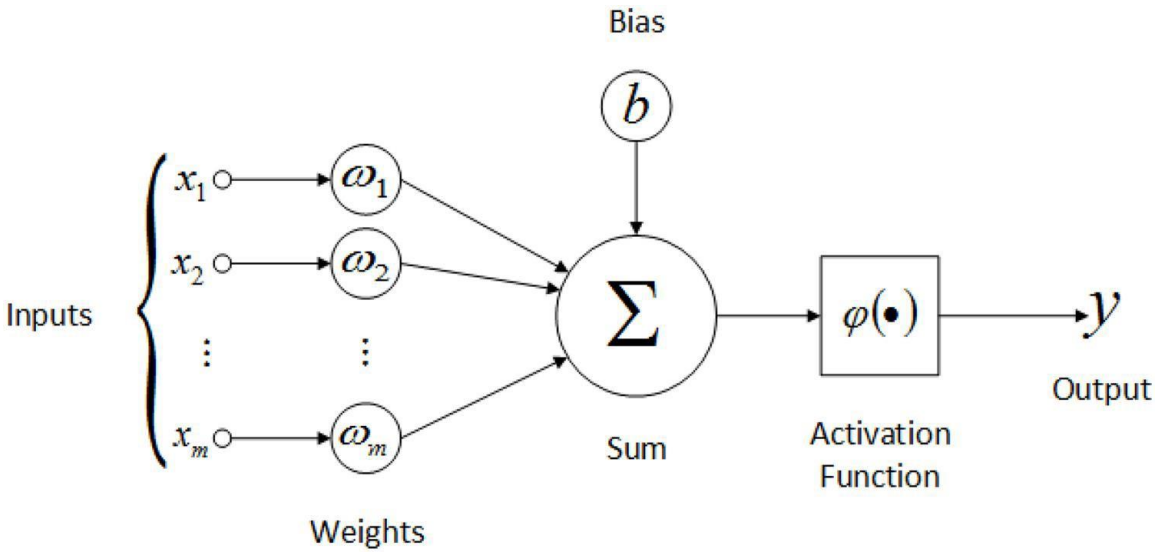



Figure 2.3. Artificial Neuron (Jayesh, 2018).

**2.6.5.4. Output layer.** The output layer summarizes all the information that has been processed by the previous layers and then generates one or more outputs depending on your specific problem. In this study, we formulate the problem of predicting aesthetics as a regression problem. Thus, the last layer will be using only one neuron to give a numerical score.

**2.6.6. Deep Neural Network (DNN).** A deep neural network is formed when a simple neural network has more hidden layers. In general, a simple neural network is enough to solve many simple regression or classification problems. But when the problem is more complex, the neural network needs a more complex structure to improve its ability to deal with the problem.

The complex structure makes deep neural networks better at fitting, but it also increases the demand for data volume. If the data is insufficient, overfitting is likely to occur. Overfitting makes the machine learning models perform well on the training data but poorly on test data or real-world data. As the number of data points increases, deep neural networks can surpass most other machine learning models. Thus, the industry may desire neural networks to be as deep as possible. In our case, however, the lack of data is a problem, as will be discussed in the next two sections.

Deep neural networks have been extensively studied and used in modern times, and scholars have developed various deep neural networks to improve the performance of the network and to cope with various complex problems. Other types of deep neural networks include convolutional neural networks and recurrent neural networks. Convolutional neural networks are often used for computer vision problems, while recurrent neural networks are used for natural language processing problems. This thesis focuses on predictive modeling of webpage aesthetics, which makes convolutional neural networks particularly appropriate. Thus, we will briefly introduce convolutional neural networks next.

**2.6.7. Convolutional Neural Network (CNN).** As a class of deep neural networks, a convolutional neural network (CNN) is often used in the field of computer vision due to its superior performance.

A classical CNN consists of one or more convolutional layers and fully connected layers at the top. These structures enable the CNN to more efficiently utilize the two-dimensional structure of the input data. As an example, a CNN sequence is shown in Figure 2.4.

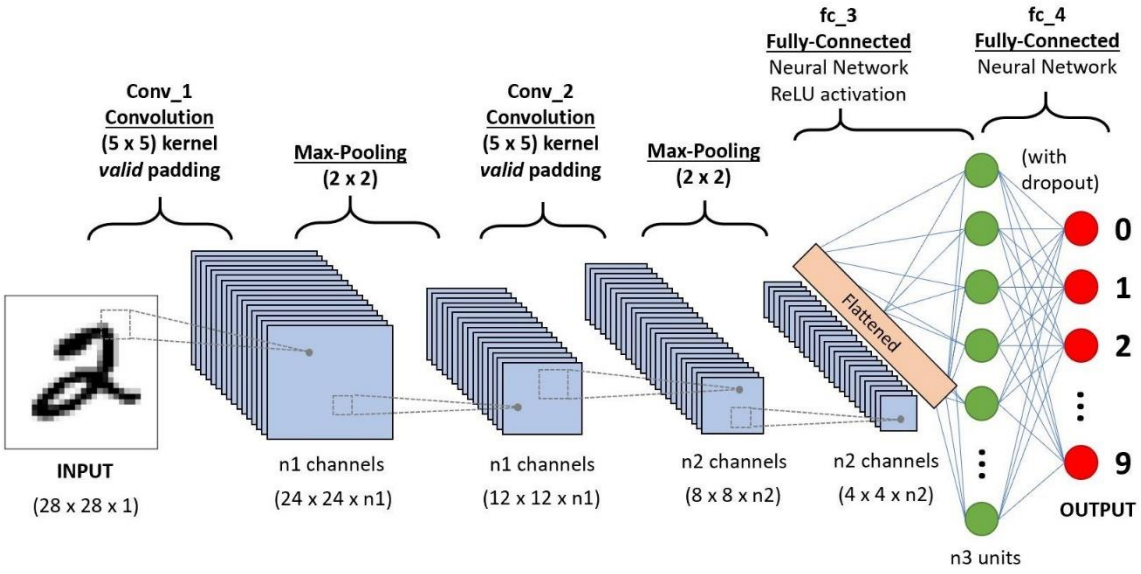



Figure 2.4. A CNN Sequence to Classify Handwritten Digits (Sumit, 2018).

Figure 2.4 is a simple plot of a CNN sequence classifying handwritten digits. This CNN is designed to recognize numbers in hand-written digit images. It consists of two convolutional layers, each followed by a max-pooling layer. After the second max-pooling process, the "image" is flattened into a one-dimensional vector that feeds into a fully connected layer. A drop-out processing is carried out to avoid the overfitting

problem before the processed information is passed to the output layer, which gives the probability of being one of the numbers from 0 to 9.

In computer vision problems, the CNN uses images as input data. Compared to other machine learning models, the preprocessing cost required on CNN is much lower. For a general machine learning model, people first need to manually design or select features for the model to learn. The CNN directly takes image data as inputs into the network, and the network learns to assign weights to various aspects/objects in the images.

For example, in an animal image recognition program, CNN can identify images containing cats by analyzing sample images that have been manually labeled "cat" or "no cat" and using the results to identify cats in other images. They did so without being “told” that cats, for example, have the following features: fur, tails, whiskers and a cat-like face. Instead, they automatically generate recognizable features from processing the training data. However, predicting aesthetic quality can be challenging for CNN because it is subjective and hard to describe. People can easily imagine what a cat looks like but it is much harder to explain, imagine, or predict aesthetic quality.

**2.6.7.1. Convolutional layer.** Convolutional layers are the core layers of CNN. They are designed to extract different features from the input data by conducting convolution computations. Although conventional neural networks can achieve this goal, it requires an incredibly high number of parameters and it incurs some computation cost. Imagine the cost of which you need 10,000 hidden units in a hidden layer to fully connect to an image of resolution of 100*100.

The convolutional layers can efficiently extract features while keeping the computation cost and parameter number at a lower level, which allows you to have a deeper neural network. The convolutional layers which are at shallow levels may only extract some low-level features, such as edges, lines and corners, etc. Convolutional layers that are deeper in the networks can iteratively extract more complex features from these low-level features.

**2.6.7.2. Pooling layer.** Pooling is another important concept in CNN, which is a form of non-linear reduced sampling. There are several types of pooling functions such as average pooling and max pooling. Max-pooling is the most common form of pooling. It divides the input image into multiple rectangular regions and outputs the maximum value for each sub-region. This characteristic of pooling directly reduced the data volume for processing. For example, a 32*32 input is going through a pooling layer. If the size of the pooling layer filter is 2×2, the size of the output data after the pool layer processing is 16×16, which means that the existing data volume is suddenly reduced to 1/4 of the pre-pool size.

Intuitively, this mechanism works because once a feature is discovered, its precise location is far less important than its relative position to other features. The pooling layer will continuously reduce the space size of the data, so the number of parameters and calculation amount will also decrease, which also controls overfitting to a certain extent.

In addition to maximum pooling, the pooling layer can also use other pooling functions, such as average pooling or even L2-norm pooling. In the past, average pooling has been widely used, but recently it has become less common due to the increased performance of maximum pooling in practice.

**2.6.7.3. Fully connected layer.** The fully connected layer (FC layer) is the same layer as you can see in a conventional neural network (non-convolutional). Each neuron in an FC layer is connected to all units in the previous layer.

Convolutional neural networks are not all about convolutional layers or pooling layers. Many of the convolutional neural networks still use layers that appear in conventional neural networks. The FC layer usually appears at the top layers of a convolutional neural network for reasoning and concluding the results from previous layers. Due to the high increase in parameters brought by the FC layers, some advanced CNN architectures use pooling layers to replace some of the FC layers. Doing this can effectively reduce the computation cost. However, some researchers found that the FC layers can work as a "firewall" in transfer learning (Zhang et al., 2017). Transfer learning refers to using a trained CNN to learn another dataset with its existing trained weights. Zhang et al. (2017) found that convolutional neural networks with FC layers have better performance in transfer learning compared to convolutional neural networks without FC layers. The FC layers gave the CNN good knowledge transferability so that the trained CNN can adapt well on a slightly different dataset.

This discovery is of great reference value to our research. Because some of the CNNs we use are pre-trained on some professional camera photo datasets, and we want to migrate these models to the webpage aesthetics screenshot dataset. However, we should pay more attention to the use of FC layers.

**2.6.8. MobileNet.** MobileNet is a class of convolutional neural network architecture that has the feature of being lightweight. This feature of MobileNet makes it

more suitable for applications in mobile phones, drones or other devices lacking strong computation power. This architecture was introduced by Google.

Although computation volume has been greatly reduced by convolutional networks, it is still relatively large for some devices. MobileNet uses depth-wise separable convolution instead of standard convolution, which significantly reduced the number of parameters. For the details of depth-wise convolution, please refer to the work of Howard et al. (2017).

While MobileNet has the advantages of high efficiency and low complexity, it also sacrifices some accuracy. However, the loss of model accuracy is negligible compared to the value MobileNet brings to some devices of poor computation power. The introduction of MobileNet is valuable in the mobile market.

**2.6.9. NasNet (Neural Architecture Search Network).** The full name of NasNet is Neural Architecture Search Network. The NasNet is a class of convolutional neural network architecture that was introduced by researchers from Google Brain (Zoph et al., 2018). The main idea of NasNet is to search for the architecture in a small dataset and transfer it to a large dataset. In their original paper, they use a small dataset (CIFAR-10, a dataset with 60,000 32x32 color images in 10 classes) to automatically design the convolutional neural network and then use transfer learning to adapt the neural network to a large dataset (i.e., ImageNet, a very large dataset with more than 14 million images in numerous classes).

**2.6.10. Inception Neural Network.** Inception neural network is a neural network architecture which was designed by Szegedy et al. (2015) from Google. Its general form is shown in Figure 2.5.

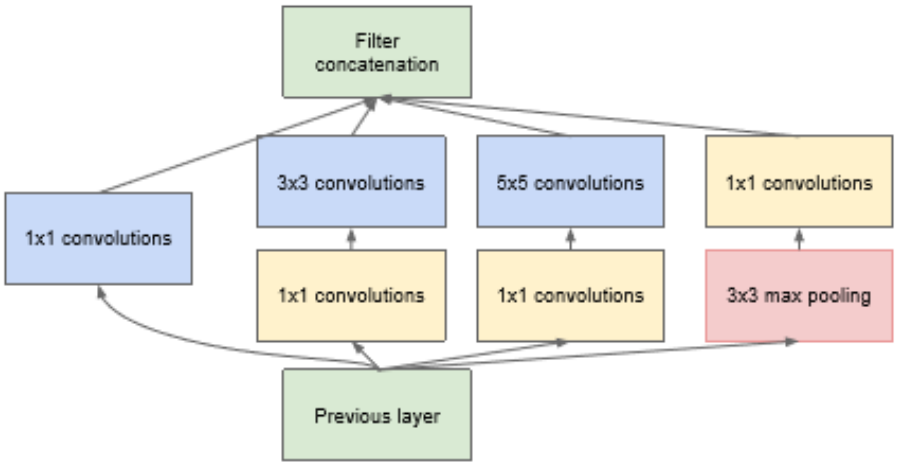



Figure 2.5. Inception Module, with Dimensionality Reduction (Szegedy et al., 2015).

As shown in Figure 2.5, the inception module used a max-pooling layer as well as several convolutional layers of sizes such as 1*1, 3*3, and 5*5. The 1*1 convolutional layers are mainly used to reduce the magnitude of computation. The variety of scales and types of layers also follow the intuition that visual information should be processed through various channels and then aggregated together (Szegedy et al., 2015).

An inception network is made by stacking the inception modules upon one another. It does not mean the inception network purely consists of the inception modules. The inception network can still contain the traditional convolutional layers. The inception modules are used more often in the higher layers, due to the infrastructural inefficiency in the current implementation.

## 2.7. AESTHETIC METRICS

Aesthetic metrics refer to the features for predicting aesthetics. The aesthetic metrics we are using are from previous researchers' work (Reinecke & Gajos, 2014). The aesthetic

metrics include color-related metrics, space-based decomposition metrics and quadtree decomposition metrics. In this section, we will provide a brief explanation of the metrics to be used.

**2.7.1. Color.** Color is one of the most important factors in human-computer interaction. Color has been shown to affect emotion (Coursaris et al., 2008; Lindgaard, 2007), perceived trustworthiness (Cyr et al., 2010; Kim & Moon, 1998), users' loyalty (Cyr, 2008) and purchase intention (Hall & Hanna, 2004). When we see a color, we generally see it as a whole. However, the colors we see contain rich information that affects our lives and perceptions in subtle ways.

**2.7.1.1. W3C colors.** The value of a W3C color feature is the percentage of pixels that are most similar to one of sixteen colors defined by the W3C system. For details of W3C colors, please refer to https://www.w3.org/TR/html401.

**2.7.1.2. Hue, saturation and value.** Average pixel values for hue, saturation, and value in the HSV (hue, saturation and value) color model. Researchers have tried various ways to define the color we perceive. One classical method is using the HSV color model to describe a color. H stands for hue, which tells what type of color it is. S stands for saturation, which represents the intensity of the color. V stands for value, which represents the visually perceived brightness of the color.

**2.7.1.3. Colorfulness.** There are two colorfulness metrics used by Reinecke et al. (2013). The first was put forward by Yendrikhovskij et al. (1998). This colorfulness metric is computed by summing the average saturation and standard deviation across an image. To calculate the saturation, the chromaticity of the image is divided by the brightness. The brightness and chromaticity are defined using the principles from CIELab

color space. For more information about CIELab, please refer to https://www.xrite.com/blog/lab-color-space.

The second colorfulness metric was proposed by Hasler and Suesstrunk (2003). This metric also follows the principles of CIELab color space. To calculate this colorfulness metric, the trigonometric length of the standard deviation of an image needs to be computed as well as the distance between the gravity center and the neutral axis. As the last step, this metric computes the weighted sum of the length and the distance.

**2.7.2. Space-based Decomposition.** The space-based decomposition is a technique that is used to divide the webpage space into different parts along the vertical and horizontal directions. Space-based decomposition splits a page by separating its components along horizontal and vertical spaces on the page. The result of the decomposition is a tree that represents the website. The root of this tree is the entire webpage. The first and second layers can be the major components such as the title and body. The lower layers represent sub-parts of the higher levels.

**2.7.2.1. Number of leaves.** It is the final number of leaves computed by space-based composition (Ha et al., 1995). A webpage is recursively divided by space-based decomposition until there is no visible separator of space or a leaf has become too small.

**2.7.2.2. Number of image areas.** Calculates the number of leaves identified as individual images by the algorithm. Adjacent images can be counted as an image area.

**2.7.2.3. Number of text groups.** Calculates the number of groups that are identified as text. An individual group can be a single word, a single or multiple lines of text, or a paragraph.

**2.7.2.4. Text area and non-text area.** These two features estimate the areas that are recognized as text or non-text based on the results of space-based decomposition.

**2.7.3. Quadtree Decomposition.** It is a decomposition technology that splits webpage based on entropy. The entropy reflects the complexity of an area. The entropy is usually calculated based on specific standards such as the size and intensity of the area. The quadtree decomposition repeatedly divides the webpage into subparts (leaves) along the horizontal and vertical directions. A threshold of area entropy will be given to indicate to the decomposition algorithm when to stop splitting the webpage. For more details of this approach, please refer to the research of Zheng et al. (2009).

**2.7.3.1. Number of quadtree leaves.** Calculates the number of leaves that have the threshold amount of entropy.

**2.7.3.2. Symmetry.** This metric evaluates the symmetricity of the layout of leaves.

**2.7.3.3. Balance.** This metric calculates whether there are equal number of leaves across the horizontal axis and vertical axis.

**2.7.3.4. Equilibrium.** This metric evaluates how the quadtree leaves are centered around the midpoint of an image.

## 2.8. PERFORMANCE METRICS DICTIONARY

Performance metrics are used to evaluate the effectiveness of trained machine learning models. Once the machine learning models have been trained using the training dataset, performance metrics will be used to measure how effective the model is using the validation dataset or testing dataset.

There are numerous performance evaluation metrics in the field. Different metrics apply to different situations. There are two main categories: evaluation metrics for classification and evaluation metrics for regression. Metrics for classification problems usually include accuracy, precision, sensitivity or recall, specificity, area under curve (AUC), and F1 score. Metrics for regression problems usually include mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R-squared ($R^2$). There are no strict limits on the use of these performance metrics. You can use metrics from regression problems on classification problems in some circumstances. But each metric has its best use and situation. In some cases, researchers even design unique metrics that are appropriate for their problems.

In this thesis, we aim at predicting an aesthetic score and used mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R squared ($R^2$) as our performance metrics. In this section, we will explain these performance metrics in more details.

**2.8.1. Mean Absolute Error.** Mean absolute error (MAE) is the average of the absolute difference between the predicted values and the actual values. The higher the MAE, the worse the model performance. Its formula is shown in Equation (1):

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \quad (1)$$

i means the $i^{th}$ sample; n is the number of samples in the dataset; $y_i$ is the actual value and $\hat{y}_i$ is the predicted value of the $i^{th}$ sample.

MAE is a score that weighs all differences equally. For example, the difference between 4 and 0 is twice the difference between 2 and 0. However, this difference will be

squared in the mean squared error (MSE) to be discussed next. MSE penalizes errors more than MAE. We will explain it in more details next.

**2.8.2. Mean Squared Error.** Mean squared error (MSE) is the average of the squared difference between the predicted values and the actual values. The higher the MSE, the worse the model performance. Its formula is shown in Equation (2):

$$\text{MSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \quad (2)$$

i means the i[th] sample; n is the number of samples in the dataset; $y_i$ is the actual value and $\hat{y}_i$ is the predicted value of the i[th] sample.

MSE is one of the most used metrics in regression problems. However, this metric often comes with some problems. On the one hand, it often overestimates the errors when the differences are mostly greater than 1. As given as an example earlier, the difference between 4 and 0 is twice the difference between 2 and 0. However, when using the MSE as the performance metric, the difference will be squared. On the other hand, when most differences are lower than 1, MSE will underestimate the errors. Thus, MSE is not a favorable performance metric when your data is "noisy" because a noisy dataset will typically have many outliers. These outliers are significantly different from most of the other observations, which further amplifies the disadvantages of overestimating MSE metrics.

**2.8.3. Root Mean Squared Error (RMSE).** Root Mean Squared Error (RMSE) is the square root of mean squared error (MSE). For explanations of MSE, please refer to the previous section. The formula for RMSE is shown in Equation (3):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (3)$$

i means the i[th] sample; n is the number of samples in the dataset; $y_i$ is the actual value and $\hat{y_i}$ is the predicted value of the i[th] sample.

RMSE is a metric of how far the prediction data points deviate from the actual data points. It is usually used in regression analysis.

RMSE is a non-negative metric. A lower RMSE is usually better than a higher RMSE when the metric is applied to the data with the same level of scale.

Compared to MSE, it is less likely to overestimate or underestimate the performance of the model. However, RMSE is still proportional to the size of the error. A large outlier will significantly impact the RMSE.

**2.8.4. R Squared ($R^2$).** R squared, also called coefficient of determination, is a metric measuring the degree to which the dependent variable can be predicted by the independent variable(s). The formulas are shown in Equation (4), (5) and (6):

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)} \tag{4}$$

$$\text{MSE(model)} = \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{n} \tag{5}$$

$$\text{MSE(baseline)} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n} \tag{6}$$

i means the i[th] sample; n is the number of samples in the dataset; $y_i$ is the actual value, $\bar{y}$ is the mean of all $y_i$, and $\hat{y_i}$ is the predicted value. MSE (model) is the mean squared error of the trained model, while MSE (baseline) is the mean squared error of a baseline model which only gives the mean value of all $y_i$.

$R^2$ can range from -∞ to 1 in the world of machine learning. The closer $R^2$ is to 1, the better the model performance. There is a common misconception that $R^2$ can only range from 0 to 1. As shown in the formula of $R^2$ above, we can see that $R^2$ can be less than 0. This situation happens when the MSE of our designed model is larger than the

MSE of the baseline model that only predicts the mean value, which is a horizontal line. Simply speaking, in the world of machine learning, when your model is doing worse than a horizontal line in fitting the data, $R^2$ can be negative.

# 3. METHODOLOGY

Aesthetic evaluation is a challenging task for machine learning because the perception of beauty or aesthetics can be subjective. Often users can only provide ratings or scores for aesthetics according to their aesthetic sense and intuition but cannot articulate the specific reasons. Experts and scholars have proposed different methods to quantify aesthetics. Traditional and mainstream methods include the development of aesthetic features (i.e., complexity, equilibrium and symmetry) based on aesthetic principles and the measurement of some low-level image features (i.e., color and texture). The advantage of this method is that the results can be directly applied by designers in practice once they are shown to be effective. The downside of this approach is that deep domain knowledge and insights from a lot of design work are necessary to arrive at effective and reliable results. Moreover, many of such methods are still in the theoretical stage, and it requires a lot of work and cost to translate them into practice. Therefore, the traditional methods can be harder to apply and less practical.

With the Renaissance and vigorous development of artificial intelligence and machine learning in recent years, some scholars advocate deep learning methods to extract features directly from images (Khani et al., 2016; Talebi & Milanfar, 2018; Dou et al., 2019). Although deep learning methods often cannot come up with effective design principles, it can be direct and effective in application.

It is hard to compare these two methods in terms of which is more superior. We will attempt to make some interesting comparisons from different angles and demonstrate the application of machine learning in this research.

## 3.1. ASSESSING AESTHETICS USING TRADITIONAL METHODS

Assessing aesthetics using traditional methods usually involves using design features associated with aesthetics. In this thesis, we employed the features proposed by previous research (Reinecke et al., 2013).

Reinecke et al. (2013) provided a traditional aesthetic measurement that uses design features to assess aesthetics. They extracted these features from screenshots of webpages. Each screenshot was taken at a 1024*768 resolution. Each pixel was identified by one of the W3C colors and the percentage of pixels of each W3C color in the image was recorded. The hue, saturation, and value were calculated based on the average of the pixels' corresponding values.


Figure 3.1. Quadtree Decomposition (Reinecke et al., 2014).

The quadtree decomposition technique was used to decompose the screenshot into subparts (also called leaves). Figure 3.1 shows a webpage being divided into subparts by

quadtree decomposition. A webpage is being divided into subparts. An area will be recursively divided until the entropy of the area drops to a required level. Based on these subparts, text, image and non-text subparts were identified and calculated. Symmetry, balance and equilibrium were calculated based on the layout arrangement of the subparts.

Space decomposition was also used to divide and classify the areas of text and image. It separates the webpage Figure 3.2 shows how the space decomposition divides a webpage. More details about the decomposition techniques can be found in Section 2.7.


Figure 3.2. Space-Based Decomposition (Reinecke et al., 2014).

## 3.2. ASSESSING AESTHETICS USING DEEP LEARNING MODELS

Assessing aesthetics using deep learning models is an approach that arises in recent years. Due to the boom of data and the increasing power of computers, deep learning techniques have become a viable approach in various fields. They are widely

used in various fields of society. For example, a convolutional neural network can help doctors detect breast cancer. A recurrent neural network can help people write patent abstracts. There are numerous opportunities and potential uses of deep learning techniques that are waiting for people to explore and evaluate.

In this study, we used several classes of convolutional neural networks to help us predict aesthetics based on the screenshots of websites. We first used some shallow convolutional neural networks with different layers, then we used several state-of-the-art convolutional neural networks, including NasNet, MobileNet and Inception-ResNet-V2. We hope to see promising results generated by these models. However, there are many difficulties and limitations about the approach as well, as will be discussed later.

### 3.3. RESEARCH METHODOLOGY

In this section, we experimented with predicting aesthetics using several different state-of-the-art machine learning models. We also conducted exploratory analysis on aesthetic features to validate the conclusions of previous research and to provide more insights. We will report the aesthetic prediction performance in the data analysis and results section. We chose Python as the programming language because it is powerful and suitable for rapid development. It is a great tool for data analysis, especially in machine learning and artificial intelligence. The Keras deep learning framework is adopted because it is easy to use and fast to develop, which allows users to assess various architectures quickly. The study was conducted using Google colab, which is a free Jupyter notebook environment provided by Google. We also used the free GPU of colab to speed up the training process, which is a Tesla T80 GPU.

All of the data analysis was carried out on the same dataset built by Reinecke and Gajos (2014). This dataset provided a data foundation for analyzing webpage aesthetics regardless of the approach used. The data set contains images with their corresponding aesthetic ratings, specific aesthetic features, as well as the demographic information of the participants.

### 3.4. DATASET

We used the public dataset collected by Reinecke et al. (2013). The dataset includes 398 screenshots of web pages and ratings from more than 40,000 participants. They conducted 10-minute online tests on their platform (LabintheWild.org) and advertised their study in online communities and college newsletters. This dataset can be found at http://labinthewild.org/data/index.php. Most of the participants are from the US (43%), followed by the United Kingdom (17%), Hungary (6%), Canada (5%), and Romania (3%). The rest were from other countries. There is demographic information in this dataset, but it is beyond the scope of this research to examine them. For more details, please refer to the original paper (Reinecke et al., 2014) and their website at LabintheWild.org.

Thanks to Reinecke and Gajos' (2014) contribution for making the first public dataset on webpage visual aesthetics available at LabintheWild.org. For each screenshot in the dataset, there are associated aesthetic ratings on a scale from 1 to 9, with 1 being the least visually appealing and 9 being the most visually appealing. We used the average aesthetic score of the participants for each image evaluated, unlike Dou et al.'s (2019) study of feeding the neural networks with these images and the aesthetic scores of each

individual user, which could potentially bias or overfit the results due to the duplication of data for each webpage. We averaged the aesthetic scores of the participants for each webpage due to the consideration that mathematical confusion can be caused by feeding the neural networks with the same image and a set of aesthetics scores that differ from one another. Future research can include the demographic information of the respondents to analyze differences in aesthetic preferences of users. Since the purpose of this study is to assess webpage aesthetics, taking the mean webpage aesthetic scores is most appropriate.

### 3.5. DATA COLLECTION PROCESS

The data collection process for the dataset was divided into two phases to determine the stability of the results. Before the two phases of the experiment, a practice phase was conducted in which participants rated a set of 5 webpages. In the first phase of the experiment, each participant was asked to rate 30 webpages that were randomly selected from 430 webpages. Of the 30 webpages, 22 are in English, 4 in foreign languages and 4 from Webby Award websites. The order of the 30 pages was randomly shuffled. Each participant was given 500ms to view a webpage to avoid excessive exposure to its contents. The participants were required to rate colorfulness, complexity, and visual appeal on a Likert scale from 1 to 9, where 1 represents “not at all colorful”, “not at all complex” and “not at all visually appealing”, and 9 represents the best scores for the three metrics. After the first phase, participants were encouraged to take a short break.

In the second phase, participants were asked to rate the same 30 webpages, but the order was randomly shuffled again. The purpose of doing this is to test the consistency of the ratings. According to Reinecke et al.'s (2013) experiment results, the ratings were consistent across the two phases.

### 3.6. DEALING WITH MISSING DATA

We started by dealing with missing values. However, most of the missing values exist in demographic features. Since demographic features are out of the scope of this research, we deleted all the demographic features.

After deleting the demographic features, we found that there are still missing values in 4 features: 'colorVerticalBalance', 'colorHorizontalBalance', 'intensityVerticalBalance', 'intensityHorizontalBalance'. From previous research, we know that color and intensity are important features. Thus, we need to further address the missing values of the features. We found that missing values in 'colorHorizontalBalance' and 'colorVerticalBalance' are all in the same rows. One possible reason is that the original authors did not record the values for some specific webpages. However, if we delete these rows, we would lose around 30% of the useful data of other features. Further, we only have a small dataset and hence, deletion would significantly decrease the model performance. Thus, we filled in the missing values with the mean value. Future research should be noted for this data processing step.

This method of filling missing data with mean values is beneficial for linear and neural network models. However, it makes tree models such as decision trees and the random forest harder to understand the patterns of the data.

### 3.7. DEALING WITH DUPLICATED DATA

After dealing with missing data, we found that the dataset has duplications because webpage features were duplicated for all respondents of the same webpage. In other words, although there were numerous rows of data for each webpage (i.e., due to multiple respondents rating it), they are only 398 webpages in the dataset. This is to say, there were duplicated data on the features of the webpages such as colorVerticalBalance and intensityHorizontalBalance. Since having such duplications in the training set would confuse the machine learning models and create biases, we decided to use the mean rating for each webpage as the only output for the webpage. Hence, we collapsed all the rows for the same webpage and averaged the aesthetic ratings for each webpage to obtain its mean aesthetic rating. Hence, the final dataset has 398 records, one for each webpage.

### 3.8. DATA SPLIT

We used 70% of the dataset as our training data to train our models and 30% as our testing data to test or assess their generalizability to external data. This split percentage is common in the field.

Data split in machine learning is done to randomly divide the data into different datasets (e.g., 80% of training data and 20% of test data). Part of the data is used to train the model, and part of the data is used to verify the correctness of the trained model. In the modern world of machine learning, we usually see two datasets as the outcomes of the data split: training dataset and testing dataset. The training data often consist of the input data and the corresponding output data, which is also known as the target or label. The model is fed with the input data and produces a predicted result, which is then

compared with the corresponding output data. The trained model will then be tested using the testing dataset to assess its performance in practice. Sometimes, a subset of the training data is used for validation purposes. However, since our dataset is small, this approach would not be appropriate. We chose to use the testing dataset for both validation and testing because we want to make full use of the dataset to improve the performance of the models.

### 3.9. FEATURE SCALING

Feature scaling is a data preprocessing technique to normalize the range of data. We have found that our dataset has a potential problem of having features of different ranges. Features with a larger range could have a huge impact on machine learning models compared to features with a relatively smaller range. Normalizing the features would make the range for all the features the same, i.e., on a scale of [0, 1], which also would speed up the computation with smaller numbers.

For example, ‘colorHorizontalSymmetry’ is a feature that ranged from 0 to 1. ‘numOfLeaves’ is a variable that depends on the specific webpages and hence, could have a maximum of 249 for this dataset. A change in numOfLeaves is more likely to change the model than a change in colorHorizontalSymmetry. This would cause colorHorizontalSymmetry to have a higher weight.

However, tree-models are usually not influenced by the feature ranges. Tree-models are based on the cut-off points instead of the ranges of data. Machine learning models rely on learning the distance between data points. These models are more likely to be influenced by the ranges of features.

Since features are of different numerical ranges and we used some non-tree models, we conducted feature scaling on our dataset by scaling all features to fall within the range of [0, 1].

### 3.10. STATISTICS OF PRE-PROCESSED DATA

Next, we describe the data to uncover potential problems and to better understand the data. First, we made a statistical description of the data to understand the general distribution. We then made histograms of the data to better understand the characteristics of the distributions.

Figure 3.3 and Figure 3.4 show the statistical description of aesthetic features such as W3C colors, ‘textArea’, etc. The table in Figure 3.3 shows the W3C color features. Figure 3.4 shows the rest of the aesthetic features used for predictive modeling. There are 398 rows of data left after being aggregated. The aesthetic ratings were averaged across each webpage. The mean value, standard deviation, min and max, quartiles (25%, 50% & 75%) of the features are shown in Figure 3.4.

Figure 3.3 shows the W3C color features. The value of a W3C color feature refers the percentage of pixels that are close to this color. The values of the W3C colors were scaled to the range of [0, 1], with 0 representing 0% and 1 representing 100%. From Figure 3.3 of W3C color features, we can see that silver and white have higher mean values than the other colors. Many colors such as black and white can have max values of over 90%, which means the color may be used widely or as the theme color of a webpage in the dataset. As a rule of thumb, we often see websites using these colors as their theme colors.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **black** | 398.0 | 0.144269 | 0.185828 | 0.000000 | 0.034250 | 0.078258 | 0.169659 | 0.958622 |
| **silver** | 398.0 | 0.710223 | 0.228808 | 0.012489 | 0.620043 | 0.766852 | 0.880754 | 0.993776 |
| **gray** | 398.0 | 0.193459 | 0.149899 | 0.001628 | 0.085487 | 0.168248 | 0.263483 | 0.956532 |
| **white** | 398.0 | 0.633602 | 0.242772 | 0.005427 | 0.505738 | 0.675616 | 0.826572 | 0.989586 |
| **maroon** | 398.0 | 0.151586 | 0.189676 | 0.000000 | 0.034599 | 0.085255 | 0.182438 | 0.970598 |
| **red** | 398.0 | 0.028714 | 0.057839 | 0.000000 | 0.001787 | 0.009868 | 0.030802 | 0.715355 |
| **purple** | 398.0 | 0.123044 | 0.124060 | 0.001024 | 0.040168 | 0.091453 | 0.166450 | 0.925477 |
| **fuchsia** | 398.0 | 0.003193 | 0.028589 | 0.000000 | 0.000000 | 0.000000 | 0.000072 | 0.521226 |
| **green** | 398.0 | 0.153009 | 0.184342 | 0.000020 | 0.037483 | 0.089272 | 0.195093 | 0.970805 |
| **lime** | 398.0 | 0.005665 | 0.030460 | 0.000000 | 0.000000 | 0.000009 | 0.000981 | 0.449007 |
| **olive** | 398.0 | 0.133130 | 0.138234 | 0.001212 | 0.041407 | 0.093382 | 0.177987 | 0.985568 |
| **yellow** | 398.0 | 0.021629 | 0.054596 | 0.000000 | 0.001389 | 0.006187 | 0.019767 | 0.647784 |
| **navy** | 398.0 | 0.160755 | 0.178725 | 0.000020 | 0.046987 | 0.100102 | 0.203150 | 0.949079 |
| **blue** | 398.0 | 0.019564 | 0.045978 | 0.000000 | 0.000105 | 0.002854 | 0.015822 | 0.395738 |
| **teal** | 398.0 | 0.148308 | 0.139692 | 0.001024 | 0.050906 | 0.110153 | 0.188051 | 0.935851 |
| **aqua** | 398.0 | 0.018839 | 0.059707 | 0.000000 | 0.000025 | 0.001457 | 0.009794 | 0.721638 |

Figure 3.3. Statistics of W3C Color Features.

Figure 3.4 displays other aesthetic features used for modeling. We can see that many features have very different ranges. Features such as ‘textArea’ and ‘nonTextArea’ have very large ranges. Some features such as symmetry, balance, equilibrium have a smaller range of [0, 1] due to the way they were calculated. For some predictive models, the differences brought by ranges can significantly affect the performance of the prediction. In general, these models could overestimate the significance of these features. Due to the reason that different ranges of features could cause an impact on predictive

modeling, we conducted feature scaling. The details of feature scaling can be found in the previous section. The statistics of the features shown in this section are not scaled, in order to show their original characteristics to readers.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| hue | 398.0 | 33.328206 | 23.138742 | 0.070104 | 16.560970 | 29.914656 | 45.861822 | 130.205659 |
| saturation | 398.0 | 49.691804 | 38.756000 | 0.672611 | 19.752175 | 42.057675 | 68.650980 | 214.977095 |
| value | 398.0 | 202.033786 | 43.735723 | 10.822652 | 188.162361 | 212.212596 | 230.598730 | 253.951909 |
| textArea | 398.0 | 108858.701005 | 56253.979296 | 3368.000000 | 68469.250000 | 104130.000000 | 140043.000000 | 353946.000000 |
| nonTextArea | 398.0 | 202314.246231 | 164829.660882 | 0.000000 | 78996.250000 | 159225.000000 | 291344.500000 | 707984.000000 |
| numOfLeaves | 398.0 | 63.698492 | 37.229082 | 5.000000 | 37.000000 | 57.000000 | 83.000000 | 249.000000 |
| percentageOfLeafArea | 398.0 | 40.079889 | 18.675232 | 3.473409 | 25.983365 | 38.319906 | 51.626046 | 98.059209 |
| numOfTextGroup | 398.0 | 17.695980 | 10.200041 | 1.000000 | 11.000000 | 16.000000 | 24.000000 | 76.000000 |
| numOfImageArea | 398.0 | 7.685930 | 5.166191 | 0.000000 | 4.000000 | 7.000000 | 11.000000 | 29.000000 |
| colorfulness1 | 398.0 | 13.147401 | 10.866959 | 0.300587 | 5.791128 | 10.787506 | 16.888253 | 73.335661 |
| colorfulness2 | 398.0 | 55.484403 | 28.154963 | 3.914874 | 35.502134 | 53.310139 | 69.794976 | 165.971061 |
| complexitymodel | 398.0 | 4.821873 | 1.219119 | 1.377732 | 4.079276 | 4.871956 | 5.607746 | 9.847887 |
| colorHorizontalSymmetry | 398.0 | 0.841244 | 0.129338 | 0.313277 | 0.753815 | 0.848998 | 0.964030 | 1.000000 |
| colorVerticalSymmetry | 398.0 | 0.788086 | 0.172137 | 0.270739 | 0.672137 | 0.794622 | 0.960289 | 1.000000 |
| colorHorizontalBalance | 398.0 | 0.440580 | 0.279062 | 0.001923 | 0.222902 | 0.440701 | 0.579552 | 1.000000 |
| colorVerticalBalance | 398.0 | 0.565392 | 0.285174 | 0.000000 | 0.356355 | 0.564381 | 0.789816 | 1.000000 |
| intensityHorizontalSymmetry | 398.0 | 0.867160 | 0.106043 | 0.301128 | 0.812688 | 0.887141 | 0.947315 | 1.000000 |
| intensityVerticalSymmetry | 398.0 | 0.804801 | 0.150572 | 0.063725 | 0.702182 | 0.824876 | 0.923246 | 1.000000 |
| intensityHorizontalBalance | 398.0 | 0.493767 | 0.323447 | 0.000954 | 0.226032 | 0.477812 | 0.744894 | 1.000000 |
| intensityVerticalBalance | 398.0 | 0.642623 | 0.313979 | 0.000954 | 0.384965 | 0.659180 | 0.995913 | 1.000000 |
| numOfQuadTreeLeaves_color | 398.0 | 629.437186 | 620.045046 | 4.000000 | 100.750000 | 482.000000 | 945.750000 | 4096.000000 |
| numOfQuadTreeLeaves_intensity | 398.0 | 552.954774 | 670.248552 | 4.000000 | 143.500000 | 360.000000 | 697.500000 | 4036.000000 |
| colorEquilibrium | 398.0 | 0.632367 | 0.336900 | 0.000000 | 0.553697 | 0.769873 | 0.874023 | 0.996643 |
| intensityEquilibrium | 398.0 | 0.677976 | 0.234778 | 0.000000 | 0.586328 | 0.738118 | 0.840007 | 0.998698 |

Figure 3.4. Statistics of Other Aesthetic Features.

Figure 3.5 and Figure 3.6 show the histograms of aesthetic features. Figure 3.5 shows the W3C color features and Figure 3.6 shows the other aesthetic features. A brief explanation of W3C color features can be found in Section 2.7.

From Figure 3.5, we can see that most of the colors such as aqua, gray and red are skewed to the right. The right-skewed shape means the mean value is greater than the median value. One possible explanation is that these colors generally are not used in large areas of webpages. Some of the colors such as fuchsia and lime are so rare in webpages

that their percentages are mostly zero for webpages. The distributions of silver and white are skewed to the left, which means these two colors are more common in webpages. However, these findings could be biased by the size of the dataset. A larger dataset would be more credible for validating these observations.

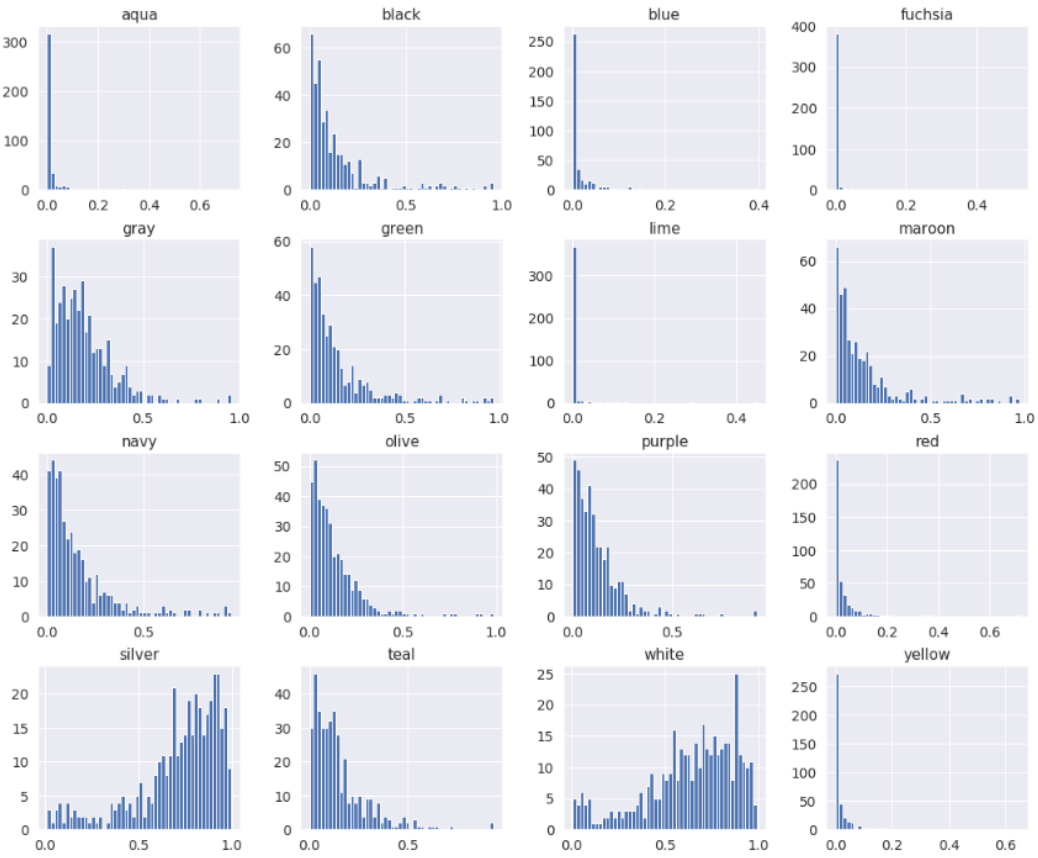



Figure 3.5. Histograms of W3C Color Features.

Figure 3.6 shows the histograms of other aesthetic features used for predictive modeling. Details of these metrics can be found in Section 2.7.

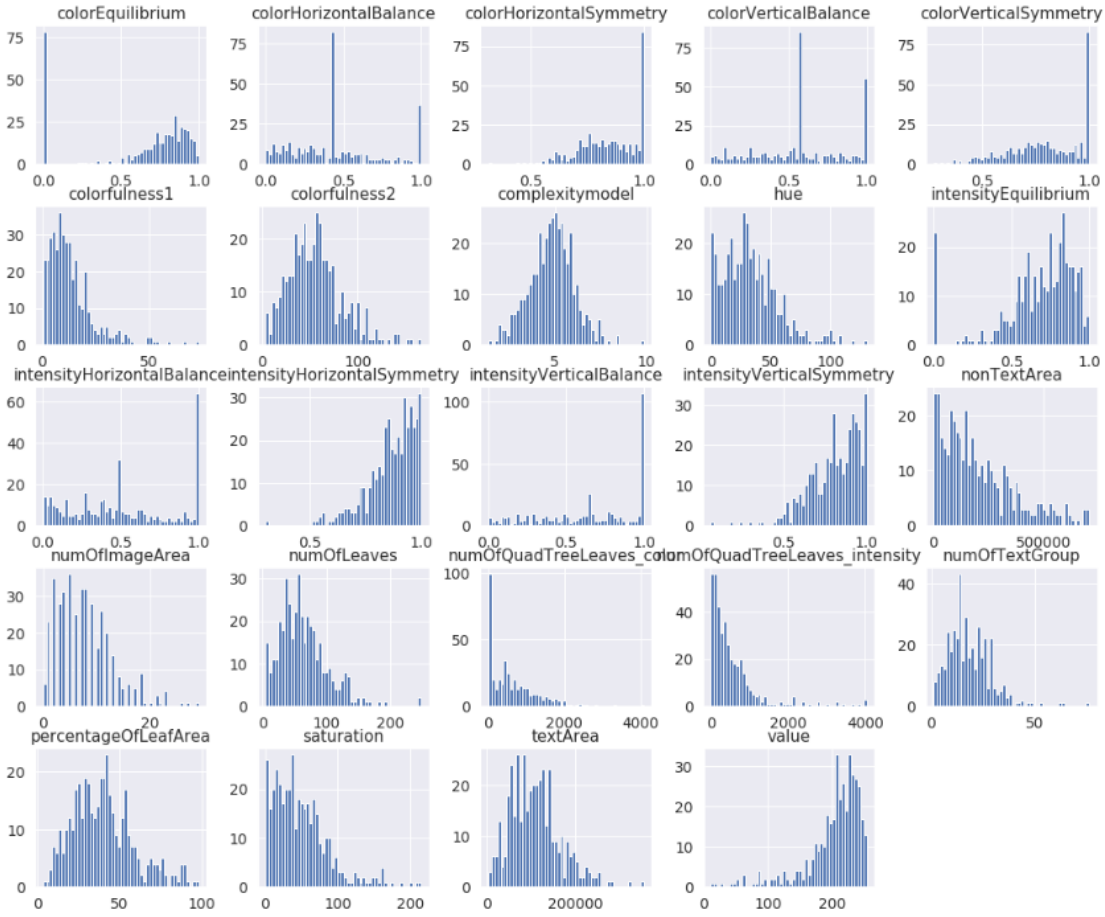



Figure 3.6. Histograms of Other Aesthetic Features.

From Figure 3.6, we can see that most of the features are either left-skewed or right-skewed, with the exception of complexity since the feature ‘complexitymodel’ approximates a normal distribution.

Previous research (Reinecke et al., 2013) has reported that complexity and colorfulness are two important features of visual appeal. From Figure 3.6, we found that complexity has a bell-shaped distribution and is approximately normally distributed, which means the webpages averaged around the medium complexity value (i.e., not too complex or too simple). For colorfulness, the distribution is skewed to the right, which

means the mean value is to the right of the median value. One possible explanation is that many of the webpages are less or not very colorful. To further validate these observations, a larger dataset and a more systematic study are required.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **mean_response** | 398.0 | 4.334137 | 1.027231 | 1.486928 | 3.684059 | 4.390343 | 5.073603 | 7.033333 |

Figure 3.7. Statistics of Average Aesthetic Rating.


Figure 3.8. Density Plot of Average Aesthetic Rating.

Figure 3.7 and Figure 3.8 show the statistics and the density plot of the average aesthetic ratings respectively. According to the original paper (Reinecke et al., 2013), the webpages were rated on a scale from 1 to 9. The statistics in Figure 3.7 show that the max

average rating can be as high as 7.03 and the lowest average rating can be as low as 1.49. The standard deviation (STD) is 1.02, which is relatively small compared to the scale of [1, 9]. The distribution of aesthetic ratings is normally distributed as shown in Figure 3.8. The median (50% quartile) aesthetic rating is 4.39, which is close to the average rating of 4.33.

# 4. DATA ANALYSIS AND RESULTS

## 4.1. MODEL PERFORMANCE (AESTHETIC FEATURE METHOD).

We trained multiple machine learning models to predict the aesthetic scores based on aesthetic features. The features used can be found in Figure 3.3 and Figure 3.4. To assess the performance of these predictive models, we evaluated the model performance using mean absolute error (MAE), R-squared (R2), mean squared error (MSE) and root mean squared error (RMSE), as shown in Figure 4.1. More explanations of these performance metrics can be found in Section 2.8 on the performance metrics dictionary.

| | Model Type | MAE | R2 | MSE | RMSE |
|---|---|---|---|---|---|
| 0 | Stat Linear Regression | 0.746463 | -0.016876 | 1.036214 | 1.017946 |
| 1 | Decision Tree | 0.878440 | -0.337621 | 1.363059 | 1.167501 |
| 2 | Random Forest | 0.601449 | 0.394111 | 0.617412 | 0.785755 |
| 3 | Gradient Boosting Regression | 0.629107 | 0.360738 | 0.651419 | 0.807105 |
| 4 | Neural Network(20,15,10) | 0.825935 | -0.029355 | 1.048931 | 1.024173 |
| 5 | Neural Network(20,15,10,5) | 0.785668 | -0.039947 | 1.059724 | 1.029429 |
| 6 | Neural Network(25,20,15,10,5) | 0.806575 | -0.080581 | 1.101131 | 1.049348 |

Figure 4.1. Machine Learning Model Performance using Aesthetic Features.

To predict aesthetic scores, we trained a multiple linear regression model, a decision tree model, a random forest model, a gradient boosting regression model, a

multilayer perceptron neural network with 3 layers (20, 15, 10 neurons), a multilayer perceptron neural network with 4 layers (20, 15, 10, 5 neurons), and a multilayer perceptron neural network with 5 layers (25, 20, 15, 10, 5 neurons). As shown in Figure 4.1, we found that the random forest model has the best overall performance among these models. The second best is the gradient boosting regression model with very similar performance.

**4.1.1. Feature Selection.** Feature selection refers to using machine learning techniques to select the features that contribute most to the predictive model or manually select the features that you are interested in.

Rich domain knowledge is extremely important for selecting important features. Machine learning practitioners like to use models and various analysis techniques to infer important feature variables. In the business world, a data analytics professional with a keen business sense can determine, based on his or her experience and intuition, what features are important for business target variables.

Good feature selection can significantly improve the performance of the model and build a robust model. By selecting aesthetic features from the feature pool, it helps us better understand the characteristics and underlying relationship of the data, which plays an important role in further improving the model and algorithm. In terms of practical application, fewer features can reduce the amount of data needed for prediction, so that the model can make relatively accurate predictions with only a few important features. In this section, we selected the features from two perspectives: research interest and importance.

**4.1.2. Feature Selection Based on Interest.** From the perspective of research interest, we would like to use colorfulness and complexity to predict the aesthetic rating. In previous research (Reinecke et al., 2013), colorfulness and complexity were shown to be important predictors for the visual appeal rating. They used quadratic terms of colorfulness and complexity to make up for the 'U-shape' nonlinear relationships with visual appeals. The experiment results of their model showed that their model explained 48% variance in aesthetic preferences (R-squared = 0.48).

Thus, we replicated the model using the random forest model because it has good explanatory power for non-linear relationships.

Table 4.1. Performance Scores of the Random Forest Model Using Only Complexity and Colorfulness.

| **Evaluation Result** | |
|---|---|
| Mean Absolute Error | 0.796155 |
| R-squared | -0.112257 |
| Mean Squared Error | 1.133410 |
| Root Mean Squared Error | 1.064617 |

Table 4.1 shows the performance scores of the random forest model using only complexity and colorfulness as features without adding the quadratic terms for them. The mean absolute error, mean squared error and root mean squared error are relatively larger than the errors of the previous random forest model before the feature selection. Further, the r-squared score is negative, which means the model does not have enough power in

explaining the variance in the aesthetic rating. In general, the performance is not favorable compared to our previous models. Thus, we decided to do some feature engineering by adding the quadratic terms of these two features to the model.

Table 4.2 shows the performance of the random forest model after we added the quadratic terms of complexity and colorfulness to the model. We can see that most of the performance metric scores are much improved. Although the performance has been improved, the scores of this model is still not as good as the one before feature selection. We expect the random forest model with non-linear relationship explanation ability to predict well by using the two features directly, but the results show that the model is more accurate when they are combined with quadratic terms. Future research should be noted about the improvement brought by adding higher order terms of the features.

Table 4.2. Performance of Random Forest Using Features: Complexity, Colorfulness, Quadratic Term of Complexity and Quadratic Term of Colorfulness.

| **Evaluation Result** | |
|---|---|
| Mean Absolute Error | 0.728622 |
| R-squared | 0.096826 |
| Mean Squared Error | 0.920350 |
| Root Mean Squared Error | 0.959349 |

Although adding quadratic performance brings improvement to the model, there is still a big gap in performance when compared to a full model using all aesthetic

features. Hence, the results show that complexity and colorfulness cannot fully explain the aesthetic rating.

**4.1.3. Feature Selection Based on Importance (Using Random Forest Model).** The idea of selecting features based on importance is to use the machine learning algorithms to select the best features to build a prediction model. The model will be used to test the importance of each feature to the response variable. We will select the top-most important features to build a simplified and more robust model.

If the relationship between a feature and a response variable is non-linear, a tree-based approach (decision tree, random forest, etc.) can be used. Tree-based methods are easier to use because they model non-linear relationships better and require less debugging. Based on previous research (Reinecke et al., 2013) and the performance of the linear regression model, many features may have a non-linear relationship with the aesthetic rating. Thus, it is very appropriate to use a tree-based model to select important features. In this thesis, I will use the random forest model to select the best features since it is the best model we found for this problem. We trained the model with all features and let the model compute the importance of these features. The importance is visualized in Figure 4.2.

From Figure 4.2, we see that 4 features stand out from the feature pool. They are ‘nonTextArea’, ‘textArea’, ‘blue’ and ‘complexitymodel’. Among these features, ‘nonTextArea’ and ‘textArea’ refer to the numbers of leaves that are recognized as non-text or text by the algorithm. These two features have been shown to have a great impact on complexity (Reinecke et al., 2013). Interestingly, these two features also suggest an important aesthetic concept—white space. White space does not necessarily mean white

color, but refers to the blank space between web design elements. White space can give web design a decent, breathable sense of beauty and expression.

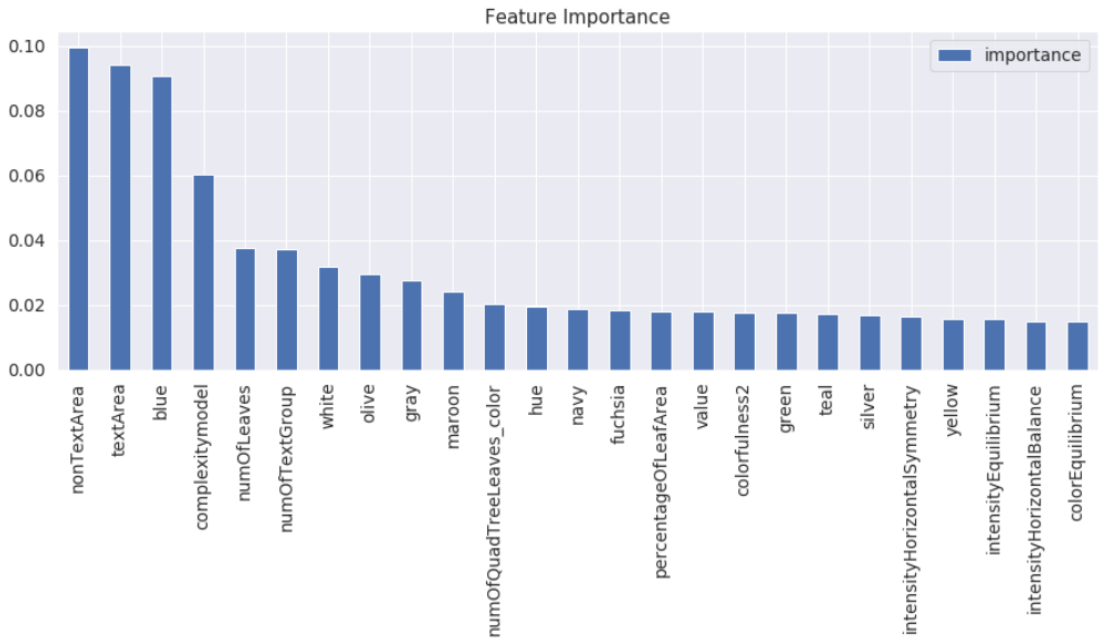



Figure 4.2. Important Features Selected by the Random Forest Model.

Complexity (complexitymodel) is a feature that has been proven by researchers to be important to visual appeal (Reinecke et al., 2013). In the latter section of regression analysis of complexity, the visualization suggests that users prefer a moderate level of complexity. Thus, it is reasonable to use complexity as one of the most important features of aesthetics.

The feature ‘blue’ estimates the percentage of pixels classified as blue in color by an algorithm using the W3C color system. Blue is the color of sky and water. Seeing blue usually means good weather and clean water. It gives a sense of spaciousness and calmness. It is one of the most popular colors around the world. Figure 4.3 shows the

results of a worldwide survey about favorite colors (William 2015). It is found that blue is the favorite color in 10 countries across 4 continents. Some countries also favor colors such as red, green and purple. Blue is significantly more popular than these colors across different countries and cultures. Blue is also an important and safe choice of color in UI design. Many apps such as Facebook, Twitter, and Safari use blue in logos and the main color of their website design. Thus, it does make sense that the color feature 'blue' can be important to a visually appealing website.

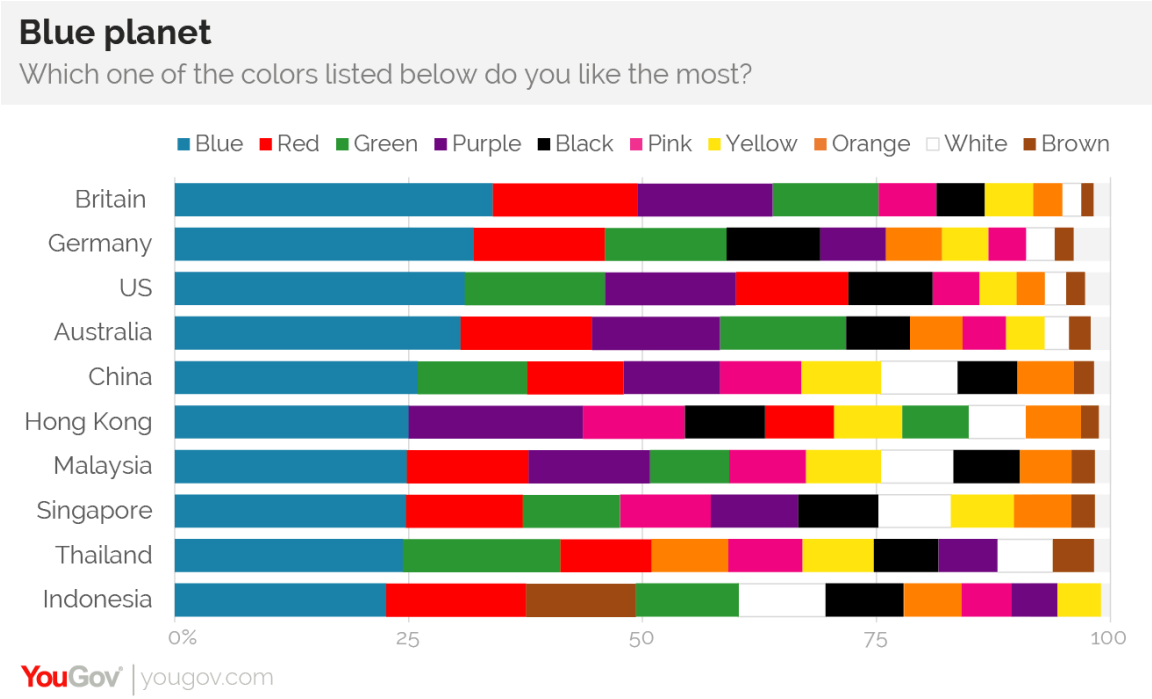



Figure 4.3. Most Popular Colors around the World (William, 2015).

'textArea' and 'nonTextArea' are two interesting features we found that are important to webpage aesthetics. As a rule of thumb in web design area, websites with fewer texts are usually perceived to be more aesthetically appealing. One of the guesses

is that more text will increase the information entropy of the web page and increase the cognitive burden of readers. To further validate our thoughts, we created scatter plots to better understand the relationships.

Figure 4.4 shows the linear regression plots of ‘nonTextArea’, ‘textArea’, ‘blue’ and ‘complexitymodel’. The dependent variable is the aesthetic rating. We can see that the feature ‘nonTextArea’ has a positive relationship with the aesthetic rating. The ‘textArea’, ‘blue’ and ‘complexitymodel’ have negative relationships with the aesthetic rating. However, non-linear relationships could have existed between these features. These linear regression plots cannot directly detect non-linear relationships.

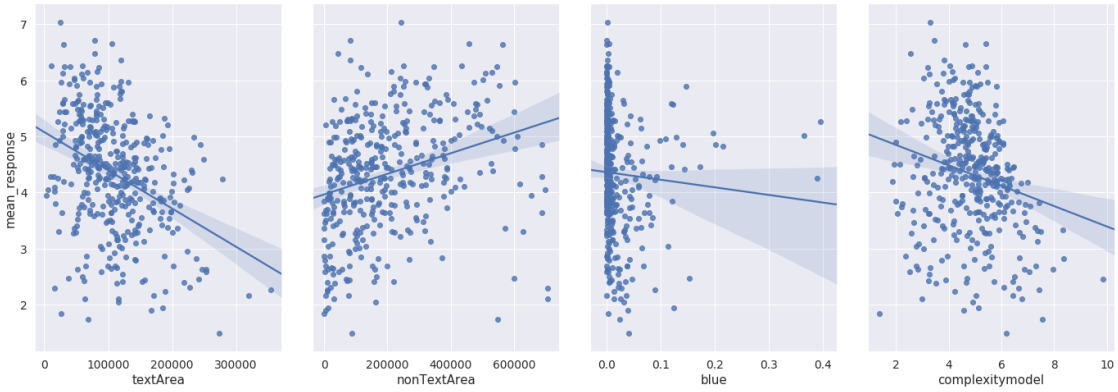



Figure 4.4. Linear Regression Plots of ‘nonTextArea’, ‘textArea’, ‘blue’ and ‘complexitymodel’.

To discover whether there are non-linear relationships between features and the aesthetic rating, Figure 4.5 was made. Figure 4.5 shows the regression plots with fitted regression lines, which can help detect non-linearity. From Figure 4.5, we can see that ‘textArea’ and the aesthetic rating has a linear relationship. It can be inferred that the aesthetic rating decreases as ‘textArea’ increases. The ‘nonTextArea’, ‘blue’ and

‘complexitymodel’ displayed different levels of non-linearity with the aesthetic rating. The aesthetic rating first increases and then stops increasing after ‘nonTextArea’ increases to an optimum point. The aesthetic rating increases rapidly and then begins to increase slowly when ‘blue’ increases to the value of around 0.2 (20%). The aesthetic rating first increases and then decreases as ‘complexitymodel’ increases.

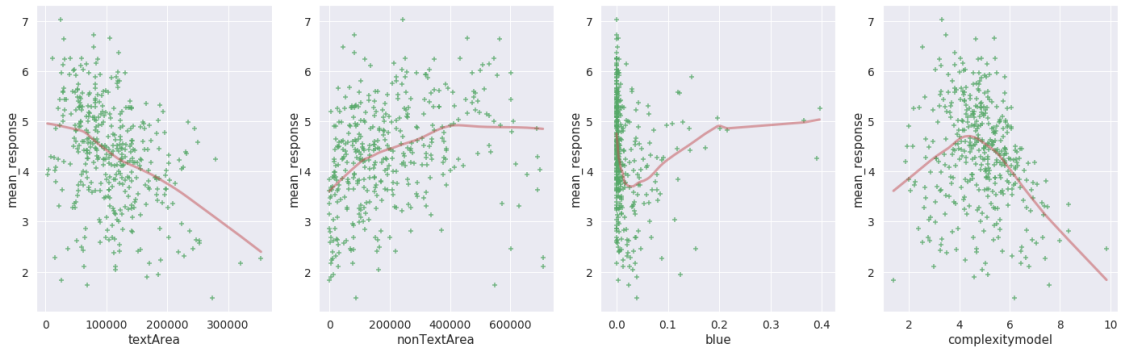



Figure 4.5. Regression Plots of ‘nonTextArea’, ‘textArea’, ‘blue’ and ‘complexitymodel’ with Fitted Regression Lines.

### 4.1.4. Model Performance on Selected Features (Based on Importance).

The table in Figure 4.6 shows the results of model performance after we selected features based on importance using the random forest model. The selected features are: ‘nonTextArea’, ‘textArea’, ‘blue’ and ‘complexitymodel’. As we expected, the performance of the random forest model did not change much. The performance of gradient boosting regression has been reduced after feature selection. Also, some models that were not performing very well have been greatly improved by feature selection. For example, the neural networks have been greatly improved. For example, the R-squared score of a neural network of 3 layers (20, 15, 10) has significantly increased from -0.029

to 0.305. The other models such as decision tree, neural network (20,15,10,5) and neural network (25,20,15,10,5) have also been improved. And we found that the neural network with only 3 layers perform better than the neural networks with 4 and 5 layers. To predict aesthetics with fewer features, neural networks with a simple structure are more appropriate. Hence, using fewer features can help models become more robust and achieve better performance.

| | Model Type | MAE | R2 | MSE | RMSE |
|---|---|---|---|---|---|
| 0 | Stat Linear Regression | 1.058453 | -0.761926 | 1.795433 | 1.339938 |
| 1 | Decision Tree | 0.813702 | 0.006793 | 1.012096 | 1.006030 |
| 2 | Random Forest | 0.641842 | 0.355002 | 0.657264 | 0.810718 |
| 3 | Gradient Boosting Regression | 0.675554 | 0.296802 | 0.716571 | 0.846505 |
| 4 | Neural Network(20,15,10) | 0.650074 | 0.305281 | 0.707931 | 0.841386 |
| 5 | Neural Network(20,15,10,5) | 0.659857 | 0.290684 | 0.722806 | 0.850180 |
| 6 | Neural Network(25,20,15,10,5) | 0.708120 | 0.186053 | 0.829426 | 0.910728 |

Figure 4.6. Performance of the Models Using Features Selected by Random Forest Model.

In theory, more features would help machine learning become more accurate and powerful. But in practice, we need to consider more factors, such as effective data volume, data noise, overfitting problems and so on. Sometimes, a few key features can make a good prediction. But it doesn't mean that most of the features that researchers have crafted are useless. The more aesthetic features we crafted, the more likely we are able to reveal the secret of aesthetics and make better predictions. Research should be

done to compare more aesthetic features using the same standard and setting and with a larger dataset. More aesthetic features should be crafted to help people explore and understand the mechanism of aesthetics.

## 4.2. MODEL PERFORMANCE (DEEP LEARNING MODELS)

We formulate the aesthetic rating prediction as a regression problem. Webpage screenshots were used as inputs for the convolutional models and the averaged aesthetic ratings were used as the target variable for the model. We made 4 shallow convolutional neural networks by adding up the convolutional layers from 2 to 5.

| | Model Type | MAE | R2 | MSE | RMSE |
|---|---|---|---|---|---|
| 0 | Shallow CNN with 2 Conv2D Layers | 0.754711 | -0.038906 | 0.937045 | 0.968011 |
| 1 | Shallow CNN with 3 Conv2D Layers | 0.840768 | -0.180009 | 1.064314 | 1.031656 |
| 2 | Shallow CNN with 4 Conv2D Layers | 0.747266 | -0.005201 | 0.906645 | 0.952179 |
| 3 | Shallow CNN with 5 Conv2D Layers | 0.742178 | -0.004758 | 0.906245 | 0.951969 |
| 4 | NasNet (without Finetuning) | 0.957529 | -0.673707 | 1.509606 | 1.228660 |
| 5 | NasNet (Finetuned on Fully Connected Layers) | 0.794741 | -0.164904 | 1.050690 | 1.025032 |
| 6 | NasNet (Finetuned on Conv and Fully Connected Layers) | 0.760792 | -0.048134 | 0.945369 | 0.972301 |
| 7 | MobileNet (Finetuned on Fully Connected Layers) | 0.885259 | -0.304059 | 1.176201 | 1.084528 |
| 8 | MobileNet (Finetuned on Conv and Fully Connected Layers) | 0.869658 | -0.190781 | 1.074030 | 1.036354 |
| 9 | Inception-ResNet-v2 (Finetuned on Fully Connected Layers) | 0.793607 | -0.117993 | 1.008378 | 1.004180 |
| 10 | Inception-ResNet-v2 (Finetuned on Conv and Fully Connected Layers) | 0.744601 | 0.009567 | 0.893324 | 0.945158 |

Figure 4.7. Performance of Deep Learning Models.

We later used the transfer learning technique to make up for the data insufficiency. Several pre-trained deep learning models including NasNet, MobileNet and Inception-ResNet were used. The performance of the deep learning models is shown in Figure 4.7. From Figure 4.7, we see that most of the deep learning models have very

similar performance. On the one hand, they all have pretty good scores on mean absolute error, mean squared error and root mean squared error. On the other hand, they also have unfavorable R-squared scores that are either negative or close to zero. More details about the setting of these deep learning models and how they were trained can be found in the following sections.

**4.2.1. Convolutional Neural Network with 2 Conv2D Layers.** Figure 4.8 show the summary information of the architecture of the CNN with two convolutional layers. The two Con2D layers were used to abstract features from the screenshots. The relu function was used as the activation function for the convolutional layers. After convolutional layers, we flattened the abstracted data and attached two fully connected layers. The last layer used 1 single neuron and the linear function as the activation function to produce a numerical aesthetic rating prediction.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 214, 214, 20)      7280
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 212, 212, 20)      3620
_________________________________________________________________
flatten_1 (Flatten)          (None, 898880)            0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               115056768
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 129
=================================================================
Total params: 115,067,797
Trainable params: 115,067,797
Non-trainable params: 0
```

Figure 4.8. Summary Information of the Convolutional Neural Network with 2 Conv2D Layers.

The Stochastic Gradient Descent method was used as the optimizer since it is appropriate for tasks with enough computation resources. We have a small dataset and a shallow CNN, which only requires a very small amount of computation resources. The mean absolute error is used as the loss function because we model predicting aesthetics as a regression problem and we want our model to directly predict aesthetics. The mean squared error is used to assess the performance of the model. The configuration details can be found in Table 4.3.

Table 4.3. Configuration of Compiler for the CNN Model with Two Conv2D Layers.

| **Compiler Configuration** | |
|---|---|
| Optimizer | stochastic gradient descent |
| Loss function | mean absolute error |
| Metrics | mean squared error |
| **Optimizer Configuration** | |
| Learning Rate | 1.00E-06 |
| Decay | 1.00E-06 |
| Momentum | 0.9 |
| Nesterov | TRUE |

After compiling our model, we started training the model. The training process is shown in Figure 4.9.

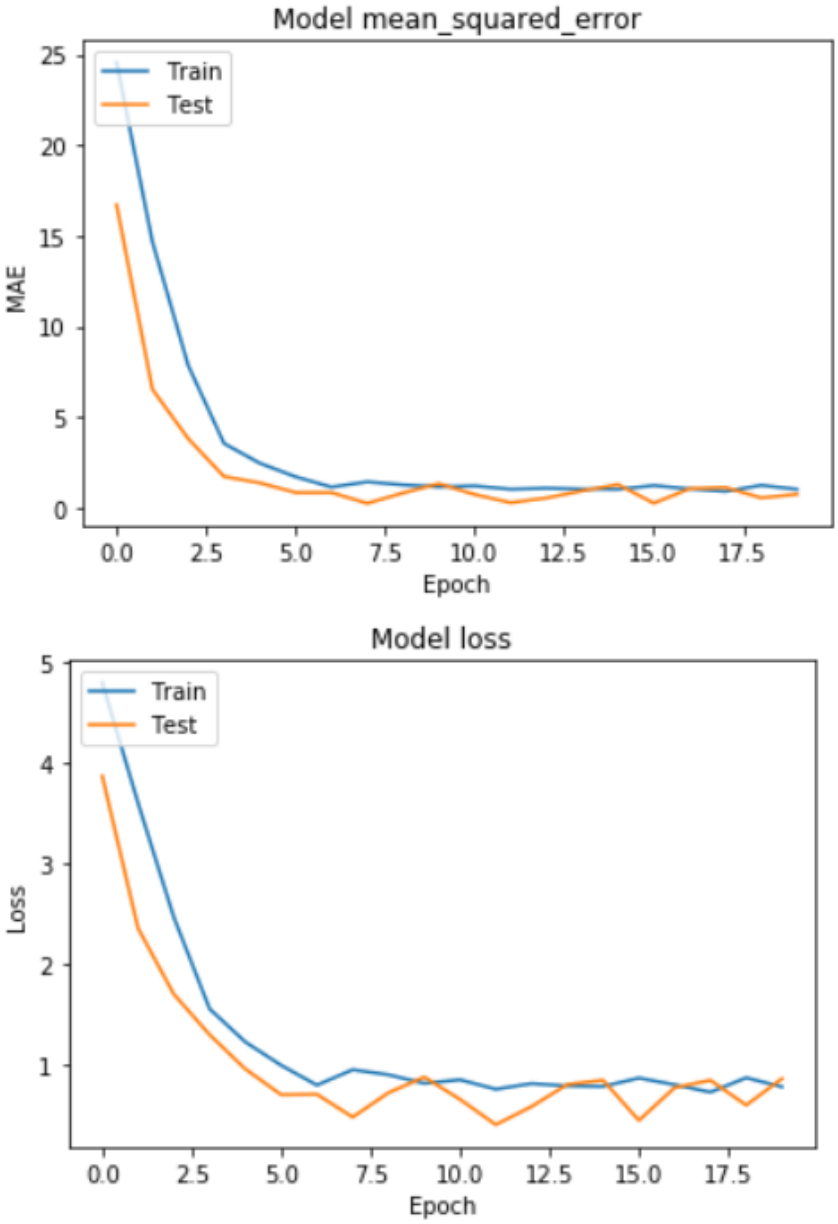



Figure 4.9. Learning Curves of a CNN Model with 2 Conv2D Layers. Mean Squared Error Curve (Upper). Model Loss (Mean Absolute Error) Curve (Lower).

As we can see in Figure 4.9, with the increase in epoch (iterations of the training process), the model loss (mean absolute error) and mean squared error kept decreasing. The decreasing loss and errors are usually a good thing, which means the model is

becoming better and better at the task of predicting webpage aesthetics. After around 6 epochs, the error almost stopped decreasing. The loss (mean absolute error) decreased to a value of around 0.8. There is no evident separation between the training line and the testing line, which means the model performs equally well on both the training dataset and the testing dataset.

After training our model, we evaluated the model. The results of the evaluation are shown in Table 4.4. The model has a good mean absolute error, a moderate mean squared error and a root mean squared error that is not as good. It has a negative R-squared score of -0.0389, which is close to zero. The results are not as good as the performance of models using the full set of aesthetic features. We suspect that our model may not be deep enough to catch the pattern of aesthetics. Thus, we decided to increase the depth of the model by adding more convolutional layers.

Table 4.4. Evaluation Result of the CNN Model with Two Conv2D Layers.

| **Evaluation Result** | |
|---|---|
| Mean Absolute Error | 0.754711 |
| R-squared | -0.038906 |
| Mean Squared Error | 0.937045 |
| Root Mean Squared Error | 0.968011 |

**4.2.2. Convolutional Neural Network with 3 Conv2D Layers.** Figure 4.10 shows the summary information on the architecture and parameters of the CNN with 3 convolutional layers. As shown in Figure 4.10, 3 Con2D layers were used to abstract

features from the screenshots. The activation functions were kept the same. After convolutional layers, we flattened the abstracted data and attached 2 fully connected layers. The last layer used 1 single neuron and the linear function as the activation function to produce a numerical aesthetic rating prediction.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_18 (Conv2D) | (None, 54, 54, 32) | 11648 |
| conv2d_19 (Conv2D) | (None, 50, 50, 32) | 25632 |
| conv2d_20 (Conv2D) | (None, 48, 48, 64) | 18496 |
| flatten_6 (Flatten) | (None, 147456) | 0 |
| dense_15 (Dense) | (None, 128) | 18874496 |
| dropout_5 (Dropout) | (None, 128) | 0 |
| dense_16 (Dense) | (None, 64) | 8256 |
| dense_17 (Dense) | (None, 1) | 65 |

Total params: 18,938,593
Trainable params: 18,938,593
Non-trainable params: 0

Figure 4.10. Summary Information of the Convolutional Neural Network with 3 Conv2D Layers.

The Stochastic Gradient Descent was used as the optimizer since we have the computation resources for a small dataset and a shallow CNN. Even though an additional convolutional layer was added, the increase in computation is not significant considering

that Google colab provided sufficient computation resources for users. The mean absolute error is used as the loss function and the mean squared error is used to assess the performance of the model. The configuration is presented in Table 4.5.

Table 4.5. Configuration of Compiler for the CNN Model with Three Conv2D Layers.

| **Compiler Configuration** | |
|---|---|
| Optimizer | stochastic gradient descent |
| Loss function | mean absolute error |
| Metrics | mean squared error |
| **Optimizer Configuration** | |
| Learning Rate | 1.00E-06 |
| Decay | 1.00E-06 |
| Momentum | 0.9 |
| Nesterov | TRUE |

The training process is shown in Figure 4.11. As we can see in Figure 4.11, with the increase in epoch (iterations of training process), our model loss (mean absolute error) kept decreasing. It means the model got better and better at fitting the data given. The model loss and mean squared error nearly stopped decreasing after around 15 epochs, which took longer than the CNN with only 2 Conv2D layers. It makes sense because CNN with more layers is more sophisticated to catch more complex patterns, which

would also require more time to train. After 15 epochs, the model loss and mean squared error barely changed.

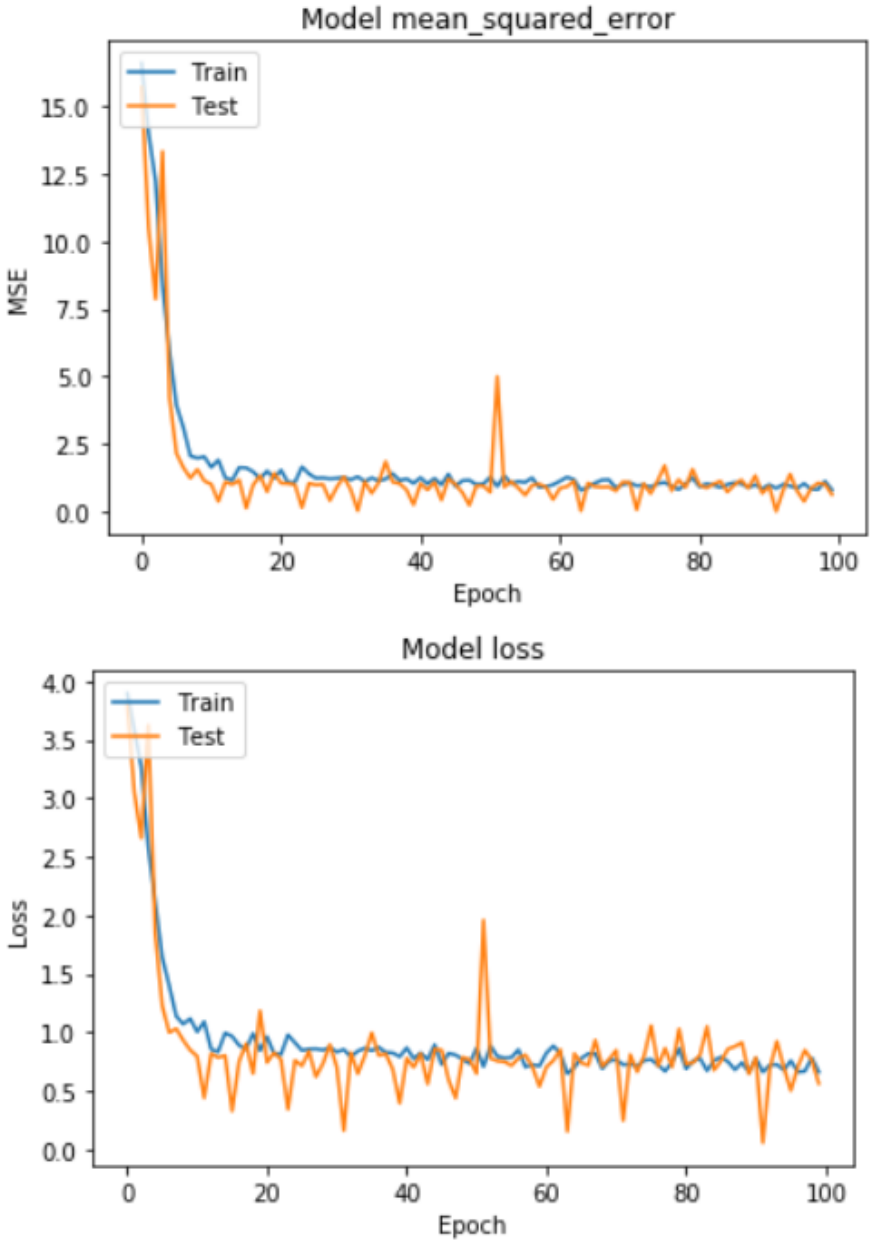



Figure 4.11. Learning Curves of the CNN Model with 3 Conv2D Layers. Mean Squared Error Curve (Upper). Model Loss (Mean Absolute Error) Curve (Lower).

After training the model, we evaluated the model with 3 convolutional layers. The result is shown in Table 4.6. Compared to the model with 2 Conv2D layers, the model performance is a little bit worse but not that much. The R-squared score is a negative value close to zero. Overall, the model gives predictions with an error of less than 1 (the mean absolute error is less than 1), but these predictions do not have much to do with the true score (R-squared is negative). The results are not as favorable as the models using all the aesthetic features. Thus, we decided to increase the depth of the model to 4 Conv2D layers and see if the performance would improve.

Table 4.6. Evaluation Results of the CNN Model with Three Conv2D Layers.

| **Evaluation Result** | |
|---|---|
| Mean Absolute Error | 0.840768 |
| R-squared | -0.180009 |
| Mean Squared Error | 1.064314 |
| Root Mean Squared Error | 1.031656 |

**4.2.3. Convolutional Neural Network with 4 Conv2D Layers.** The table in Figure 4.12 shows the summary information on the architecture and parameters of the CNN with 4 convolutional layers. As shown in Figure 4.12, 4 Con2D layers were used to abstract features from the screenshots. The activation functions were kept the same. After convolutional layers, we flattened the abstracted data and attached 2 fully connected layers. The last layer used 1 single neuron and the linear function as the activation function to produce a numerical aesthetic rating prediction.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_28 (Conv2D)           (None, 54, 54, 32)        11648
_________________________________________________________________
conv2d_29 (Conv2D)           (None, 50, 50, 32)        25632
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 48, 48, 64)        18496
_________________________________________________________________
conv2d_31 (Conv2D)           (None, 46, 46, 64)        36928
_________________________________________________________________
flatten_9 (Flatten)          (None, 135424)            0
_________________________________________________________________
dense_22 (Dense)             (None, 128)               17334400
_________________________________________________________________
dense_23 (Dense)             (None, 1)                 129
=================================================================
Total params: 17,427,233
Trainable params: 17,427,233
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.12. Summary Information of the Convolutional Neural Network with 4 Conv2D Layers.

The Stochastic Gradient Descent was used as the optimizer since we have enough computation resources for a small dataset and a shallow CNN. The mean absolute error is used as the loss function and mean squared error is used to measure the performance of the model. The configuration is shown in Table 4.7.

The training process is shown in Figure 4.13. As we can see in Figure 4.13, with the increase in epoch (iterations of training process), our model loss (mean absolute error) kept decreasing. The model loss and mean squared error nearly stopped decreasing after around 35 epochs. After 35 epochs, the model loss and mean squared error barely

changed. It took a long time for the model to converge. One possible reason is that we have a deeper model and the complexity is high, which requires a longer time to learn.

Table 4.7. Configuration of Compiler for the CNN Model with Four Conv2D Layers.

| **Compiler Configuration** | |
|---|---|
| Optimizer | stochastic gradient descent |
| Loss function | mean absolute error |
| Metrics | mean squared error |
| **Optimizer Configuration** | |
| Learning Rate | 1.00E-06 |
| Decay | 1.00E-06 |
| Momentum | 0.9 |
| Nesterov | TRUE |

After training our model, we evaluated the model with 4 convolutional layers. The results are shown in Table 4.8.

Compared to the model with 3 Conv2D layers, the model performance scores are a little bit better but not that much. The R-squared score is still a negative value close to zero. Overall, the model gives predictions with an error of less than 1 (mean absolute error is 0.747), but these predictions cannot match the true scores very well (R-squared is still negative). The results are still not as favorable as the models using all the aesthetics

features. Thus, we decided to increase the depth of the model to 5 conv2D layers as the final trial of increasing the depth of the model.

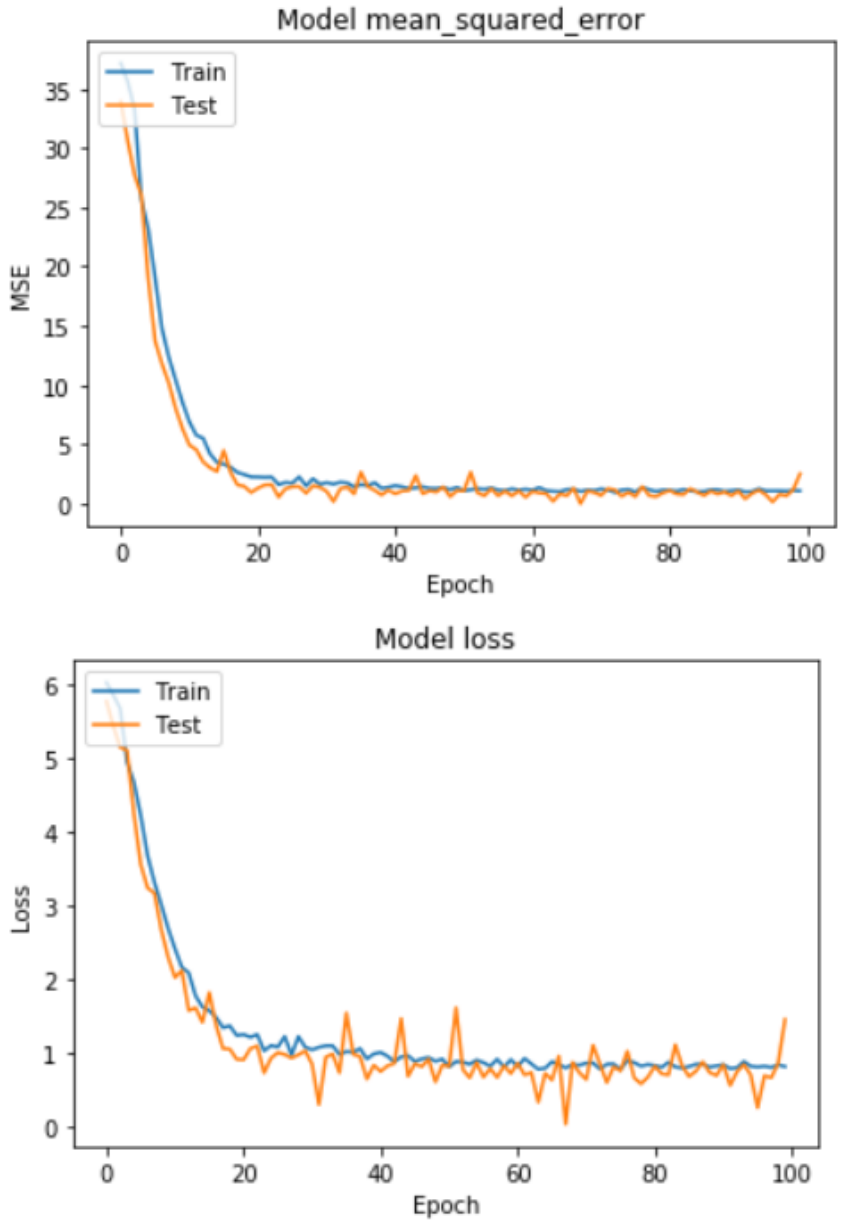



Figure 4.13. Learning Curves of the CNN Model with 4 Conv2D Layers. Mean Squared Error Curve (Upper). Model Loss (Mean Absolute Error) Curve (Lower).

Table 4.8. Evaluation Result of Model CNN Model with Four Conv2D Layers.

| **Evaluation Result** | |
|---|---|
| Mean Absolute Error | 0.747266 |
| R-squared | -0.005201 |
| Mean Squared Error | 0.906645 |
| Root Mean Squared Error | 0.952179 |

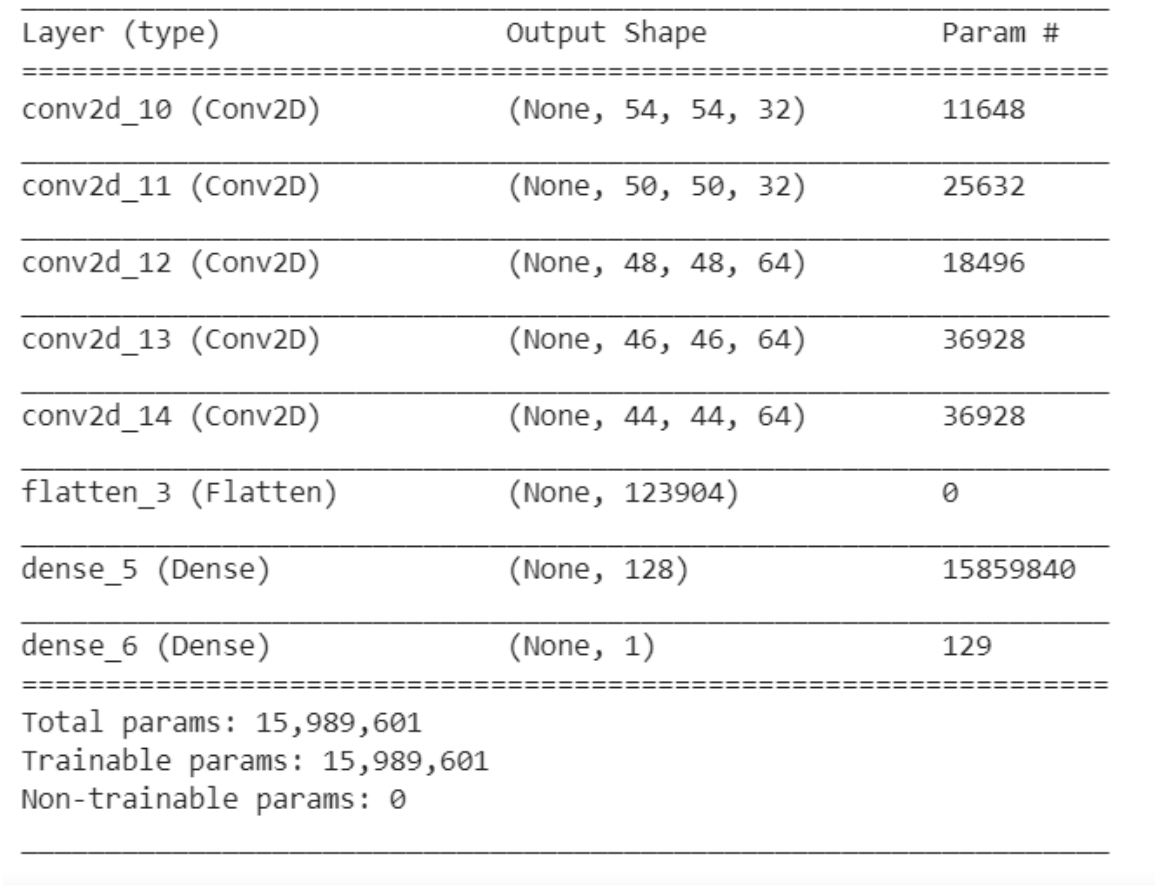
**4.2.4. Convolutional Neural Network with 5 Conv2D Layers.** Figure 4.14 shows the summary information on the architecture and parameters of the CNN with 5 convolutional layers.

As shown in Figure 4.14, 5 Con2D layers were used to abstract features from the screenshots. The activation functions were kept the same as the relu function. After convolutional layers, we flattened the abstracted data and attached 2 fully connected layers. The last layer used 1 single neuron and the linear function as the activation function to produce a numerical aesthetic rating prediction.

The Stochastic Gradient Descent was used as the optimizer since we have enough computation resources for a small dataset and a shallow CNN. The mean absolute error is used as the loss function and mean squared error is used to measure the performance of the model. The configuration is set as in Table 4.9.

The training process is as in Figure 4.15. As we can see in Figure 4.15, with the increase in epoch (iterations of training process), our model loss (mean absolute error) kept decreasing. The model loss and mean squared error nearly stopped decreasing after

around 18 epochs. After 18 epochs, the model loss and mean squared error barely changed.



| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_10 (Conv2D) | (None, 54, 54, 32) | 11648 |
| conv2d_11 (Conv2D) | (None, 50, 50, 32) | 25632 |
| conv2d_12 (Conv2D) | (None, 48, 48, 64) | 18496 |
| conv2d_13 (Conv2D) | (None, 46, 46, 64) | 36928 |
| conv2d_14 (Conv2D) | (None, 44, 44, 64) | 36928 |
| flatten_3 (Flatten) | (None, 123904) | 0 |
| dense_5 (Dense) | (None, 128) | 15859840 |
| dense_6 (Dense) | (None, 1) | 129 |

Total params: 15,989,601
Trainable params: 15,989,601
Non-trainable params: 0

Figure 4.14. Summary Information of Convolutional Neural Network with 5 Conv2D Layers.

After training our model, we evaluated the model with 5 convolutional layers. The results are shown in Table 4.10. Compared to the model with 4 Conv2D layers, the model performance scores are a little bit worse but not that much. The R-squared score is a negative value close to 0. The model gives predictions with an error of less than 1 (the mean absolute error is 0.742), but these predictions do not match the true score very well

(R-squared is still negative). The results are still not as favorable as the models using all the aesthetic features.

Table 4.9. Configuration of Compiler for the CNN Model with Five Conv2D Layers.

| **Compiler Configuration** | |
|---|---|
| Optimizer | stochastic gradient descent |
| Loss function | mean absolute error |
| Metrics | mean squared error |
| **Optimizer Configuration** | |
| Learning Rate | 1.00E-06 |
| Decay | 1.00E-06 |
| Momentum | 0.9 |
| Nesterov | TRUE |

Over the trials of increasing the number of convolutional layers of CNN models, it is found that the models did achieve good performance on mean absolute error, mean squared error and root mean squared error. However, the predictions are rather irrelevant to the true user ratings. Overall, using a shallow convolutional neural network trained on the screenshots of webpages could not achieve performance as good as normal machine learning models using aesthetic features. This shortcoming of deep learning models is especially obvious by looking at the R-squared scores.

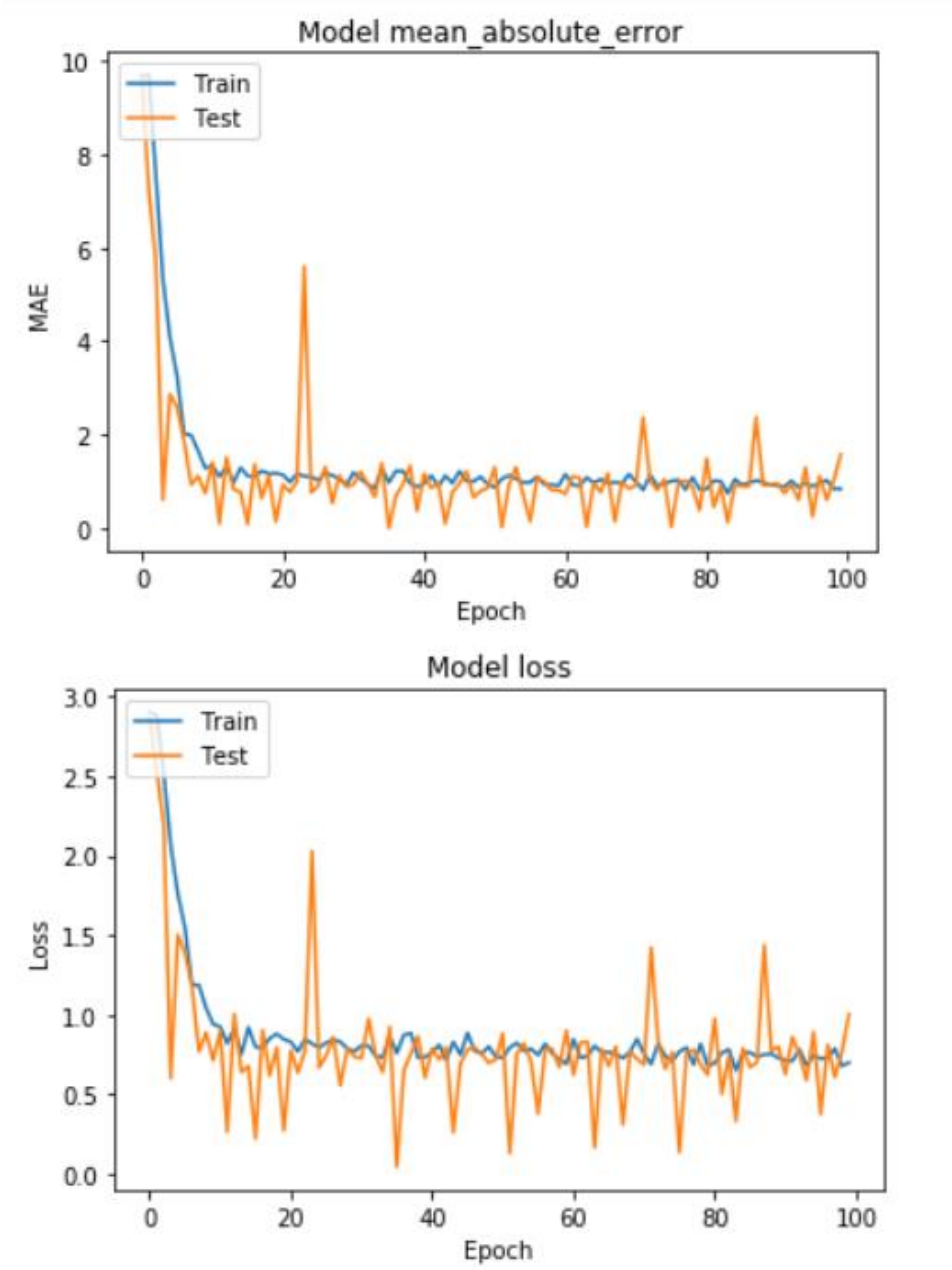



Figure 4.15. Learning Curves of the CNN Model with 5 Conv2D Layers. Mean Squared Error Curve (Upper). Model Loss (Mean Absolute Error) Curve (Lower).

Since the increase in performance brought about by adding convolutional layers is negligible or not significant, I decided to use the transfer learning technique with the

objective of achieving better performance. The details of transfer learning will be discussed in following sections.

Table 4.10. Evaluation Result of the CNN Model with Five Conv2D Layers.

| **Evaluation Result** | |
|---|---|
| Mean Absolute Error | 0.742178 |
| R-squared | -0.004758 |
| Mean Squared Error | 0.906245 |
| Root Mean Squared Error | 0.951969 |

**4.2.5. NIMA NasNet Model.** Since deep learning can give full play to its performance only when the amount of data is adequate, we believe that the insufficient amount of data may be the reason for the poor performance of the previous models. Since we only have screenshots of 398 websites, the objective prediction of aesthetics is a challenging task for computers. Transfer learning is a deep learning technology that allows models to transfer knowledge from a dataset to another similar one. The advantage of transfer learning is that one does not need large amounts of data to apply a trained model to a new task, which might be exactly what we need for this webpage aesthetic rating task.

We decided to use a NasNet neural network, which is a type of CNN architecture. The NasNet that we used is from Google`s NIMA (Neural Image Assessment) research (Talebi & Milanfar, 2018) about predicting image aesthetics using CNN. The networks

were trained on both ImageNet and Aesthetic Visual Analysis (AVA) datasets. ImageNet is a very large database designed for visual recognition tasks and research. There are over 14 million images of different classes of objects. The AVA dataset is a large-scale database designed for photography competitions. Each photo was rated by an average of 200 users based on the aesthetic quality of the images.

As a first attempt at using the NasNet Model, we tried to use the model to directly predict the aesthetic quality of webpage screenshots without any training on our webpage aesthetics dataset. The results of the evaluation are listed in Table 4.11.

Table 4.11. Evaluation Results of NIMA NasNet without Fine-tuning.

| **Evaluation Results** | |
|---|---|
| Mean Absolute Error | 0.957529 |
| R-squared | -0.673707 |
| Mean Squared Error | 1.509606 |
| Root Mean Squared Error | 1.228660 |
| Pearson Correlation | 0.238258 |
| 2 Tailed P-value | 0.0181522 |

We found that the ratings given by the NIMA NasNet model and the ground truths are correlated to some degree. The Pearson correlation is around 23.8% and the 2 tailed p-value is 0.018 <0.05, which means the two sets of data are correlated. It also has a pretty good performance on mean absolute error, mean squared error and root mean

squared error. However, the predictions did not fit well with the ground truths as the R-squared is around -0.67.


NIMA NasNet Score: 4.959

User Average Rating: 4.926


NIMA NasNet Score: 5.275

User Average Rating: 2.408


NIMA NasNet Score: 4.907

User Average Rating: 4.818


NIMA NasNet Score: 4.718

User Average Rating: 1.487

Figure 4.16. Examples of NIMA NasNet Predicting Aesthetics of Webpage.

We selected some webpage screenshots from the testing dataset that the NIMA NasNet model predicts the best and the worst. Figure 4.16 shows part of the webpage

screenshots with corresponding aesthetic predictions by NIMA NasNet (without any training on the webpages) and user average ratings. More screenshots and scores can be found in the appendix section.

Figure 4.16 presented some prediction examples of the NIMA NasNet model. The predictions on the left are more accurate while the ones of the right are less accurate. We can see that the screenshots of the left column are less complex and colorful than the ones on the right column. The NIMA NasNet model (without fine-tuning) seemed to do a good job in predicting these screenshots. However, these screenshots are just a few samples. To get concrete conclusions or insights, more research is needed to assess why the NIMA NasNet predicted well on some screenshots and bad on others. It might lead to the clue to discover the difference between predicting image aesthetics and predicting webpage aesthetics.

To fine-tune the NasNet, we decided to train the NasNet with replaced fully connected layers. First, we replaced a few top layers. We removed the top layer, which is a classification layer using softmax as the activation function. The softmax function is a frequently used activation function for classification problems. Since we formulate the aesthetic prediction as a regression problem, we decided to remove this classification's top layer. We added a regression layer as the top layer using the linear function as the activation function. We added a fully connected layer with 128 neurons, a dropout layer and a regression layer with only 1 neuron, then we trained the top layers. The training process is plotted in Figure 4.17, the configurations are listed in Table 4.12 and the evaluation results are in Table 4.13.

Table 4.12. Compiler Configuration for the NasNet Model Fine-tuned on Fully Connected Layers.

| **Compiler Configuration** | |
|---|---|
| Optimizer | stochastic gradient descent |
| Loss function | mean absolute error |
| Metrics | mean squared error |
| **Optimizer Configuration** | |
| Learning Rate | 1.00E-04 |
| Decay | 1.00E-06 |
| Momentum | 0.9 |
| Nesterov | TRUE |

From Table 4.13 of evaluation results, we can see that the NasNet with its fully connected layers trained on the webpage screenshot dataset was improved by the training. But the improvement is not as good as expected. From the learning curve in Figure 4.17, we can see that the errors and losses were decreasing as the epoch increased. It can be inferred that the model's error in predicting aesthetics is decreasing. However, the R-squared score of the trained model is still negative.

To further improve the model, I decided to unfreeze part of the convolutional layers inside of the NasNet (unfreeze layers higher than the $761^{th}$ layer). After training the convolutional layers of NasNet, a performance evaluation was conducted, and the results are shown in Table 4.13.

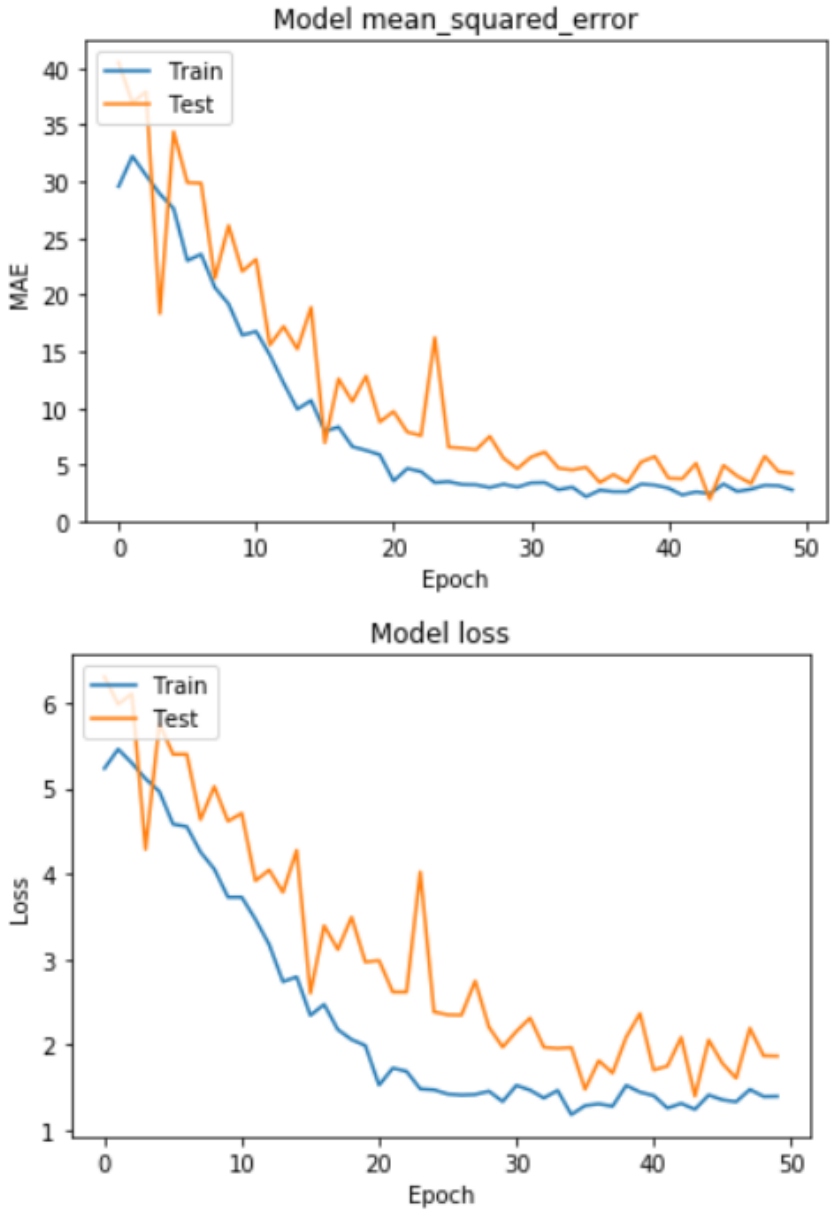



Figure 4.17. Learning Curves of NasNet Model Fine-tuned on Fully Connected Layers. Mean Squared Error Curve (Upper). Model Loss (Mean Absolute Error) Curve (Lower).

From the evaluation results, we can see that the mean absolute errors, mean squared errors and root mean squared errors are further improved but the R-squared score is still negative. One possible guess is that the results of training did help the model

improve to some point, but the true pattern of webpage aesthetics was not caught. The model is probably predicting values around the mean value of average user ratings. However, this guess is based on the rule of thumb. More data are required to verify this point and to help deep learning models capture the patterns.

Table 4.13. Comparison of Evaluation Results of the NasNet Fine-tuned on Fully Connected Layers and Part of the Convolutional Layers.

| **Evaluation Results** | | |
|---|---|---|
| | Before Unfreezing | After Unfreezing |
| Mean Absolute Error | 0.794741 | 0.760792 |
| R-squared | -0.164904 | -0.048134 |
| Mean Squared Error | 1.050690 | 0.945369 |
| Root Mean Squared Error | 1.025032 | 0.972301 |

**4.2.6. NIMA MobileNet Model.** Similar to the NasNet neural network, MobileNet is also a type of CNN architecture that has the feature of being lightweight. The MobileNet is very appropriate to be applied to mobile phones, drones or other devices which lack computation power. The MobileNet we use also comes from Google's NIMA (Neural Image Assessment) research (Talebi & Milanfar, 2018) that predicts image aesthetics using CNN. In this section, we will fine-tune the MobileNet and observe its performance.

As the primary stage of fine-tuning the MobileNet, we decided to first replace a few top layers. We removed the top layer, which is a classification layer using softmax as

the activation function. We added a fully connected layer with 128 neurons, a dropout layer and a regression layer with only 1 neuron, then we trained the top layers. The learning curves are shown in Figure 4.18. The configurations are listed in Table 4.14 and the evaluation results are in Table 4.15.

Table 4.14. Compiler Configuration of MobileNet Fine-tuned on Fully Connected Layers.

| **Compiler Configuration** | |
|---|---|
| Optimizer | stochastic gradient descent |
| Loss function | mean absolute error |
| Metrics | mean squared error |
| **Optimizer Configuration** | |
| Learning Rate | 1.00E-03 |
| Decay | 1.00E-06 |
| Momentum | 0.9 |
| Nesterov | TRUE |

From the learning curves in Figure 4.18, we can see that the error and loss kept decreasing as learning epochs increased. The MobileNet fine-tuned on fully connected layers achieved good scores on most of the error metrics such as mean absolute error, root mean squared error, etc. However, the R-squared score is negative, which suggests that even though the model might give a prediction with a small error, the predictions were not associated with the true user ratings.

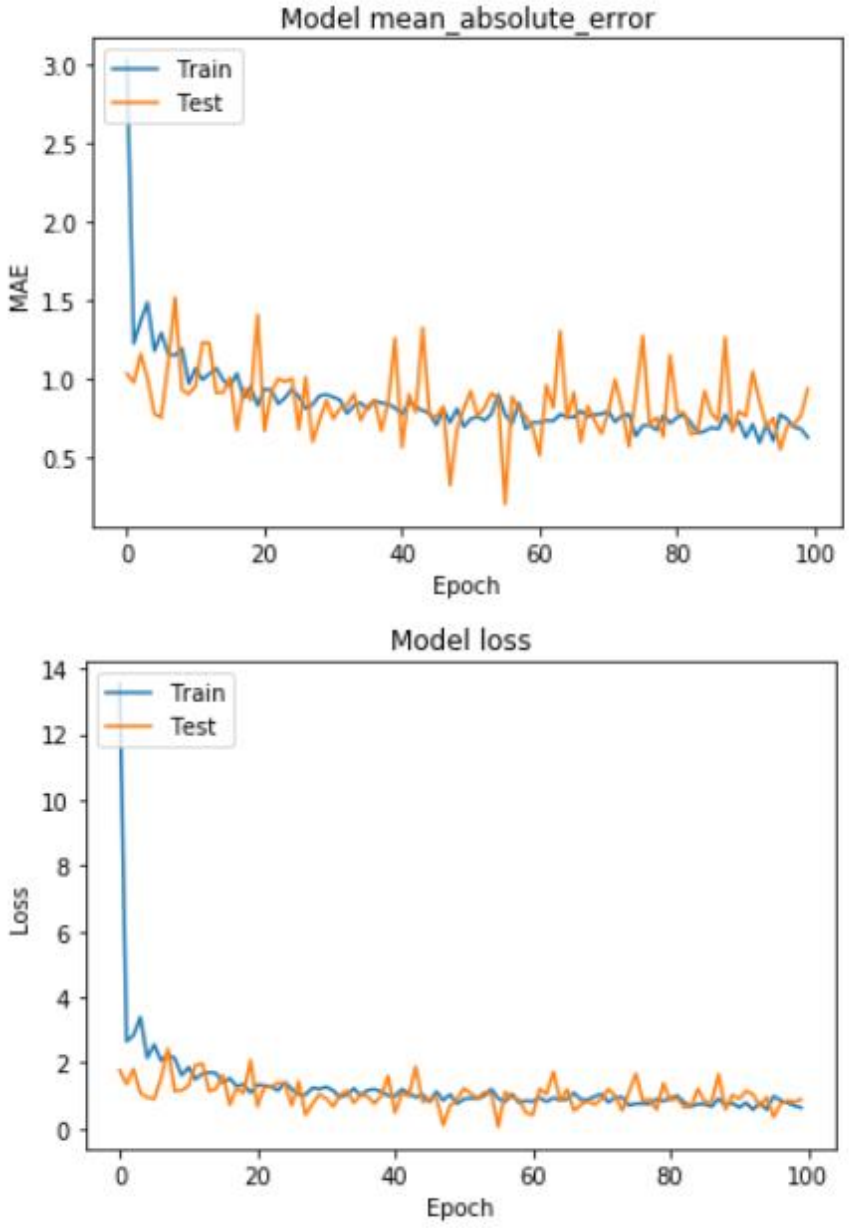



Figure 4.18. Learning Curves of MobileNet Model Fine-tuned on Fully Connected Layers. Mean Squared Error Curve (Upper). Model Loss (Mean Absolute Error) Curve (Lower).

We decided to further the MobileNet by unfreezing part of the convolutional layers within the original MobileNet architecture (unfreeze layers higher than the 59$^{th}$ layer). For the details of the fine-tuning, please refer to the Jupyter notebook. The

evaluation results are shown in Table 4.15. The mean absolute error, mean squared error and root mean squared error are slightly better after training on convolutional layers but the difference is not significant. The R-squared score is still negative.

Table 4.15. Comparison of Evaluation Results of the MobileNet Fine-tuned on Fully Connected Layers and Part of the Convolutional Layers.

| **Evaluation Results** | | |
|---|---|---|
| | Before Unfreezing | After Unfreezing |
| Mean Absolute Error | 0.885259 | 0.821984 |
| R-squared | -0.304059 | -0.109418 |
| Mean Squared Error | 1.176201 | 1.000644 |
| Root Mean Squared Error | 1.084528 | 1.000322 |

**4.2.7. NIMA Inception-ResNet-v2 Model.** Inception-ResNet-v2 is an architecture of the CNN. The Inception-ResNet-v2 model we used is employed from Google`s NIMA (Neural Image Assessment) research (Talebi & Milanfar, 2018). This neural network was pre-trained on both ImageNet and AVA datasets. This neural network learned rich features from a wide range of images. Thus, we decided to fine-tune the Inception-ResNet-v2 for our task. The pre-trained weights are provided by Somshubra Majumdar (2019) on GitHub at https://github.com/titu1994/neural-image-assessment.

From previous experience of training NasNet and MobileNet, we also trained the Inception-ResNet-v2 network on the webpage screenshots. The configurations are shown

in Table 4.16. As the initial stage of fine-tuning, I first trained the fully connected layers. The learning curves are shown in Figure 4.19. It shows the same pattern as the previous deep learning models. The loss and error of the model are decreasing to a certain level.

Table 4.16. Compiler Configuration of the Inception-ResNet-v2.

| **Compiler Configuration** | |
|---|---|
| Optimizer | stochastic gradient descent |
| Loss function | mean absolute error |
| Metrics | mean squared error |
| **Optimizer Configuration** | |
| Learning Rate | 1.00E-03 |
| Decay | 1.00E-06 |
| Momentum | 0.9 |
| Nesterov | TRUE |

As mentioned earlier, the evaluation results are shown in Table 4.17. I unfroze part of the inner architecture (layers higher than the 765$^{th}$ layer) of the Inception-ResNet-v2 model. A comparison of their evaluations is shown in Table 4.17 where the performance has improved. The evaluations show that the Inception-ResNet-v2 network has a good performance on mean absolute error, mean squared error and root mean squared error. Before unfreezing part of the convolutional layers, the R-squared score was negative. After unfreezing part of the convolutional layers, the R-squared score

improved to a positive value. However, the positive score is still too small to be regarded as a good R-squared score. One possible guess is that the model is fitting the scores instead of learning the webpage screenshots.

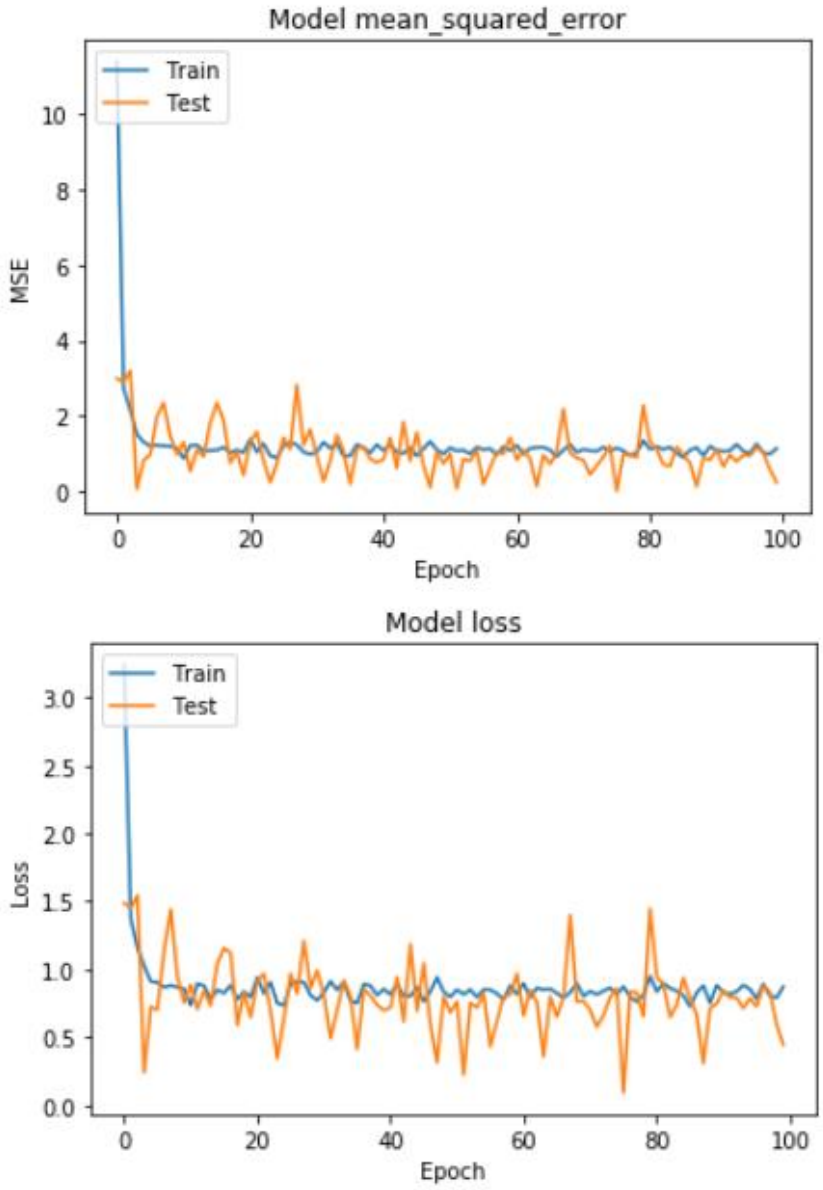



Figure 4.19. Learning Curves of the Inception-ResNet-v2 Training on Fully Connected Layers. Mean Squared Error Curve (Upper). Model Loss (Mean Absolute Error) Curve (Lower).

Table 4.17. Comparison of Evaluation Results of the Inception-ResNet-v2 Fine-tuned on Fully Connected Layers and Part of the Convolutional Layers.

| **Evaluation Results** | | |
|---|---|---|
| | Before Unfreezing | After Unfreezing |
| Mean Absolute Error | 0.793607 | 0.744601 |
| R-squared | -0.117993 | 0.009567 |
| Mean Squared Error | 1.008378 | 0.893324 |
| Root Mean Squared Error | 1.004180 | 0.945158 |

## 4.3. REGRESSION ANALYSIS

We know that complexity and colorfulness are important indicators of aesthetic ratings provided by previous studies (Reinecke & Gajos, 2014; Reinecke et al., 2013). They are expected to be important estimators of aesthetic ratings. Let`s explore their relationships below.

**4.3.1. Analysis of Complexity.** We examine the relationship between webpage complexity and aesthetics next.

**4.3.1.1. Linear regression.** Figure 4.20 shows the scatter plot reflecting the relationship between complexity and aesthetic rating. This figure shows that aesthetic rating tends to decrease with an increase in complexity. The feature ‘complexity’ has a negative relationship with the target variable ‘mean_response’, which is the averaged aesthetic ratings given by the participants. We can also see that the data points are clustered in the center of the figure, which has the ‘complexitymodel’ value around 5 and ‘mean_response’ value around 4.5. Further, we can see that there are few points existing

in the upper-right part of the figure. The upper-right part is an area for webpages with very high complexity and very high aesthetics. It may suggests that it is quite rare for webpages with both very high complexity rare and high aesthetics. To verify this, we need more data of webpages with high aesthetics. Future research should be noted about this.

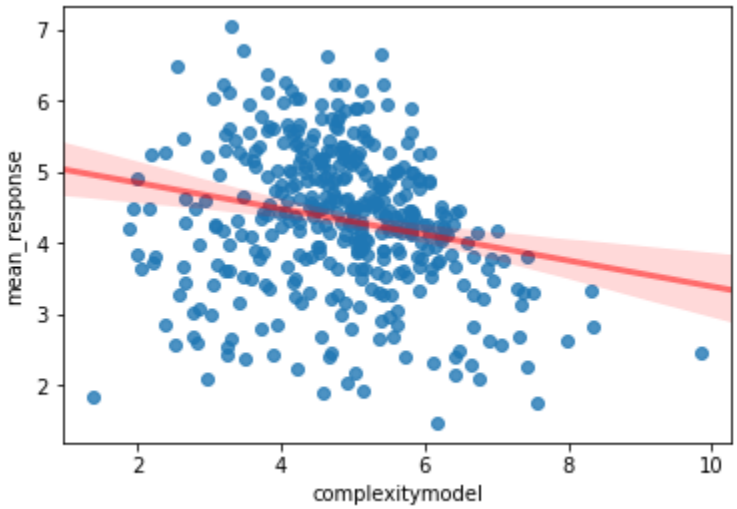



Figure 4.20. Scatter Plot with a Linear Regression Line of Complexity and Aesthetic Rating.

**4.3.1.2. Locally weighted average scatterplot smoothing (lowess).** Figure 4.21 shows a scatter plot with lowess smooth of complexity and aesthetic rating (mean_response). It shows that aesthetic ratings are decreasing rapidly at a high level of complexity. We also observed that there is a slight decrease in aesthetic ratings at a low level of complexity.

Locally weighted average scatterplot smoothing method (lowess) is a non-parametric technique that can create a fit using a smooth curve through points in a scatter

plot by utilizing locally weighted regression. Regression usually can handle most of the problems. However, for data with periodicity and fluctuation, it cannot be simply fitted linearly. Otherwise, the model will have a large error from the truth. Locally weighted regression (lowess) can better deal with this problem. The calculated locally weighted average range moving from left to right, and a continuous curve is fitted.

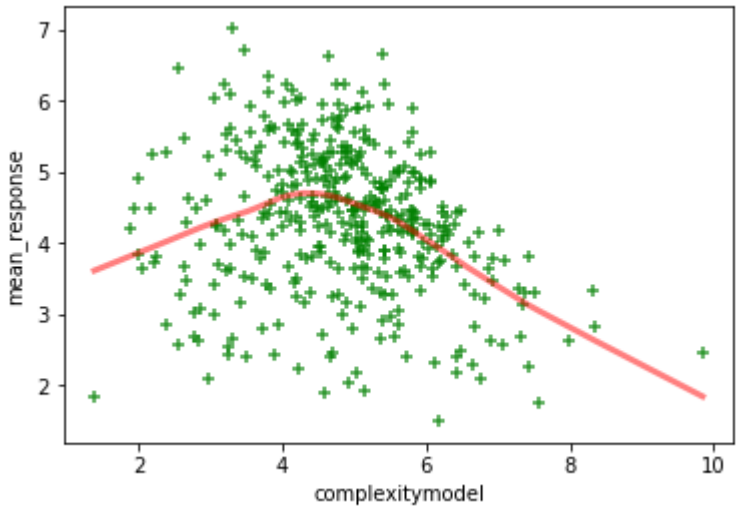



Figure 4.21. Scatter Plot of Complexity and Aesthetic Rating using Lowess Smooth Function

### 4.3.2. Analysis of Colorfulness.

We examine the relationship between webpage colorfulness and aesthetics next.

**4.3.2.1. Linear regression.** Figure 4.22 shows the scatter plot reflecting the relationship between colorfulness and aesthetic rating. This figure shows that aesthetic rating is increasing with an increase in colorfulness. We observed that the data points are scattered around the figure. In the center of the figure, though, the data points are more clustered. The ‘colorfulness’ is around 5.5 in this clustered area.

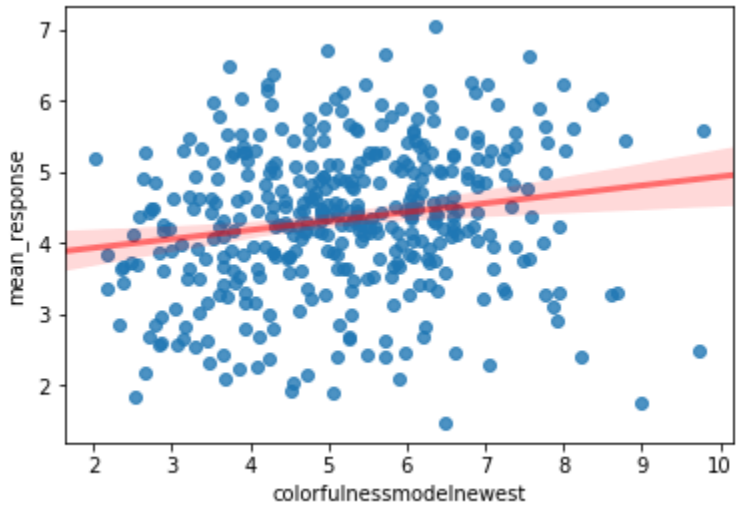



Figure 4.22. Scatter Plot with A Linear Regression Line of Colorfulness (colofulnessmodelnewest) and Aesthetic Rating (mean_response).

**4.3.2.2. Locally weighted average scatterplot smoothing (lowess).** Figure 4.23 shows a scatter plot with a lowess smooth of colorfulness (colofulnessmodelnewest) and aesthetic rating (mean_response).

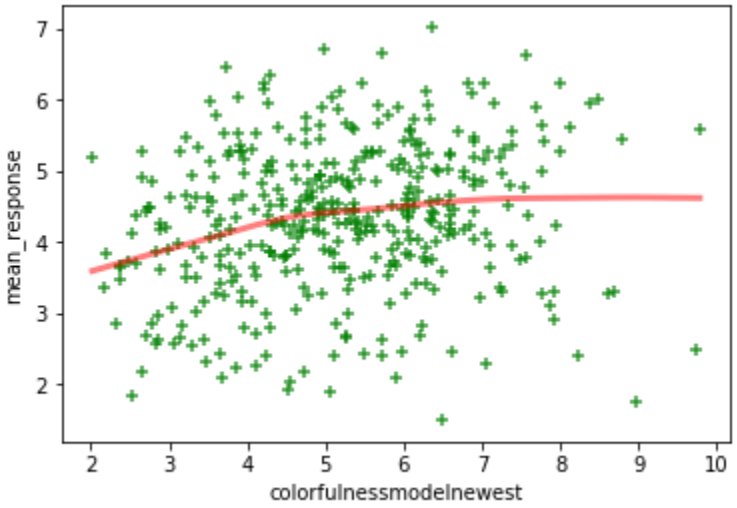



Figure 4.23. Scatter Plot of Colorfulness (colofulnessmodelnewest) and Aesthetic Rating (mean_response) using Lowess Smooth Function

Figure 4.23 shows that aesthetic ratings are slightly increasing when colorfulness increases from a low level to a medium level. And we also observed that there is barely any increase in aesthetic ratings when colorfulness increases from medium level to the high level.

**4.3.3. Why Some Models Have Better Performance.** It is found that the best (gradient boosting regression) and second-best model (random forest regression) are both tree-models. The third belongs to a neural network with 4 layers (20, 15, 10, 5). The three models have much better performance by having a smaller mean absolute error, mean squared error, root mean squared error and larger R squared value. Generally speaking, these three models are superior to the rest of the algorithms. However, the performance difference in this problem is so great that it is worth considering the reasons behind it.

**4.3.3.1. Non-linear relationship.** The top 2 models for this webpage aesthetics prediction problem are both tree models. Tree models have a reasonably good ability for fitting nonlinear relationships. Through the lowess method we conducted, we found that some features might have non-linear relationships with the aesthetic rating. The relationship between complexity and aesthetic rating is a good example. The lowess method showed that aesthetic ratings first increase then decrease with increasing levels of complexity. The non-linear relationships between independent variables and dependent variables could be the reason that makes tree-models superior to other models.

Compared to tree models and neural networks, multiple linear regression is weak in fitting non-linear relationships. For a single decision tree, its capability of capturing the pattern is too weak. Although neural networks have the ability of fitting nonlinear relationships, its interpretability is not strong, and its parameters are not easy to be tuned.

Multiple linear regression models cannot simply fit the non-linear relationships without adding terms with higher orders.

**4.3.3.2. Data noise.** Data noise usually refers to corrupted data such as useless information or erroneous information, which is hard for models to interpret. Our dataset is quite small, and hence, any erroneous data can have a significant impact on our predictive models. Noisy data can also lead to poor performance of the models and increase prediction errors such as bias and variance.

Data noise is a common problem for datasets from the real world. Our cleaned dataset has 46 features and the probability of noise could be substantial. In this thesis, we used feature selection to select the important features to reduce data noise problem.

**4.3.3.3. Over-fitting problem.** Over-fitting problems can happen when the model fits too well on training data to predict well on testing data. High dimensionality and low data volume are two common causes of over-fitting problems. Our dataset has 46 features with 398 rows of data. It is easy for most of the machine learning models to overfit due to the high dimensions and low data volume. For deep learning models, 398 screenshots are far from being enough to learn. Aesthetics is an abstract concept, which can be perceived but very hard to be objectively described or codified. Thus, it can be extremely hard for deep learning models to learn the features of webpages that contribute to or predict aesthetic quality with such a small dataset.

# 5. DISCUSSIONS

The results of our study validated the conclusions of previous research and provide further evidence that aesthetics can be predicted to a certain extent. We have demonstrated that complexity and colorfulness are associated with website aesthetic quality. We found that aesthetic quality is highest when complexity is at a moderate level. Lower complexity and higher complexity can decrease aesthetic quality. This finding is consistent with previous works (Tuch et al., 2012; Reinecke et al., 2013). We also found that the enhancement of colorfulness can improve aesthetic evaluations, but excessive colorfulness barely has any effect on aesthetic evaluations. By using the random forest model to select important features, we found that aesthetics has a strong relationship with some less-noticed features. These key features include non-text area, text area, blue and complexity. Some other features also influence the aesthetic ratings given by users. We found that some color features, such as blue, olive, maroon and so on, influence perceived aesthetic quality. Some features that are correlated with complexity such as 'numOfLeaves' and 'percentageOfLeafArea' are important to webpage aesthetics as well. However, these conclusions need to be confirmed and verified systematically, and more data needs to be collected to increase the credibility of the conclusions.

We also used the deep learning method to predict the aesthetic scores solely based on the screenshots of webpages. However, the results obtained are not better than those predicted by statistical models using aesthetic features. We then used the transfer learning method. However, the prediction given by the deep learning model is not highly correlated with the average user ratings. We speculate that the main reason that the deep

learning approach did not achieve the desired results was that our dataset was too small, with only 398 screenshots of websites. The deep learning method works best when the amount of data is large. Second, webpage screenshots and photos are quite different, with some web elements and text having a lot of complex patterns (e.g., images embedded in webpages, text styles, and buttons). Therefore, it is difficult for deep learning models to learn which of the various and complex features have a real impact on aesthetics. Another reason is that the deep learning model can effectively learn various edges, but it is not sensitive to color information. But color is often very important to web page aesthetics. However, this does not mean that the deep learning method is not suitable for the prediction of aesthetic evaluation. Deep learning research has shown that the aesthetics of photographs are predictable (Talebi & Milanfar, 2018). Moreover, when we use the model that has been pre-trained on photography dataset to directly predict the aesthetic quality of webpages, the predictions given by the model showed a certain level of correlation with the webpage aesthetic ratings. The Pearson correlation was 23.8% and the two-tailed p-value is 0.182. Through this discovery, I suspect that there may be some connection between picture aesthetics and web aesthetics. However, this statement must be verified by collecting more data for systematic investigation. Therefore, future research should collect more picture data to provide a dataset that is big enough for the deep learning method. Different methods can be used to reduce the difficulty of learning, such as blurring the edge of the text content.

# 6. LIMITATIONS AND FUTURE RESEARCH

There are several limitations of this research, which can be addressed by future research.

First, more data needs to be collected. Our dataset consists of just around 400 rows of valid data and 398 screenshots of webpages, which is far from enough for our task. Too little data not only limits the capability of machine learning methods but also makes it easier to obtain biased conclusions. In this study, the amount of data seriously limits the capability of deep learning methods. At present, there is a lack of good data resources on webpage aesthetics. If future research can collect more data of webpage aesthetic quality, it will be a great contribution to research in this direction.

Second, we can look into combining the aesthetic feature method and the deep-learning method. This thesis studies and compares the aesthetic feature method and the deep-learning method. However, data on aesthetic features are limited and fail to cover every aspect of a webpage. Researchers need to explore more aesthetic features using different methods. The deep learning method may require more data and is insensitive to color information, which is something that aesthetic feature methods can make up for. If the two methods can be combined well, a more powerful prediction model can be developed.

Third, there is a need to explore and discover more aesthetic features. In the case of the Reinecke & Gajos' (2014) dataset, higher precision can be achieved by using the aesthetic features with the general models (random forest, Gradient boosting, etc.) than using the deep learning models (NasNet, MobileNet, etc.). However, this conclusion may

change with a large dataset. Discovering and studying more design features and their effects on aesthetics can not only help improve the models but also help provide more comprehensive design guidelines for web designers.

# 7. CONCLUSIONS

In this thesis, we used a variety of machine learning techniques including feature selection, deep learning, and transfer learning to build models that can automatically evaluate the aesthetic quality of webpages. We trained predictive models that can evaluate the aesthetics of a webpage based on the aesthetic features, and ones that can predict the aesthetic score by directly reading the screenshots of a webpage. We also made an exploratory analysis of the effects of complexity and colorfulness on the webpage aesthetics and found that the relationships between them are non-linear. We used the random forest model to find out features that have an important influence on the aesthetics of webpages but are often overlooked by researchers and practitioners. These features include non-text area, text area, blue color and so on. We built models based on these key features and found that using a few but important features can lead to a more accurate and robust model. By doing the feature selection, the risk of data problems is also reduced and the applicability of the models in real world is increased. We also compared the performance of the general models and the deep learning models using various evaluation metrics that are commonly used in the machine learning area. In our case, it is suggested that general models such as random forest and gradient boosting regressor that use aesthetic features are more accurate than the deep learning models.

By selecting important features that are suggested by previous research, I validated the findings of previous literature on the impact of complexity and colorfulness on aesthetic ratings. The relationships from complexity and colorfulness to webpage aesthetics are non-linear. By documenting the findings of this study and providing a

review of the interdisciplinary research area of webpage aesthetics and machine learning, this thesis summarizes the basic knowledge needed to study machine learning methods and aesthetics. It provides an introductory foundation for researchers who are new to this research area and are interested in this research direction. By providing details on the data processing and analysis involving aesthetic features, the thesis also introduces data science techniques such as feature scaling and visualization to the research. By summarizing the standard process of analyzing problems using predictive models, the thesis can also be used as an example for analyzing research questions using a machine learning approach. The thesis also examines the use of a variety of predictive models to automatically evaluate the aesthetics of webpages, which should be interesting for researchers and practitioners who are interested in applying them to answer specific questions. By comparing the difference in model performance and through discussions based on the author's experience in applying machine learning methods, knowledge can be learned or acquired on how to choose an appropriate model and compare the efficiency of models. Although the accuracy and reliability of the models (especially the deep learning models) still need to be further improved, they can be enhanced with more data in the future.  These models can be applied to industry practice with large datasets.

# APPENDIX

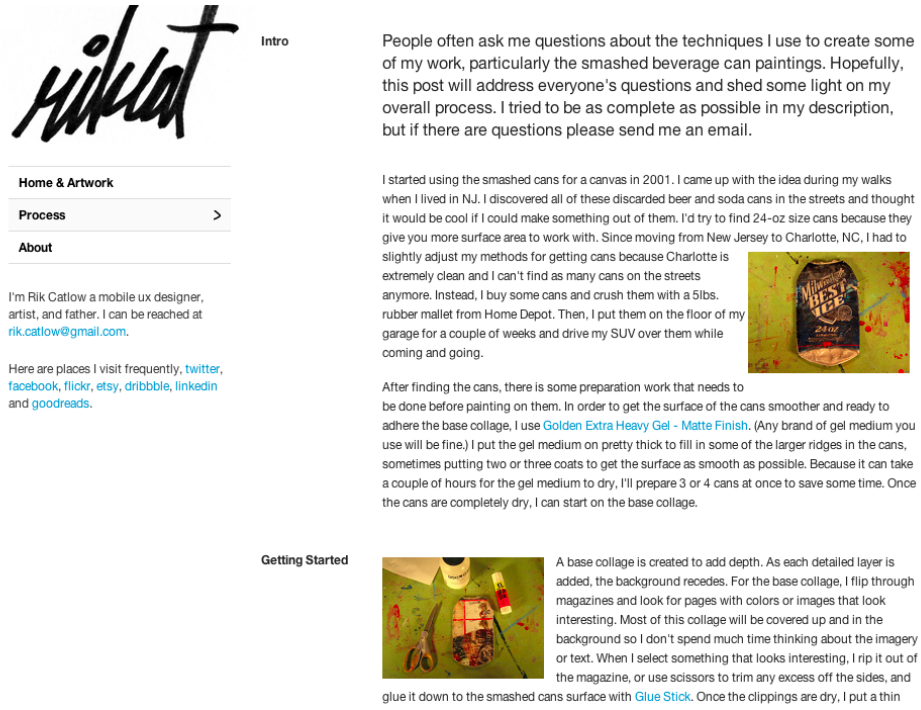
## NIMA NASNET PREDICTIONS ON WEBPAGE SCREENSHOTS


Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/english_90.png

NIMA Score: 4.667 +- (1.089)

True Score: 4.505313496280553

Difference between NIMA and Ground Truth:0.162



rikulat

Home & Artwork

Process >

About

I'm Rik Catlow a mobile ux designer, artist, and father. I can be reached at rik.catlow@gmail.com.

Here are places I visit frequently, twitter, facebook, flickr, etsy, dribbble, linkedin and goodreads.

Intro

People often ask me questions about the techniques I use to create some of my work, particularly the smashed beverage can paintings. Hopefully, this post will address everyone's questions and shed some light on my overall process. I tried to be as complete as possible in my description, but if there are questions please send me an email.

I started using the smashed cans for a canvas in 2001. I came up with the idea during my walks when I lived in NJ. I discovered all of these discarded beer and soda cans in the streets and thought it would be cool if I could make something out of them. I'd try to find 24-oz size cans because they give you more surface area to work with. Since moving from New Jersey to Charlotte, NC, I had to slightly adjust my methods for getting cans because Charlotte is extremely clean and I can't find as many cans on the streets anymore. Instead, I buy some cans and crush them with a 5lbs. rubber mallet from Home Depot. Then, I put them on the floor of my garage for a couple of weeks and drive my SUV over them while coming and going.

After finding the cans, there is some preparation work that needs to be done before painting on them. In order to get the surface of the cans smoother and ready to adhere the base collage, I use Golden Extra Heavy Gel - Matte Finish. (Any brand of gel medium you use will be fine.) I put the gel medium on pretty thick to fill in some of the larger ridges in the cans, sometimes putting two or three coats to get the surface as smooth as possible. Because it can take a couple of hours for the gel medium to dry, I'll prepare 3 or 4 cans at once to save some time. Once the cans are completely dry, I can start on the base collage.

Getting Started

A base collage is created to add depth. As each detailed layer is added, the background recedes. For the base collage, I flip through magazines and look for pages with colors or images that look interesting. Most of this collage will be covered up and in the background so I don't spend much time thinking about the imagery or text. When I select something that looks interesting, I rip it out of the magazine, or use scissors to trim any excess off the sides, and glue it down to the smashed cans surface with Glue Stick. Once the clippings are dry, I put a thin

Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/english_336.png

NIMA Score: 4.959 +- (0.941)

True Score: 4.926940639269406

Difference between NIMA and Ground Truth:0.032


Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/foreign_12.png

NIMA Score: 4.907 +- (0.951)

True Score: 4.818141592920354

Difference between NIMA and Ground Truth:0.089


Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/english_100.png

NIMA Score: 4.939 +- (0.917)

True Score: 4.953846153846154

Difference between NIMA and Ground Truth: -0.015

# Educate-Yourself

*The Freedom of Knowledge, The Power of Thought ©*

Current News | Introduction | Colloidal Silver | Chemtrails | Sylphs | Emerging Diseases | Forbidden Cures |Ozone | Immunity Boosting | Nutrition | The CIA Mind-Body Connection | Ozone | Bioelectrification | Story on Drugs | Vaccine Dangers | Cancer | Newsletter | New World Order | NWO News | Pam Schuffert James Casbolt | Phil Schneider | Al Bielek | Trevor James Constable | Mind Control | Brice Taylor | Ted Gunderson | The Relfes | Free Energy || Montalk Dr. Robert Bitzer | T. Lobsang Rampa | Ruth Drown | ZS Livingstone | David Brandt | Red Elk | Phil Ledoux | Gary Wade | BBB | The Draft | Veterans Awaken Tone Gen | Depleted Uranium | Discussion | Dowsing | Police & Tasers | Rev. Sun Myung Moon | British Israel | The End Times | Amy Goodman Gatekeeper 'Peak Oil' | Amitakh Stanford | Military Draft | Rosie's Predictions | Project Blue Beam | Otto Skorzeny | Insights on Aliens | Cell Towers | Cell Phone Dangers CPS/DCF Tyranny | Adrenal Burnout | The Women Warriors | Orgone Adventures | Dr. John Coleman | Railroading Dr. Jeffrey MacDonald | Henry Makow Bush Family & Nazis | Holistic Dentists | Metal Free Dentistry | Water Supply Sabotage | Dr. Hulda Clark Books | Planet X Sequel | 'Undocumented Immigrants' War on Terror | Tavistock | U.S. Concentration Camps | FEMA | Aliens Are Coming! | Guiding Principles | Global Warming | Gang Stalking | Monoatomic Gold Common Law | Hope | Healing Thought Forms | Vanquish Fear | Prevent Alien/Demonic Attacks | Rethinking Noam Chomsky | Rockefeller File | War is a Racket Letters | Codex Alimentarius | Zeitgeist Refuted | Airport Authoritarianism | Daily Blog | Global Warming | Allies Contact Sheet | Hydrogen Peroxide | Protocols of Sion | Radio Interviews | The Strawman Explained | Swine Flu Hoax/Vaccine | Gary Null on Vaccine Dangers | Don Nicoloff | "Everything is OK" | Daily Blog | Products | Orgone Generators | The Succor Punch | The Mini Silver Terminator | Home | Links | Contact Us/ E-mail

Want to Contact the Editor? First read this

**Translate this page:**

Select Language

Powered by Google™ Translate

Educate-Yourself.org is a free educational forum dedicated to the dissemination of accurate information in the use of natural, non-pharmaceutical medicines and alternative healing therapies in the treatment of disease conditions. Free Energy, Earth Changes, and the growing reality of Big Brother are also explored since *survival itself* in the very near future may well depend on self acquired skills to face the growing threats of bioterrorism, emerging diseases, and the continuing abridgement of constitutional liberties. It is strongly recommended that visitors to this web site *print out hard copies* of the information that is of interest. Do not assume that your hard drive, this web site, or even the Internet itself will always be there to serve you....Ken Adachi, Editor

http://educate-yourself.org/index.shtml

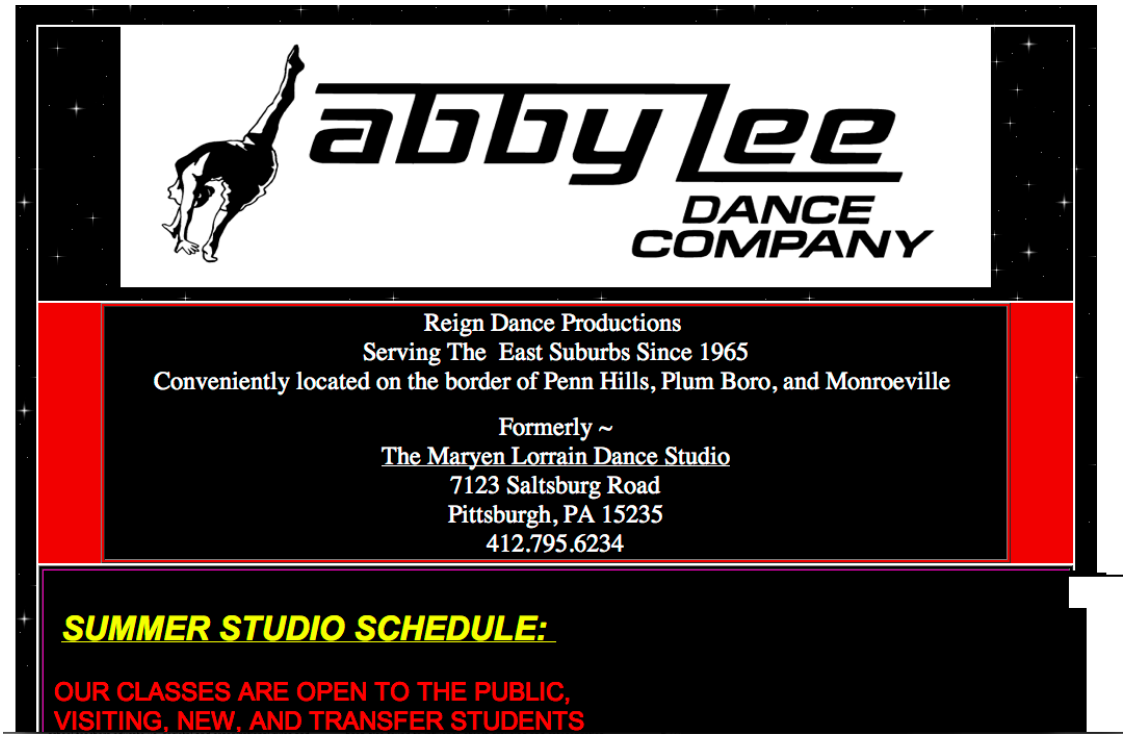
Write Down this Mirror web site address of educate-yourself.org in Switzerland in case you cannot access any page at this web site (our thanks to Stephane Meier for maintaining this mirror site):
http://mirrors.wordsforgood.org/educate-yourself.org/

Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/english_87.png

NIMA Score: 4.722 +- (1.062)

True Score: 2.1668404588112615

Difference between NIMA and Ground Truth:2.556




Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/english_314.png

NIMA Score: 5.454 +- (0.498)

True Score: 2.9968421052631578

Difference between NIMA and Ground Truth:2.458


Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/english_38.png

NIMA Score: 5.275 +- (0.607)

True Score: 2.4079915878023135

Difference between NIMA and Ground Truth:2.867

**UMX Lanyards: Plain, Custom Printed and Badge Holder ID Lanyard Supplies.**

**Lanyards** can be worn on neck, wrist, arm or waist. Lanyards are mainly designed to carry or display ID cards, name badges or security access control identification holders. As a leading lanyard designer, manufacturer and factory direct supplier, the high quality **badge holder** ID lanyards, **badge reels**, retractable and **badge clips** supplies are available in **wholesale** low cost. We supply high-quality and low budget **plain**, blank, non-printed, **custom** imprinted, **neck**, **wrist**, wristband, arm, waist and **safety** breakaway lanyards for **school, college or university** students, business employees, government offices, factory workers, military, **trade shows**, exhibits, EXPO, events, **conventions**, medical hospitals, motels, hotels, restaurants, churches, **fundraising**, promotion giveaway free gifts, **kids** birthday parties, airport security staffs, concert, clubs, sports or meeting. **Key** lanyards, keychain lanyards or **key ring** lanyards are lanyard key holders made for carrying all sort of keys. Multi-color silk screen imprinted, dye-sublimation heat transfer printed or woven logo for **customized crafts** and **personalized** lanyards with a variety of hardware attachment options are available. A great discounted **promotional giveaway** and **ID** lanyards for sale now, you can shop or buy lanyards, name badges, ID holders and retractable from our online lanyard store with lanyard factory direct discount cost. Please check our online catalogs for more detail information.

**Category:** Holders, Reels, Clips, Custom, Safety, Ez, UL, SC, DA, Woven, Wrist, Wristband, Cell-Phone, Leashes, Making, Hooks

**Index:** A B C D E F G H I J K L M N O P Q R **Helpful Pricing Short Cut** **Buy**

A

**Welcome to UMX Lanyard Factory - Manufacturer Direct Lanyards Store**

Large Inventory of High Quality and Low Cost **Plain, Blank, or Non-Printed Lanyards** Available
For School, College or University Student, Business, Convention, Meeting, Event, Fundraising, Government Employees' ID name Badge Holders.
As Small Order As 1 Piece Are Welcome! In Stock - Ship Immediately !

< Click> Super Sale !!! Plain Lanyards - from $0.05 / each High Quality - Low Cost - Factory Direct

| LY-401 1/8" Low Cost Braided Round Cord Plain Lanyards | LY-402 3/8" Economic Cotton & Polyester Flat Blank Lanyards | LY-403 3/8" Low Cost Secured-Breakaway Safety Plain Lanyards. | LY-EC-32-BH180 Package Deal: Lanyard + Holder Super Low Budge Pre-Assembled | LY-421-BH-180 Small Money For Big Event Package Deal Pre-Assembled | LY-422-BH-180 Package Deal: Adjustable Length High Quality Pre-Assembled |
|---|---|---|---|---|---|
| | | Safety Breakaway | | | |

B

< Click> Super Sale !!! Badge Holders from $0.05 / each High Quality - Low Cost - Factory Direct

Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/english_309.png

NIMA Score: 4.718 +- (1.039)

True Score: 1.4869281045751634

Difference between NIMA and Ground Truth:3.231


Evaluating: ../content/drive/My Drive/webthetics/Webthetics-master/data/togethe/foreign_33.png

NIMA Score: 5.155 +- (0.699)

True Score: 2.416454622561493

Difference between NIMA and Ground Truth:2.739

# BIBLIOGRAPHY

Altaboli, A., & Lin, Y. (2011, July). Objective and Subjective Measures of Visual Aesthetics of Website Interface Design: The Two Sides of the Coin. In International Conference on Human-Computer Interaction (pp. 35-44). Springer, Berlin, Heidelberg.

Ben-Bassat, T., Meyer, J., & Tractinsky, N. (2006). Economic and Subjective Measures of the Perceived Value of Aesthetics and Usability. ACM Transactions on Computer-Human Interaction, 13(2), 210-234.

Bloch, P. H. (1995). Seeking the Ideal Form: Product Design and Consumer Response. Journal of Marketing, 59(3), 16-29.

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Coursaris, C. K., Swierenga, S. J., & Watrall, E. (2008). An Empirical Investigation of Color Temperature and Gender Effects on Web Aesthetics. Journal of Usability Studies, 3(3), 103-117.

Cyr, D. (2008). Modeling Web Site Design Across Cultures: Relationships to Trust, Satisfaction, and E-Loyalty. Journal of Management Information Systems, 24(4), 47-72.

Cyr, D., Head, M., & Larios, H. (2010). Colour Appeal in Website Design within and across Cultures: A Multi-method Evaluation. International Journal of Human-Computer Studies, 68(1-2), 1-21.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006, May). Studying Aesthetics in Photographic Images Using a Computational Approach. In European Conference on Computer Vision (pp. 288-301). Springer, Berlin, Heidelberg.

Dong, Z., & Tian, X. (2015). Multi-Level Photo Quality Assessment with Multi-View Features. Neurocomputing, 168, 308-319.

Dou, Q., Zheng, X. S., Sun, T., & Heng, P. A. (2019). Webthetics: Quantifying Webpage Aesthetics with Deep Learning. International Journal of Human-Computer Studies, 124, 56-66.

Faria, J., Bagley, S., Rüger, S., & Breckon, T. (2013, July). Challenges of Finding Aesthetically Pleasing Images. In 2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) (pp. 1-4). IEEE.

Feng, B. C. (2015). An Introduction to Neural Networks. Retrieved from https://blog.csdn.net/fengbingchun/article/details/50274471

Fogg, B. J., Soohoo, C., Danielson, D. R., Marable, L., Stanford, J., & Tauber, E. R. (2003, June). How Do Users Evaluate the Credibility of Web Sites? A Study With over 2,500 Participants. In Proceedings of the 2003 Conference on Designing for User Experiences (pp. 1-15). ACM.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 1189-1232.

Ha, J., Haralick, R. M., & Phillips, I. T. (1995, August). Recursive XY Cut using Bounding Boxes of Connected Components. In Proceedings of 3rd International Conference on Document Analysis and Recognition (Vol. 2, pp. 952-955). IEEE.

Hall, R. H., & Hanna, P. (2004). The Impact of Web Page Text-Background Colour Combinations on Readability, Retention, Aesthetics and Behavioural Intention. Behaviour & Information Technology, 23(3), 183-195.

Hasler, D., & Suesstrunk, S. E. (2003, June). Measuring Colorfulness in Natural Images. In Human Vision and Electronic Imaging VIII (Vol. 5007, pp. 87-95). International Society for Optics and Photonics.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).

Hoegg, J., Alba, J. W., & Dahl, D. W. (2010). The Good, the Bad, and the Ugly: Influence of Aesthetics on Product Feature Judgments. Journal of Consumer Psychology, 20(4), 419-430.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

Hutcheson, G. D. (2011). Ordinary Least-Squares Regression. In L. Moutinho and G. D. Hutcheson (eds.), The SAGE Dictionary of Quantitative Management Research (pp. 224-228). SAGE.

Jayesh, B. A. (2018). The Artificial Neural Networks Handbook: Part 4. Retrieved from https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e

Jin, X., Chi, J., Peng, S., Tian, Y., Ye, C., & Li, X. (2016, October). Deep Image Aesthetics Classification Using Inception Modules and Fine-Tuning Connected Layer. In 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP) (pp. 1-6). IEEE.

Kao, Y., He, R., & Huang, K. (2016). Visual Aesthetic Quality Assessment with Multi-Task Deep Learning. arXiv preprint arXiv:1604.04970, 5.

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., & Winnemoeller, H. (2013). Recognizing Image Style. arXiv Preprint arXiv:1311.3715.

Khani, M. G., Mazinani, M. R., Fayyaz, M., & Hoseini, M. (2016, April). A Novel Approach for Website Aesthetic Evaluation Based on Convolutional Neural Networks. In 2016 Second International Conference on Web Research (ICWR) (pp. 48-53). IEEE.

Kim, J., & Moon, J. Y. (1998). Designing Towards Emotional Usability in Customer Interfaces—Trustworthiness of Cyber-Banking System Interfaces. Interacting with computers, 10(1), 1-29.

Kong, S., Shen, X., Lin, Z., Mech, R., & Fowlkes, C. (2016, October). Photo Aesthetics Ranking Network with Attributes and Content Adaptation. In European Conference on Computer Vision (pp. 662-679). Springer, Cham.

Kruft, H. W. (1994). History of Architectural Theory. Princeton Architectural Press.

Lindgaard, G. (1999). Does Emotional Appeal Determine Perceived Usability of Web sites. In Proceedings of CybErg: The Second International Cyberspace Conference on Ergonomics (pp. 202-211).

Lindgaard, G. (2007). Aesthetics, Visual Appeal, Usability and User Satisfaction: What Do the User's Eyes Tell the User's Brain? Australian Journal of Emerging Technologies & Society, 5(1), 1-14.

Lindgaard, G., Fernandes, G., Dudek, C., & Brown, J. (2006). Attention Web Designers: You Have 50 Milliseconds to Make a Good First Impression! Behaviour & Information Technology, 25(2), 115-126.

Liu, Y. (2003). Engineering Aesthetics and Aesthetic Ergonomics: Theoretical Foundations and a Dual-Process Research Methodology. Ergonomics, 46(13-14), 1273-1292.

Lu, X., Lin, Z., Jin, H., Yang, J., & Wang, J. Z. (2014, November). Rapid: Rating Pictorial Aesthetics Using Deep Learning. In Proceedings of the 22nd ACM International Conference on Multimedia (pp. 457-466). ACM.

Lu, X., Lin, Z., Jin, H., Yang, J., & Wang, J. Z. (2015). Rating Image Aesthetics Using Deep Learning. IEEE Transactions on Multimedia, 17(11), 2021-2034.

Lu, X., Lin, Z., Shen, X., Mech, R., & Wang, J. Z. (2015). Deep Multi-Patch Aggregation Network for Image Style, Aesthetics, and Quality Estimation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 990-998).

Mai, L., Jin, H., & Liu, F. (2016). Composition-Preserving Deep Photo Aesthetics Assessment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 497-506).

Maity, R., & Bhattacharya, S. (2017, September). A Model to Compute Webpage Aesthetics Quality Based on Wireframe Geometry. In IFIP Conference on Human-Computer Interaction (pp. 85-94). Springer, Cham.

Martindale, C., Moore, K., & Borkum, J. (1990). Aesthetic Preference: Anomalous Findings for Berlyne's Psychobiological Theory. The American Journal of Psychology, 103(1), 53-80.

McKnight, D. H., Choudhury, V., & Kacmar, C. (2002). Developing and Validating Trust Measures for E-Commerce: An Integrative Typology. Information Systems Research, 13(3), 334-359.

Moshagen, M., Musch, J., & Göritz, A. S. (2009). A Blessing, Not a Curse: Experimental Evidence for Beneficial Effects of Visual Aesthetics on Performance. Ergonomics, 52(10), 1311-1320.

Moss, G. A., & Gunn, R. W. (2009). Gender Differences in Website Production and Preference Aesthetics: Preliminary Implications for ICT in Education and Beyond. Behaviour & Information Technology, 28(5), 447-460.

Ngo, D. C. L., Teo, L. S., & Byrne, J. G. (2003). Modelling Interface Aesthetics. Information Sciences, 152, 25-46.

Postrel, V. (2001). Can Good Looks Really Guarantee a Product's Success. The New York Times, 100(2).

Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81-106.

Reinecke, K., & Bernstein, A. (2011). Improving Performance, Perceived Usability, and Aesthetics with Culturally Adaptive User Interfaces. ACM Transactions on Computer-Human Interaction, 18(2), Article 8, A1-A29.

Reinecke, K., & Gajos, K. Z. (2014, April). Quantifying Visual Preferences Around the World. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 11-20). ACM.

Reinecke, K., Yeh, T., Miratrix, L., Mardiko, R., Zhao, Y., Liu, J., & Gajos, K. Z. (2013, April). Predicting Users' First Impressions of Website Aesthetics with A Quantification of Perceived Visual Complexity and Colorfulness. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (pp. 2049-2058). ACM.

Russo, K. D., Peach, R. K., & Shapiro, L. P. (1998). Verb Preference Effects in the Sentence Comprehension of Fluent Aphasic Individuals. Aphasiology, 12(7-8), 537-545.

Schenkman, B. N., & Jönsson, F. U. (2000). Aesthetics and Preferences of Web Pages. Behaviour & Information Technology, 19(5), 367-377.

Somshubra, M. (2019). Implementation of NIMA: Neural Image Assessment in Keras. Retrieved from https://github.com/titu1994/neural-image-assessment

Sonderegger, A., & Sauer, J. (2010). The Influence of Design Aesthetics in Usability Testing: Effects on User Performance and Perceived Usability. Applied Ergonomics, 41(3), 403-410.

Sumit, S. (2018). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Retrieved from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).

Talebi, H., & Milanfar, P. (2018). Nima: Neural Image Assessment. IEEE Transactions on Image Processing, 27(8), 3998-4011.

Tractinsky, N. (2004). Toward the Study of Aesthetics in Information Technology. ICIS 2004 Proceedings, 62.

Tractinsky, N., Cokhavi, A., Kirschenbaum, M., & Sharfi, T. (2006). Evaluating the Consistency of Immediate Aesthetic Perceptions of Web Pages. International Journal of Human-Computer Studies, 64(11), 1071-1083.

Tractinsky, N., Katz, A. S., & Ikar, D. (2000). What is Beautiful is Usable. Interacting with Computers, 13(2), 127-145.

Tuch, A. N., Presslaber, E. E., StöCklin, M., Opwis, K., & Bargas-Avila, J. A. (2012). The Role of Visual Complexity and Prototypicality Regarding First Impression of Websites: Working Towards Understanding Aesthetic Judgments. International Journal of Human-Computer Studies, 70(11), 794-811.

Vaibhav, S. (2018). Power of a Single Neuron. Retrieved from https://towardsdatascience.com/power-of-a-single-neuron-perceptron-c418ba445095

Wang, H.-F. (2014). Picture Perfect: Girls' and Boys' Preferences Towards Visual Complexity in Children's Websites. Computers in Human Behavior, 31, 551-557.

Wang, W., Zhao, M., Wang, L., Huang, J., Cai, C., & Xu, X. (2016). A Multi-Scene Deep Learning Model for Image Aesthetic Evaluation. Signal Processing: Image Communication, 47, 511-518.

Wang, Z., Chang, S., Dolcos, F., Beck, D., Liu, D., & Huang, T. S. (2016). Brain-Inspired Deep Networks for Image Aesthetics Assessment. arXiv Preprint arXiv:1601.04155.

William, J. (2015). Why is Blue the World's Favorite Color? Retrieved from https://today.yougov.com/topics/international/articles-reports/2015/05/12/why-blue-worlds-favorite-color

Wikipedia Contributors. (2019, October). Biological Neuron Model. Retrieved from https://en.wikipedia.org/w/index.php?title=Biological_neuron_model&oldid=921785827

Wong, L. K., & Low, K. L. (2009, November). Saliency-Enhanced Image Aesthetics Class Prediction. In 2009 16th IEEE International Conference on Image Processing (ICIP) (pp. 997-1000). IEEE.

Wu, Y., Bauckhage, C., & Thurau, C. (2010, August). The Good, the Bad, and the Ugly: Predicting Aesthetic Image Labels. In 2010 20th International Conference on Pattern Recognition (pp. 1586-1589). IEEE.

Yendrikhovskij, S. N., Blommaert, F. J., & de Ridder, H. (1998, January). Optimizing Color Reproduction of Natural Images. In Color and Imaging Conference (Vol. 1998, No. 1, pp. 140-145). Society for Imaging Science and Technology.

Zain, J. M., Tey, M., & Soon, G. Y. (2008, October). Using Aesthetic Measurement Application (AMA) to Measure Aesthetics of Web Page Interfaces. In 2008 Fourth International Conference on Natural Computation (Vol. 6, pp. 96-100). IEEE.

Zhang, C. L., Luo, J. H., Wei, X. S., & Wu, J. (2017, September). In Defense of Fully Connected Layers in Visual Representation Transfer. In Pacific Rim Conference on Multimedia (pp. 807-817). Springer, Cham.

Zheng, X. S., Chakraborty, I., Lin, J. J. W., & Rauschenberger, R. (2009, April). Correlating Low-Level Image Statistics with Users-Rapid Aesthetic and Affective Judgments of Web Pages. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1-10). ACM.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8697-8710).

# VITA

Ang Chen was born in Shihezi, Xinjiang, China. He received his Bachelor`s degree in Petroleum Engineering from both Missouri University of Science & Technology and China University of Petroleum (Hua Dong) in May 2017.

Ang continued to pursue further study in the Department of Information Science & Technology of Missouri University of Science & Technology. In Dec 2019, He received his M.S. in Information Science & Technology from Missouri University of Science & Technology.