Please note:

- This is Chapter 4 from my Master Thesis at University of Saskatchewan.
- This chapter explains the development of a method to quantify one of the aesthetics attributes of design.
- For that prediction, an Artificial Neural Network was trained in MATLAB environment.
- It was published in Canadian Psychology CPA Annual Conventions in Halifax.
- If you need the complete Thesis, please contact me.Thanks, Ryan Mokarian

CHAPTER 4

MEASURMENT OF VISUAL BALANCE

4.1 Introduction

In order to apply the visual balance principle to the uncoupled design variables which were determined in the last experiment, there was a need to develop a quantitative method for measuring visual balance, in particular measuring the degree of visual balance of a product based on both the color and area of its component. For this purpose, after describing balance and its theoretical views, an existing method for measuring visual balance was reviewed. Then a method was presented for quantifying the equivalent weight of any color. The proposed method was then integrated into the existing method, leading to a new method to measure visual balance given to a component which has both area and color.

4.2 Balance

Balance is the achievement of equilibrium in the overall perceived visual weight of a composition among various parts. The balance is not always achieved symmetrically. There are two general types of balance: Symmetrical and Asymmetrical. In the former, the left and right sides are mirror images of each other, whereas in the latter the two sides are not the same. To achieve balance, weight distribution should be designed to be pleasing to the eye, and visual weight should be focused around a perceived center of gravity [Preble and Preble, 1994]. Without balance, a composition looks awkward and unstable. Balance evokes a feeling of stability and confidence in a viewer, while an unbalanced display creates a feeling of stress. Feelings about balance are connected to human experience with the actual physical balance [Preble and Preble, 1994; Reilly and Roach, 1984].

There are some rules for visual balance, and they are: (1) a larger form is heavier; (2) an object near the frame of a picture (i.e. product face) is heavier than an object close to a center; (3) a complex form is heavier than a simple one; (4) warm colors such as red and orange are heavier than cold ones such as blue and green; (5) intense colors are heavier than pale/weak colors; (6) the weight of any color increases as the background color approaches to its complementary color [Preble and Preble, 1994]. By controlling the color, size, tone, and shape of components in a design, one could influence the user's perception of balance [Ngo et al., 2000A; Ngo et al., 2000B; Reilly and Roach, 1984].

4.3 Color Models

Before describing background regarding the roles of area and color in balance, important color models need to be clarified. Colors can be presented based on different models including the *HSL* (Hue, Saturation, and Luminosity) model and the *RGB* (Red, Green, and Blue) model.

In the *HSL* model, each color is presented by its Hue, Saturation, and Luminosity. Hue refers to special wavelength of color to which a name is given. Value, also called luminosity, points to relative lightness or darkness of the color and its gradation is between black and white. Intensity, also called saturation and chrome, refers to the purity of a hue at its highest saturation i.e., in its brightest form [Preble and Preble, 1994].

In the *RGB* color model, each color can be defined by appointing how much of each of the red, green and blue color is contained. The variation of each color is between the minimum which is no color and maximum which is full intensity. In the case that all the colors are at minimum the result is black and when all the colors are at maximum, the result is white. These three colors may be numerically written in different ways. One way is in a range between 0.0 (minimum) to 1.0 (maximum). For instance based on this way of formulating, full intensity blue is 0.0, 0.0, 1.0. Another way is to write the color values in terms of percentages, from 0% (minimum) to 100% (maximum). According to this framework, full intensity blue is 0%, 0%, 100%. Another form of appointing values to the colors is by writing numbers in the range 0 to 255. According to this framework, full intensity blue is 0.0, 0.0, 255.0. This range is frequently found in computer science, where programmers have found it convenient to encode each color value in eight bits. This type of encoding would able the programmers to have 2^8 (= 256) different values. A picture on computer screen consists of a set of pixels. The RGB model is applied to each pixel. For 24-bit color, the triplet (0,0,0) represents black, and the triplet (255, 255, 255) represents white. When the three *RGB* values are set to the same value, for example (63, 63, 63), (127, 127, 127), or (191, 191, 191), the resulting color is a shade of gray.

The presence of all basic colors in sufficient amounts creates pure white, and the absence of all basic colors creates pure black [Murray and Van Ryper, 1996]. Because *RGB* is the most widely used additive color model for television and computer screens [Murray and Van Ryper, 1996], for the study of visual balance in this research the *RGB* model was used.

4.4 Role of Area and Color in Balance

Paul Klee, a well known Swiss-German painter (1879-1940), described a model for understanding balance. Based on his model, three formal elements of balance are dimension (area), quality (hue or wavelength of the color), and tone (value or luminosity of the color) [San Lazzaro, 1957]. Based on Klee's model, balance can be valued with area and two dimensions of the *HSL* model, hue and luminosity.

Munsell [1905] presented a principle by which balance can be quantified with the area and two dimensions of the *HSL* model (i.e., luminosity and saturation). Based on Munsell's law, the area of color in a combination is inversely proportional to the product of their luminosity and saturation. Experiments showed that Munsell's formula works quite well [Linnett et al., 1991; Morris and Dunlap, 1987].

Pinkerton and Humphrey [1974] compared the heaviness of 5 colors with constant saturation and luminosity. They found the following order from heaviest to lightest: red, blue, green, orange, and yellow. Locher et al. [2005] compared the perceived weight of 3 colors: red, blue, and yellow. In addition, they examined the size of a colored area. They found the following order from heaviest to lightest: red, blue, and yellow. Also, they found that the perceived weight of a color varied as a function of the size of the area it occupied (bigger size, more perceived weight).

Klee's model is concerned with the area and only the hue and luminosity aspects of colors in the *HSL* model. Munsell's model is concerned with the area and only the saturation and luminosity aspects of color in the *HSL* model. On the other hand, Pinkerton and Humphrey [1974] considered the hue aspect of only five colors in the *HSL* model. And finally Locher et al. [2005] considered the area and the hue aspect of three colors in the *HSL* model.

In the following, a model which would be able to quantify the balance of a picture based on the area and color of its components was presented, while all three elements of each color in *RGB*/*HSL* model were considered. It should be noted that with three elements of one model (*RGB*/*HSL* model), three elements of the other models can be determined. The *RGB* model is more comprehensive than the *HSL* model, for the three attributes in the *HSL* model are not as uniform in semantics and measurements as those in the *RGB* model. This might be the reason that in previous studies by others using the *HSL* model, only two attributes were considered. Additionally, this study was not limited to a few numbers of colors; which others did. Further, in the method that was presented in this study, the influence of background color on the perception of foreground color was considered.

4.5 Balance Measurement

A quantification method for determining the balance of the area was presented by Ngo et al. [2003]. They used the rule that larger objects are perceived heavier, whereas smaller objects are perceived lighter. They presented Equations 4.1, 4.2, 4.3, and 4.4 for measuring the area balance in a display.

$$BM = 1 - \frac{\left|BM_{vertical}\right| + \left|BM_{horizontal}\right|}{2} \in [1,0] \tag{4.1}$$

where BM is the total balance. $BM_{vertical}$ and $BM_{horizontal}$ are the vertical and horizontal balances, respectively, with

$$BM_{vertical} = \frac{w_L - w_R}{\max(|w_L|,|w_R|)} \tag{4.2}$$

$$BM_{horizontal} = \frac{w_T - w_B}{\max(|w_T|,|w_B|)} \tag{4.3}$$

when

$$Wj = \sum_{i}^{n_j} a_{ij} d_{ij} \qquad j = L, R, T, B \tag{4.4}$$

Where w, a, d, n, L, R, T, and B stand for weight, area, distance from a central axis, number of components in the j area, left, right, top, and bottom, respectively. The equation developed by Ngo et al. [2003] considered area only for measuring balance.

In this study, Equation 4.4 was extended to include both the geometric area Ag and the equivalent area Ac of the color of a component. Ac of each color is measured with respect to white color. Ac is equivalent to the amount of the white color area which balances a unit area of the color. Equations 4.5 and 4.6 are equations that were proposed in this study. Equation 4.5 is the equation for measuring the weight of each component in a plane, and Equation 4.6 is for measuring the weight of each component.

$$Wj = \sum_{i}^{n_j} (Ac + Ag)_{ij} d_{ij} \qquad j = L, R, T, B \tag{4.5}$$

$$W = (Ac + Ag)d \quad \textbf{(4.6)}$$

As mentioned before, there is a sense of "weight" for each colour; yet the relationship between such a weight and a colour is absent in literature. In other words, there does not exist a first principle from the aesthetic literature regarding the equivalent weight of a colour. The equivalent weight of color may contain non-linear characteristics. Therefore, an artificial neural network (ANN) approach, which is able to model nonlinear features, was used in this study for determination of the equivalent weight of color. This approach is explained in Section 4.6.4.

4.6 Determination of the equivalent weight of color

Perception of the weight of each color is under the influence of its background color as well. As Preble and Preble [1994] mentioned, the weight of any color increases as the background color approaches to its complementary color; therefore here, in determination of the equivalent weight of color, its background color was considered in this study.

The general procedure to develop a model for the equivalent weight of color can be summarized. The first step was to design a code system to represent the structure of color. The second step was to generate a collection of samples that were representative based on the code system. The third step was to conduct the experiment to solicit the human subject perception of the "weight" of the color samples; the solicited information served as so-called training data. The forth step was to conduct *ANN* learning to determine the *ANN* model from the training data, so eventually, given a color and its background color to the *ANN* model, the equivalent weight can be obtained.

4.6.1 Step One: Design a code system to represent the structure of color

In the first step, the way of representing colors with codes was introduced. According to the *RGB* model, each color can be decomposed to three basic colors (red, green, and blue) in specific ratios. As well, the variation range of each basic color is from 0 to 255 [Murray and Van Ryper, 1996]. Therefore, it can be established a so-called color space which is spanned by three dimensions as shown in Figure 4.1. More specifically, this is a cube space bounded by three points, i.e., (255, 0, 0), (0, 255, 0), (0, 0, 255). The values which vary between 0 and 255 are dimensionless and as mentioned earlier represent the intensity of red, green and blue colors from 0 (no contribution) to 255 (full intensity). Thus, each color in the real-world has a representative point that is on the surface or inside this cube.

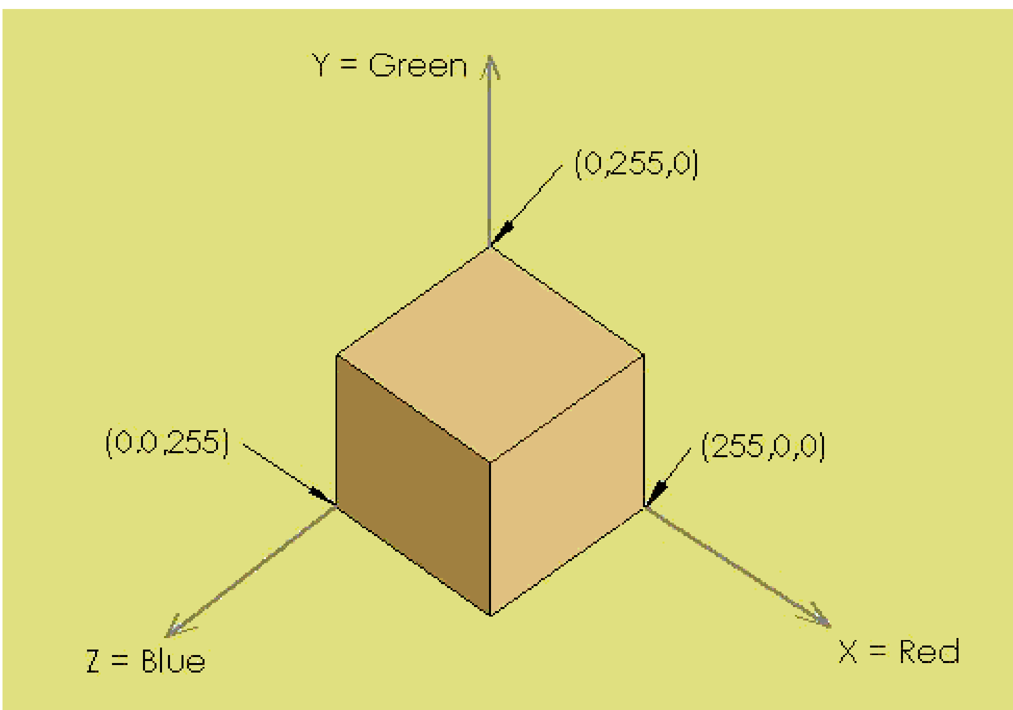


Figure 4.1 Cube of Color (three dimensions representing three basic colors).

4.6.2 Step Two: Preparation of Samples of Pictures with Different Colors

In the second step, using the prepared color code system, a collection of samples was prepared to be used in the third step conducting an experiment to solicit the human subject perception of the "weight" of the color samples. In this connection a reasonable number of pictures with different foreground and background colors was needed to be determined.

At the beginning, it was decided to have three levels of each basic color (0% that is lack of color or black color, 50% or dark level of the color, and 100% or light level of the color) in combination with each other. In that case, 3^3 or 27 colors were produced from a combination of 3 basic colors at 3 levels of percentages. From 27 colors, if one color is used in the foreground and one in the background, the total number of different foreground and background colors would be the combination of 2 on 27 (i.e., $\frac{27 \times 26}{2}$ or 351). Substituting the foreground and background colors to each other would add another 351 cases. Therefore totally, there would be (351 + 351) or 702 modeled colors to show each participant. Showing this huge number of pictures to participants for evaluation was not practically possible. Therefore, a decision was made to use two levels of each basic color (0%, or lacking, and 100%, or existing) in combination with each other. In that case 2^3 or 8 colors were produced from combination of 3 basic colors at 2 levels of percentages. Figure 4.2 shows these 8 colors: Black, Red, Green, Blue, Yellow (mixture of Red and Green), Magenta (mixture of Red and Blue), Cyan (mixture of Green and Blue), and White. R, G, and B in Figure 4.2 stand for red, green, and blue, respectively, and indicate the colors that have been used in each circle. For example, yellow color is identified with RG, which stands for red and green colors. It is further noted that these eight colors occupy, respectively, eight corners of the color space (Figure 4.1).

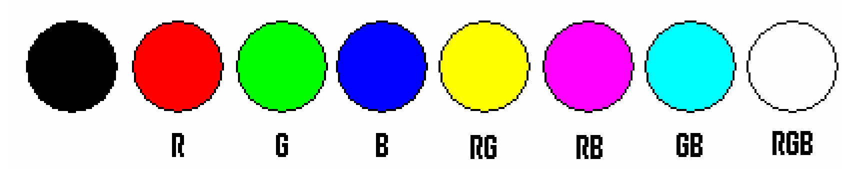


Figure 4.2 Colors produced from combination of 3 basic colors at 2 levels (0% and 100%)

From 8 colors, if one is used in the foreground and one in the background, the total number of different foreground and background colors would be the combination of 2 from 8 (i.e., $\frac{8 \times 7}{2}$ or 28). Substituting the foreground and background colors to each other would add another 28 cases. Therefore, in total there would be (28 + 28) or 56 pictures. The 56 pictures are a practically reasonable number of pictures to be shown to participants in an experiment for evaluating the weights of colors. It is observable that all 56 different foreground and background colors were selected from corners in the cube of color, or in other words, from the boundary of the color space (Figure 4.1). In addition to these 56 pictures from the colors on the edge of the cube of color, a decision was made to use a quarter of 56 (or 14) randomly prepared foreground and background colors from inside of the cube of color (in Figure 4.1). Therefore, finally based on (56 + 14) or 70 proposed combination of foreground and background colors, 70 pictures were modeled. Figure 4.3 shows one instances of the 70 modeled pictures.

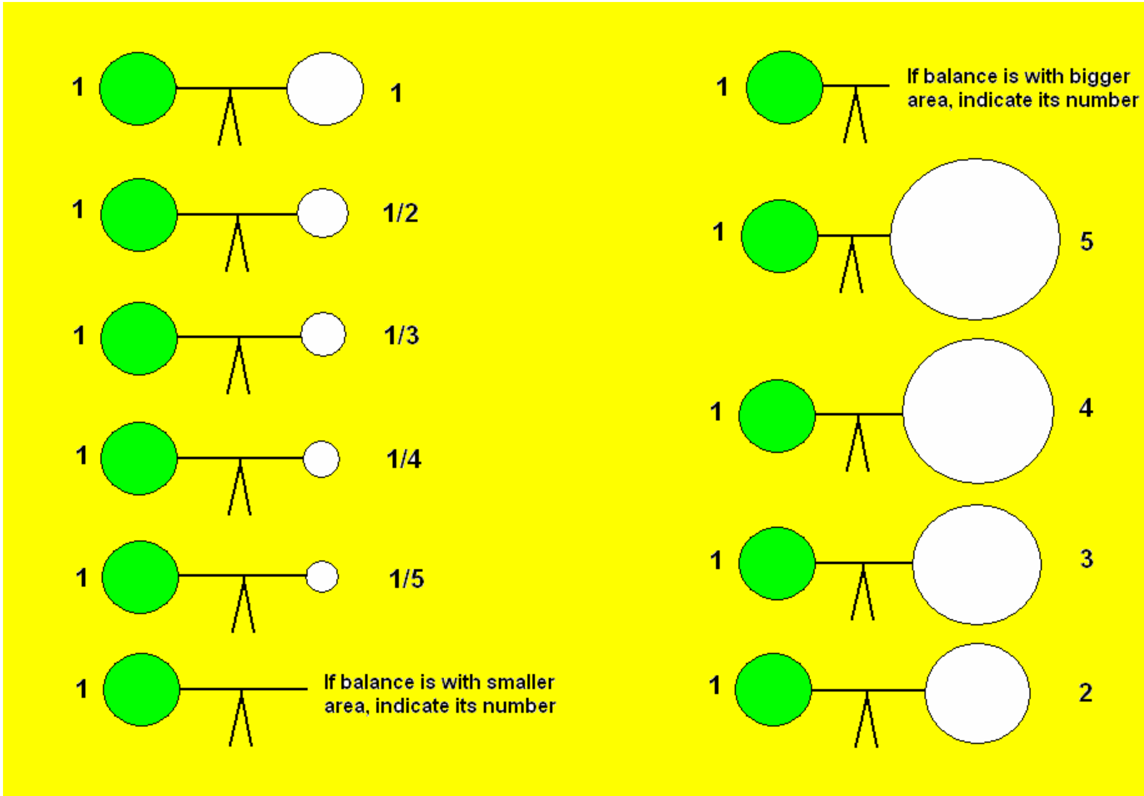


Figure 4.3 One instance from 70 modeled foreground and background colors

On all 70 modeled pictures there were some seesaw scales with two circles at both ends. The area of the left side circles were set to unit area (cm^2). The area of the right side circles varied to the smaller and larger areas than unit. The value beside each right circle represents the area of the right side circle in cm^2 and also is a ratio that the circle has respect to the unit area. The color of the right side circles in the all pictures was white. For the left side circles and background, the 70 proposed combinations of foreground and background colors were used.

4.6.3 Step Three: Statistic Design of an Experiment

In the third step, the 70 modeled pictures were shown to a group of participants twice. The factor of this experiment is “Picture with different Foreground and Background Color” with 70 levels.

4.6.3.1 Participants

One question was to determine how many participants to be recruited for the experiment. As Lin [2003] mentioned, in human factor-related experiments it is hard to determine the sample size according to the standard rules in statistics [Montgomery, 2001]. This may be seen from the published human factor research. As an example, Kotval [1998] did a survey for determining sample sizes of 118 published studies in the area of eye movement in human factor fields of study and found that 13 subjects was the best number. Two instances of slightly larger sample size were Bauerly [2003], who used 16 subjects for research on computer interfaces, and Lin [2003], who conducted a study regarding interface design with 20 subjects.

Therefore, in this experiment 20 students (10 male and 10 female between 20 and 30 years old and with an average age of 26 years old) were asked to do a computer survey and assigned a value to each prepared picture. Participants were selected using a convenience sampling method. It is further noted that two replications of each evaluation were done in order to increase the reliability of the evaluation.

4.6.3.2 Procedure

70(2) or 140 pictures were shown to each participant. Each participant was asked to look at each picture and evaluate that a seesaw scale with what a ratio size of the right circle would give him/her the most feeling of balance or stability. The purpose of this question was to determine what amount of the area of a foreground color (color of the right circle) was perceived equal to one cm^2 of white color in the context of a background color. The experiment length was about 45 minutes and to appreciate participants' time, a $10 University of Saskatchewan bookstore gift card was given to each of them.

4.6.4 Step Four: Training and Testing a Neural Network Model

Finally in the fourth step, an attempt was made to find a model that would be able to give a weight for each new color after training with color codes and their weights from the third step. For this purpose a neural network model in MatLab 7 was used. The neural network algorithm which was used called, "multi-layer feed-forward neural networks trained by a back propagation algorithm". Back propagation networks are very popular artificial neural networks and are used more than other networks [Lek and Guegan, 1999]. Sigmoid functions were used for the input layer along with two hidden layers, and for the output layer linear functions were considered. Six nodes were used at the input and hidden layers and one node at the output layer.

Data from the experiment was used to train the network model. Six codes that indicated specific ratios of foreground and background color were used as inputs of the network model, and participants' evaluations of the weight of the foreground color were used as the output, of the model. Therefore, as Figure 4.4 shows, the neural network model had six inputs and one output.

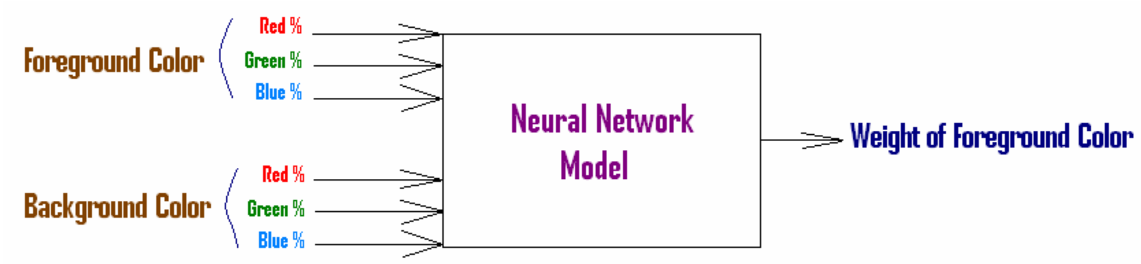


Figure 4.4 Neural Network Schema for measuring color's weight

An average of duplicate data that had been obtained from 20 participants regarding the 70 pictures was taken. Therefore, the total number of data that entered in the network model as input of the model was 1176000 values {20 (number of participants) × 2 (number of replications) × 70 (number of pictures presented to each participant) × 6 (number of codes for two basic colors used as foreground and background)}.

After training, a network model should be tested. There are two main approaches for testing neural network models performance. The first approach is to use an independent data set, called testing data, which compares predicted data with observed data. The second approach use cross-validation, leave-one-out, jackknife, or bootstrapping techniques for testing models performance [Guisan et al., 2000]. The second approach is used when all the data is necessary to train the model and the available data set is small [Spitz and Lek, 1999; Stacy et al., 2006]. In an attempt to separate 70 data elements to training and testing data, it was figured out that all the data elements were needed for training to improve the accuracy of the model. A term was defined in this study, named accuracy error, as an index in the prediction of network models. Proper accuracy error is met when the average accuracy error of the model for all testing data would be less than 0.05. Accuracy error is defined by Equation 4.7.

$$Accuracy \quad error = \frac{(O - \Pr)}{O} \qquad \textbf{(4.7)}$$

In the Equation 4.7, "*O*" stands for Data observed by the survey, and "Pr" stands for Data predicted by the model. In order to decrease the model accuracy error one of the techniques from the second approach was used, called leave-one-out cross validation technique which uses all of the available data elements (except one data element) as training data. Before further explaining regarding leave-one-out cross validation technique, in the following it is explained the reason of reaching to the conclusion that it was necessary to use the most data elements as training data in order to improve the accuracy error of the *ANN* model.

On the first trial, 70 data elements regarding the 70 pictures were separated into two categories: 56 data elements regarding border points in the cube of color, and 14 data elements randomly selected from inside of the cube of color. The network was trained 10 times with 56 data elements and then simulated the network with 14 testing data elements. The purposes were to get to the best accuracy error in network model prediction while network model performance maintain less than 0.05. The best accuracy error achieved by the model after the 10 training runs, was 0.26. Therefore, it was decided to increase the number of training data, which meant decreasing the number of testing data. On the second trial, 70 data elements were separated into 60 training and 10 testing data. Again the network was trained 10 times. In this case, the best accuracy error obtained was 0.19. On the third trial, 70 data elements were separated into 63 training and 7 testing data and the network again was trained 10 times. In this case the best accuracy error obtained was 0.13. The trend line of decreasing accuracy error showed that all the data are needed to train the network model. It was concluded that having data for training the network from inside of the color's cube, rather than just from the boundary of the color's cube, was a necessity for improving the accuracy error of the network. Based on this conclusion it was decided to use the leave-out cross validation technique.

The leave-one-out cross validation technique should be used when every observation is unique and can be added to the training data. In this technique, only one data need to be retained for testing and all others should be used for training the network [Spitz and Lek, 1999]. Then the testing parameter for the network, here accuracy, should be tested with the one data element left out. Further, another data need to be retained for testing, all others should be used for training the network, and accuracy should be tested with the one data element left out. This process need to be continued until all data once retains for testing. As a result, for a network with "n" data element, "n" accuracy errors should be determined. Final accuracy error of the model would be the average of all "n" accuracy errors. Therefore, ideally, this study needed to have 70 set of 69-1 data and for each one the network should be trained and simulated several times in order to get reasonable performance from the network (less than 0.05) as well as proper accuracy error (less than 0.05).

In the first data assignment, 70 data were separated to a set of 69-1 training-testing data. 70^{th} data were left out, the network was trained with other 69 data, and the accuracy error was calculated. Network performance was less than 0.05; but accuracy error was not less than 0.05. The network was run for the first set of data with other initial weights. Again accuracy error was poor. This procedure was continued until the 9th run upon which 0.03 network performance and 0.03 accuracy error were reached.

In the second data assignment, another element of data (69th data) was left out as testing data and the network model was trained with the other 69 data (data from 1-68 and 70), and accuracy error was calculated. After the 6th run of this set of data 0.006 network performance and 0.025 accuracy error were reached.

In the third data assignment, another element of the data was set out as testing data (68th data) and the network model was trained with the other 69 data (data from 1-67 and 69 and 70), and accuracy error was calculated. The 9th run of this set of data yielded 0.048 network performance and 0.03 accuracy error.

Just for the first three sets of 69-1 data, the network model was run 24 times and each length of each run was around 5000 epochs. Therefore, it was figured out that the process would be very time consuming if the network wanted to be performed for the remaining 67 sets of 69-1 data considering each set of data needed to run several times (average for three set of 69-1 training testing data was 8 times running).

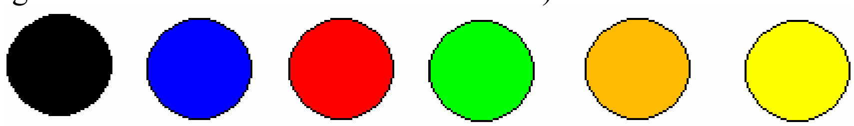
Therefore, although it was supposed to continue this task to its completion, running the network an average of 8 times for all 70 sets of 69 training and one testing data combinations (altogether 560 times), instead it was decided to stop the procedure and to generalize the information it was obtained from 24 runs done of the first three set of 69-1 data. It was supposed that for each other set of 69-1 data, a performance and accuracy error less than 0.05 would be arrived, although number necessary for network training for some sets might be high.

Based on this assumption, among the three obtained acceptable networks in each of the three sets of 69-1 training testing data, the one with best accuracy error (least value that is 0.025) and also proper performance (0.006) was selected as the final network model. This selected network model was used for the next experimental work of this study, which was to determine the weight of a new color when compared to a white color based on area ratio. Input and output of training and testing data of the selected network model are in Appendix B. Additionally, the following characteristics of the selected network model are listed in Appendix C: number of layers, number of neurons of each layer, transfer function names of each layer, initial weights, number of epochs, network performance, survey values for simulated data, network values for simulated data, and accuracy error.

4.7 Comparison of the Network Model Prediction with other Studies

The averages of participants' evaluations obtained from the survey regarding the weight of the five following colors with white backgrounds were as follows: black= 3.48, blue= 3.19, red= 3.18, green= 2.47 and yellow= 2.15. Values which the network model suggested for the five colors and the color orange were black= 3.42, blue= 3.19, red= 2.71, green= 2.47, orange= 2.24, and yellow= 2.15. It should be noted that orange color code had not been presented to the network model as training data and its value was predicted by the network model. With the exception of red and blue, these weight rankings of colors from participants and network model are conformable with intuition and also close to the results of studies done by Locher et al. [2005] and Pinkerton and Humphrey [1974]. Comparison of the rankings of this study with the rankings of Locher's and Pinkerton's studies is shown in Table 4.1. It should be mentioned that Pinkerton's ranking in Table 4.1 is based on the average rank which participants attributed to each color; however, their Wilcoxon matched pairs test showed that yellow was significantly lighter than all other colors and red was significantly heavier than green, orange, and yellow. Therefore, in Pinkerton's study, although red was somewhat heavier than blue, their difference was not significant.

Table 4.1 Comparison of Weight Ranking of Colors among the model with others' (*: Based on average of participants evaluations obtained in the survey; **: Based on output of the neural network model; ***: According to the study of Pinkerton and Humphrey [1974]; ****: According to the study of Locher et al. [2005]; -: Research related to the row did not study weight of the color mentioned on the column)



Colors		Black	Blue	Red	Green	Orange	Yellow
RGB values		(0,0,0)	(255,0,0)	(0,0,255)	(0,255,0)	(255,187,0)	(255,255,0)
Weight Ranking of Colors	Participants*	1	2	3	4	-	5
	Network**	1	2	3	4	5	6
	Pinkerton***	-	2	1	3	4	5
	Locher****	-	2	1	-	-	3

4.8 Conclusion

A network model was obtained with the performance and accuracy error necessary to reliably measure the weight of any color with consideration of its background color. The model was tested for the weight of five colors and found that the hierarchy of color weights proposed by the network model reflects answers from a group of surveyed participants as well as results of other studies. It was concluded that the neural network model is valid and comprehensive in quantifying the weight of colors and the properties of balance in product pictures.

A method was developed for integrating the measured weight of colors with the weight of areas to quantify visual balance. In the method, color's weight is transformed into an equivalent area; therefore, it easily can be added with the weight resulted from area. Therefore, the balance of a picture with respect to color and area of its components can be measured.

APPENDIX A. DISASSOCIATION BETWEEN DESIGN ASPECTS OF VARIABLES

Note: The following tables indicate the percentage of relevancy between design attributes (aesthetics with function and aesthetics with ergonomics) of 24 cell phone design variables.

First trial, Aesthetics-Function disassociation:

Rank	Variable	Function-Beauty	Disassociation %	Description
1	19	2.7	89.7	19. "Decorated area"
2	24	2.7	89.7	24. "Color of body"
3	22	2.7	89.7	22. "Frontal shape of body"
4	12	2.2	74.8	12. "Ratio of body width to height"
5	17	2.2	74.8	17. "Number of colors used in body"
6	18	2.2	74.8	18. "Existence of colorful area"
7	15	2.2	74.7	15. "Degree of body luster"
8	16	2.2	74.7	16. "Brightness of body color"
9	5	2.1	70.4	5. "Color of mainly used button"
10	23	2.1	70.4	23. "Degree of body roundness"
11	3	2.0	66.7	3. "Shape of most salient button"
12	13	2.0	66.7	13. "Horizontal length of body"
13	1	1.9	64.0	1. "Number of differently shaped buttons"
14	9	1.9	62.1	9. "Ratio of display width to height"
15	6	1.7	58.1	6. "Degree of button size variation"
16	14	1.7	56.3	14. "Vertical length of body"
17	2	1.6	52.5	2. "Shape of mainly used button"

18	10	1.3	43.7	10. "Area of display"
19	21	1.3	43.7	21. "Layout of components"
20	11	0.9	30.7	11. "Degree of emphasis on speaker design"
21	4	0.4	15.0	4. "Size of mainly used button"
22	8	0.3	10.4	8. "Existence of menu navigation buttons"
23	7	0.3	8.4	7. "Number of exposed frontal buttons"
24	20	0.3	8.3	20. "Open mechanism"

First trial, Aesthetics-Ergonomics disassociation:

Rank	Variable	Ergonomics-Beauty	Disassociation %	Description
1	24	2.3	75.0	24. "Color of body"
2	19	2.0	66.9	19. "Decorated area"
3	15	1.6	51.7	15. "Degree of body luster"
4	8	1.5	51.6	8. "Existence of menu navigation buttons"
5	18	1.1	35.1	18. "Existence of colorful area"
6	17	1.0	33.5	17. "Number of colors used in body"
7	4	1.0	33.2	4. "Size of mainly used button"
8	22	1.0	32.4	22. "Frontal shape of body"
9	16	0.8	26.7	16. "Brightness of body color"
10	9	0.7	22.4	9. "Ratio of display width to height"
11	20	0.7	22.1	20. "Open mechanism"
12	11	0.6	19.1	11. "Degree of emphasis on speaker design"
13	21	0.6	18.5	21. "Layout of components"

14	14	0.4	14.1	14. "Vertical length of body"
15	5	0.4	14.1	5. "Color of mainly used button"
16	6	0.3	11.2	6. "Degree of button size variation"
17	10	0.3	10.4	10. "Area of display"
18	2	0.3	9.3	2. "Shape of mainly used button"
19	13	0.2	8.2	13. "Horizontal length of body"
20	7	0.2	6.0	7. "Number of exposed frontal buttons"
21	1	0.2	5.9	1. "Number of differently shaped buttons"
22	3	0.2	5.9	3. "Shape of most salient button"
23	23	0.1	4.4	23. "Degree of body roundness"
24	12	0.1	1.8	12. "Ratio of body width to height"

Second trial, Aesthetics-Function disassociation:

Rank	Variable	Function-Beauty	Disassociation %	Description
1	19	3.0	100.0	19. "Decorated area"
2	24	3.0	100.0	24. "Color of body"
3	22	2.5	85.0	22. "Frontal shape of body"
4	15	2.5	85.0	16. "Brightness of body color"
5	5	2.5	84.8	5. "Color of mainly used button"
6	14	2.5	84.8	12. "Ratio of body width to height"
7	17	2.5	84.8	17. "Number of colors used in body"
8	18	2.5	84.8	18. "Existence of colorful area"
9	11	2.5	84.8	23. "Degree of body roundness"
10	9	2.4	80.7	9. "Ratio of display width to height"
11	13	2.2	72.1	13. "Horizontal length of body"

12	12	2.2	72.1	14. "Vertical length of body"
13	16	2.1	69.9	15. "Degree of body luster"
14	3	2.0	66.6	3. "Shape of most salient button"
15	1	2.0	66.4	1. "Number of differently shaped buttons"
16	2	1.8	61.1	2. "Shape of mainly used button"
17	6	1.8	61.1	6. "Degree of button size variation"
18	21	1.8	60.2	21. "Layout of components"
19	10	1.7	56.4	10. "Area of display"
20	7	1.2	39.7	7. "Number of exposed frontal buttons"
21	4	1.1	37.6	4. "Size of mainly used button"
22	20	0.5	16.6	20. "Open mechanism"
23	23	0.5	16.1	11. "Degree of emphasis on speaker design"
24	8	0.0	1.5	8. "Existence of menu navigation buttons"

Second trial, Aesthetics-Ergonomics disassociation:

Rank	Variable	Ergonomics-Beauty	Disassociation %	Description
1	24	2.5	83.3	24. "Color of body"
2	19	2.1	70.0	19. "Decorated area"
3	15	2.0	68.3	16. "Brightness of body color"
4	8	2.0	66.7	8. "Existence of menu navigation buttons"
5	18	1.5	51.6	18. "Existence of colorful area"
6	17	1.2	40.8	17. "Number of colors used in body"

7	5	1.0	34.8	5. "Color of mainly used button"
8	7	1.0	34.8	7. "Number of exposed frontal buttons"
9	4	1.0	33.3	4. "Size of mainly used button"
10	16	0.8	28.2	15. "Degree of body luster"
11	22	0.8	26.6	22. "Frontal shape of body"
12	11	0.7	23.7	23. "Degree of body roundness"
13	21	0.7	23.1	21. "Layout of components"
14	14	0.5	15.4	12. "Ratio of body width to height"
15	1	0.4	14.8	1. "Number of differently shaped buttons"
16	9	0.4	14.3	9. "Ratio of display width to height"
17	2	0.4	13.7	2. "Shape of mainly used button"
18	20	0.3	11.0	20. "Open mechanism"
19	6	0.3	9.6	6. "Degree of button size variation"
20	10	0.2	8.2	10. "Area of display"
21	13	0.1	3.8	13. "Horizontal length of body"
22	12	0.1	3.8	14. "Vertical length of body"
23	23	0.1	3.7	11. "Degree of emphasis on speaker design"
24	3	0.0	0.1	3. "Shape of most salient button"

APPENDIX B. INPUT AND OUTPUT OF TRAINING AND TESTING DATA

Note: The following table indicates the input (i.e., the codes connected to foreground and background colors) and the output (i.e., weight of foreground color) of training and testing data of the selected network model. The model enable user to measure weight of a new color considering its background color.

Training Data #	Input						Output
1	255	0	255	255	0	0	2.16125
2	255	255	255	255	0	255	1.19575
3	255	255	255	0	0	0	1.2375
4	255	255	255	0	255	0	1.3425
5	0	0	0	255	0	255	2.94375
6	0	0	255	0	0	0	2.04375
7	0	0	0	0	255	255	2.8
8	0	0	255	0	255	255	2.7875
9	255	255	0	0	255	255	2.04375
10	255	0	0	255	255	255	2.23825
11	255	255	255	255	0	0	1.37075
12	255	255	0	0	255	0	2.14025
13	255	0	255	0	0	0	2.555
14	0	255	255	0	255	0	2.016425
15	255	0	0	255	255	0	2.7375
16	0	255	255	255	255	255	2.4
17	255	255	255	255	255	0	1.4
18	255	0	0	0	0	255	2.7175
19	0	0	255	0	255	0	2.73325

20	255	0	255	0	0	255	2.3665
21	255	0	0	255	0	255	2.5925
22	255	0	255	0	255	0	2.83125
23	255	0	255	255	255	0	2.3125
24	255	0	0	0	0	0	2.76225
25	0	0	255	255	255	0	2.7875
26	0	255	255	255	0	0	2.05275
27	0	255	255	255	255	0	1.9915
28	255	255	0	255	0	255	2.1125
29	0	255	0	0	255	255	2.37075
30	0	0	0	0	255	0	3.61875
31	0	0	255	255	0	255	2.577
32	0	0	255	255	255	255	3.1915
33	0	255	0	255	255	0	2.379
34	0	255	255	0	0	0	1.86725
35	255	255	0	255	0	0	2.255
36	255	0	255	0	255	255	2.64575
37	0	255	0	0	0	0	1.9415
38	0	255	0	255	0	255	2.11875
39	0	0	0	255	255	255	3.4175
40	0	255	0	255	0	0	2.54375
41	0	0	255	255	0	0	2.543675
42	0	255	0	255	255	255	2.471425
43	0	255	255	0	0	255	2.288
44	255	255	0	0	0	255	2.54375
45	255	0	0	255	255	255	3.175
46	0	0	0	255	255	0	3.375

47	255	0	0	0	255	0	2.82075
48	255	0	0	0	255	255	2.925
49	0	0	0	255	0	0	3.1925
50	255	255	255	0	0	255	1.4
51	0	255	255	255	0	255	2.10625
52	255	255	0	255	255	255	2.15
53	255	255	0	0	0	0	2.17375
54	0	0	0	0	0	255	3.21525
55	255	255	255	0	255	255	1.1125
56	0	255	0	0	0	255	2.3645
57	115	10	238	33	167	21	2.331175
58	120	62	211	4	178	111	2.457
59	169	84	81	42	78	218	2.066575
60	222	177	92	244	131	165	2.226875
61	70	2	36	206	133	9	2.706175
62	141	182	79	64	210	185	2.07075
63	16	204	6	168	216	108	2.005
64	106	61	132	30	56	164	1.751625
65	87	153	94	150	155	137	2.02575
66	242	147	1	83	90	128	2.2665
67	60	220	193	86	255	97	2.092625
68	249	17	32	226	52	47	2.007675
70	3	185	55	130	231	170	2.055
Testing Data #	Input						Output
69	159	233	76	110	122	71	1.66325

APPENDIX C. CHARACTERISTICS OF THE NETWORK MODEL

Note: The following information indicates the characteristics of the selected network model. The model enable user to measure weight of a new color considering its background color.

Number of layers= 4

Number of neurons in input layer= 6

Number of neurons in the first hidden layer= 6

Number of neurons in the second hidden layer= 6

Number of neuron in the third hidden layer= 6

Number of neuron in output layer= 1

Transfer function of the first hidden layer= TANSIG

Transfer function of the second hidden layer= TANSIG

Transfer function of the third hidden layer= TANSIG

Transfer function of output layer= PURELIN

Initial Weights (1, 1)= [-0.001936 -0.0017266 -0.00064767 -0.013644 0.004449 0.0024488;

-0.0040904 0.0070168 0.01101 -0.00031899 0.0015415 -0.0054249;

-0.0034381 -0.010491 -0.0045279 -0.00011293 0.003208 0.0081492;

0.0068994 -0.00776 -0.004415 0.0061267 -0.0070876 -0.0020017;

0.0053548 -0.002559 0.0017398 0.0062758 -0.0070031 -0.0096129;

0.010258 -0.0033809 0.00063362 0.009564 -0.0023082 0.0022855]

Initial Weights (2, 1)= [0.15002 -1.1549 -1.1109 0.36296 0.11694 -0.90867;

-0.94129 0.33387 -0.39534 0.93841 -0.67822 -1.033;

-1.1701 0.13145 -0.71155 -0.18427 -0.36641 -1.225;

0.79505 -0.7417 0.77971 0.88297 -0.79886 0.59461;

-0.72466 0.41258 0.39001 -1.2168 0.13495 -1.1024;

-0.79284 -0.1529 1.0139 0.073654 0.90151 1.0312]

Initial Weights (3, 2)= [0.041057 -0.27472 -0.93891 -0.35573 1.3033 -0.88193;

1.4074 -0.23982 0.7548 0.88878 -0.061294 -0.39974;

-1.0632 -0.34471 -0.35281 0.82709 -0.36888 1.1695;

-1.2624 1.1384 -0.12789 -0.32721 0.73938 0.04448;

-0.31977 0.45401 -1.2312 -0.65959 -0.32778 -1.0931;

-0.24133 1.278 0.17277 -0.43466 1.1078 0.65117]

Initial Weights (4, 3)= [-0.99058 0.20625 0.91373 -0.20514 0.4631 0.36928]

Number of epochs= 5824

Network performance= 0.0063587

Survey value for simulated data= 1.66325

Network value for simulated data= 1.6206

Accuracy error= 0.025643