https://doi.org/10.2352/issn.2169-2629.2019.27.28

Estimating concentrations of pigments using encoder-decoder type of neural network.

Kensuke Fukumoto[1), 2)], Norimichi Tsumura[1)], Roy Berns[2)]
1) Graduate School of Science and Engineering, Chiba University, Chiba, Japan
2) Munsell Color Science Laboratory, Program of Color Science, Rochester Institute of Technology, Rochester, New York, USA

Abstract

In this paper, we propose a method to estimate the concentration of pigments mixed in a painting, using the encoder-decoder model of neural networks. Encoder-decoder model is trained to output value which is same as input and its middle output extracts a certain feature as compressed information of the input. In this instance, the input and the output are spectral data of a painting. We trained the model to have pigments concentration as compressed information as a middle output. We used the dataset which was obtained from 19 pigments. The dataset has scattering coefficient and absorption coefficient of each pigment. We applied Kubelka-Munk theory to the coefficients to obtain many patterns of spectral data. It's shown that the accuracy of estimation is very high, and the speed of execution is very fast compared with a conventional method using simple for-loop optimization. We concluded our method is more effective and practical.

1. Introduction

Fine art such as paintings is a valuable asset to humanity. It is not uncommon that it has passed more than hundreds of years since those paintings were created. The color appearance has changed with the passage of time. Besides, most of the paintings are displayed in a place like a museum and exposed to the air and the light, which will change the color of the paintings. Imaging simulation using concentration mapping of pigments is one of the ways to view how the painting looked like before [1, 2, 3]. It is also beneficial that concentration mapping of pigment can be used for retouching the painting with damage because of the passage of time. To estimate the pigment concentrations, it is needed to calculate a value at which the error is minimum between the captured value and the estimated value based on the physical model. The error minimization process involves iterative calculations on a computer and has the disadvantage that it takes time depending on the computer to apply it to the entire spots on a painting.

In this study, we propose a fast and efficient method for estimating pigment concentration using the encoder-decoder model of neural network.

2. Related Work

By applying Kubelka-Munk theory, the type of pigment and its concentration can be estimated from the reflectance of the pigment [4]. Kubelka and Munk showed that in the opaque pigment model shown in Figure 1, the reflectance R of the pigment will be represented using the ratio of the absorption coefficient K to the scattering coefficient S when the pigment thickness X is infinite, which is described in Equation 1. By using the formula of Saunderson shown in Equation 3, the internal reflectance obtained by Kubelka-Munk theory is converted to measured reflectance.

$$R_{\lambda,i} = 1 + \left(\frac{K}{S}\right)_\lambda - \sqrt{\left(\frac{K}{S}\right)_\lambda^2 + 2\left(\frac{K}{S}\right)_\lambda} \quad (1)$$

$$R_{\lambda,m} = \frac{(1-K_1)(1-K_2)R_{\lambda,i}}{1-K_2R_{r,i}} + K_{ins}K_1 \quad (2)$$

This theory can be applied when the reflectance of pigments is not dependent on the increase in pigment thickness. In addition, the ratio of the absorption coefficient to the scattering coefficient of the pigment made by mixing several kinds of pigments can be determined from the absorption coefficient of the original pigment, the scattering coefficient, and each concentration, which is showed in Equation 3 where c represents the concentration for the ith pigment.

$$\left(\frac{K}{S}\right) = \frac{\sum_i c_i K_i + (1-\sum_i c_i) K_{white}}{\sum_i c_i S_i + (1-\sum_i c_i) S_{white}} \quad (3)$$

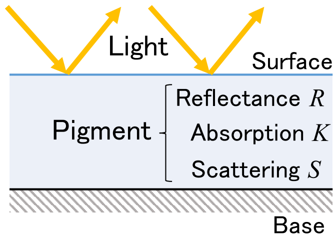


Figure 1. *The model of pigment*

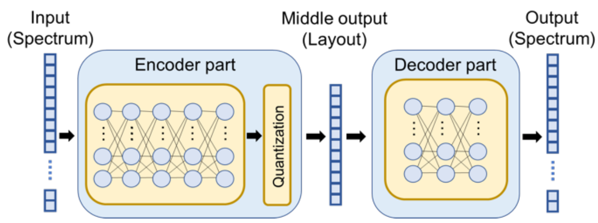


Figure 2. *Illustration of the network structure* [6]

Conversely, if the absorption coefficient and the scattering coefficient of each pigment used to mix are obtained in advance, it's possible to estimate their concentrations. The actual procedure is to find the concentration of each pigment which minimizes the difference between the measured value and the theoretical value of the reflectance calculated by Kubelka-Munk theory. In this way, the concentration of each pigment can be estimated. However, the number of combinations of pigments and their concentrations is enormous, and this procedure has the disadvantage that it takes a considerable amount of time to apply it to the spots on the entire painting.

We used their theory to build a concentration database used to train our neural network. At this time, we assumed the pigments to be opaque, so we used opaque model of Kubelka-Munk theory.

Shi et al. proposed the method which applies an encoder-decoder model of a neural network [5] for estimating the ink layout in color patches printed by a 3D printer [6]. Furthermore, they demonstrated the reproduction of a real painting by using their method.

In this method, spectral reflectance obtained by measuring a color patch is used as an input and they estimate the color and thickness of the layers constituting the color patch. They prepared 20,878 valid layouts and used 18,878 out of them as training data.

The structure of the neural network model used in their method is shown in Figure 2. In their model, the input and output are 31-dimensional vectors, which represents the spectral reflectance in the wavelength of visible light, which is from 420 to 720nm. The intermediate output, which is a compressed representation of the input, is an 11-dimensional vector, which represents the layer structure of the ink. In their model, the encoder has 8 layers and 150 neurons in each layer, on the other hand, the decoder has 8 layers and 160 neurons in each layer. The encoder is designed to have deeper structure, which allows the model to estimate more complicated distribution. Basically, encoder-decoder model has no training data and loss function for the intermediate output, so they extract some compressed features automatically. In their method, they prepared the training data for the intermediate output and set the loss function so that the model can extract the desired feature through the network, which is the layout in this case.

In general, deep learning takes time to train, but once learning is performed, there is an advantage that estimation can be performed at high speed. In this study, we will construct a high-speed pigment concentration estimation method referring to their method.

The difference between Shi et al. and ours is the target to estimate. Their application is reproducing paintings using 3D printer and They estimates the ink layer structure. On the other hand, our target is concentration of pigment and we try to apply proposed method to pigment mapping and image simulation on a painting.

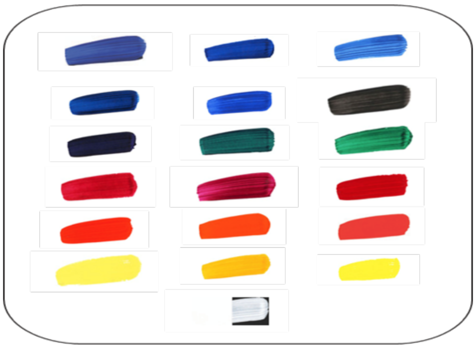


Figure 3. *Nineteen color pigments selected from Golden's Heavy body*

3. Training detail for estimating pigments concentration

In this section, we will explain the detail about how we trained the neural network. First, we describe what dataset we used and how we prepared it. The model structure of neural network is showed in next subsection. Finally, the training protocol is explained.

3.1 Pigment dataset used in training

We will explain the data set used in this study. We used the colorant measurements dataset measured by Roy Berns et al[4]. They picked nineteen color pigments from Golden's Heavy body as showed in Figure 3. They measured absorption coefficient and the scattering coefficient of each pigment. The measurement was performed at 10 nm intervals from 350 nm to 780 nm. We applied Kubelka-Munk's theory to the measured values to determine the reflectance. At this time, the number of pigments to be mixed was up to four colors including white, and the ratio of mixing was in 5% steps. We prepared two datasets. One is 1771 data made from only one combination of pigments, which are diarylide yellow, quinacridone red, cobalt blue and white. This dataset was used as a trial to see how this method works. The other one is 1445136 data, which was considered the combinations of 3 colors out of 18 in addition to white. In both cases, white pigment was added to the combinations. Mixing procedure causes the mixed pigment to be darken but by adding white it let the dataset have the variety range of reflectance.

3.2 The structure of encoder-decoder model

The structure of the model is basically same as that of Shi et al. In their method, the compressed output, which is the intermediate output of the auto encoder, is 11 dimensions while it is 19 dimensions in our case, so we increased the number of neurons. Also, since their intermediate outputs are discrete values, the quantization layer was added before intermediate output. In our case, intermediate output is the concentration of the pigment, which is continuous values, so we removed the quantization layer. We note the detail about the model that we used. The hidden layer of the encoder part has eight layers of 300 neurons. Although the rectified liner unit (ReLU) was used as the activation function of each layer, only the output of the encoder unit was set to a soft max function. Soft max function returns certain possibility whose sum is one. The target we try to estimate is the concentration of the pigment and we can think the possibility to be the concentration. We used an absolute error average between the ground truth and the estimated value as the loss function in encoder part. In the decoder part, we set four layers of 500 neurons. The activation function was also ReLU for all layers.

The mean squared error was used for the loss function in decoder part.

3.3 Training protocol for neural network

We randomly divided the dataset prepared in subsection 3.1 with 80% of the entire dataset as the training data, and 20% as the test data. At this time, the dataset was divided not to be biased in the combination of pigments. We used Adam's optimization algorithm for learning [5]. We set the hyper parameters as follows. The initial learning rate was 0.001. This value means the proportion in which the weights in each neuron are updated. Beta 1 was 0.9. This is the exponential decay rate for the first moment estimates. Beta 2 was 0.999. This is the exponential decay rate for the second moment estimates. The number of epochs was 10, and the batch size was 32. The learning rate was multiplied by 0.1 every 2 epochs. Weights are regularized by penalizing their L^2 norm weighted by 10^{-5}. The learning was first performed only in the decoder unit, and then the weight of the decoder was fixed, and learning was performed on the entire model. At this time, we imposed 7 to 3 weights for the two loss functions in encoder unit and the decoder unit respectively.

4. Result of training and estimating

The section 4.1 shows the trial result, which was obtained by using only one combination to train the model. The section 4.2 shows the main result, which was performed by applying the whole dataset we prepared.

4.1 Trial Result

First, learning and estimation were performed using a data set with only one combination of color. We calculated the average of the mean square error of concentrations between each estimation result and the correct value. The distribution of error rate between GT and estimated is shown in Figure 4. The histogram was normalized. You can see that most result are distributed to low error. Each statistical value regarding the error of the estimation result has an average value of 0.0002, a median and a standard deviation of 0.0001. The results with error rates of best and worst accuracy are shown in Figure 5. The vertical axis represents the error rate, and the horizontal axis represents its frequency. The red bin is the correct value and the blue bin is the estimated value. It can be seen in Figure 5 that even in the results with the worst estimation accuracy, the network successfully estimated the value close to the correct value. Considering Figure 4 and 5, it can be understood that the estimation with high accuracy is performed as a whole.

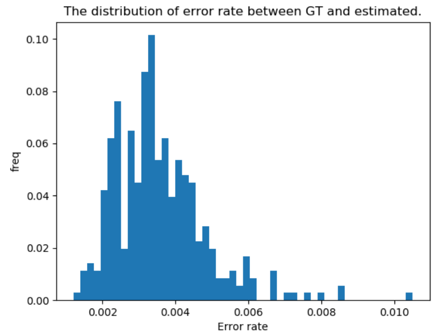
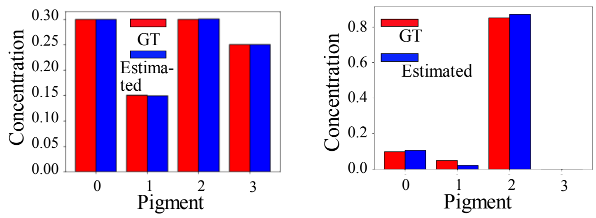


Figure 4. *The distribution of error rate between GT and estimated.*



(a) Best **(b) Worst**

Figure 5. *The results with error rates of best and worst accuracy.*

4.2 Main Result

We trained the model with main data and estimated the pigments concentration. The mean square error was calculated as in the previous section, and the distribution of error rate between GT and estimated is shown in Figure 6. Comparing Figure 4 and 6, the error rate of main result is higher than trial result entirely. The statistics of the error rates are as follows. Average value is 0.0041, median is 0.0017, and standard deviation is 0.0064. The results with the highest and lowest error rates are shown in Figures 7. Figure 7(a) shows that the accuracy of the estimation result is quite good, and it can be seen in Figure 7(b) that the estimation results are largely incorrect, which means our method can potentially estimate almost perfect value. The remarkable point is that although the training data has concentration values with only 5% steps, it accomplished finer estimation. At first, the complexity of the data seemed the reason why some estimation failed. The median value is much lower than the average value, which means that although the overall accuracy is good, just some of the estimation results are low. Thus, we decided to compare the color using reflectance. Here, Figure 8 shows two reflectances. One is the original and the other one is the reflectance which we obtained by applying Kubelka-Munk's theory to the estimated pigment concentration.

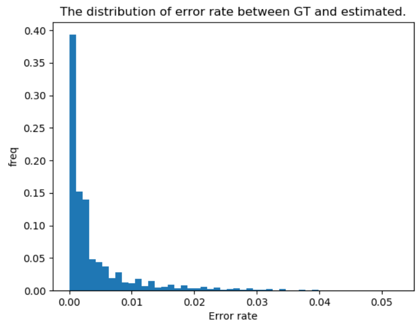
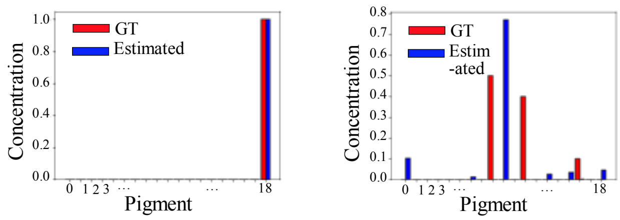


Figure 6. The distribution of error rate between GT and estimated.



(a) Best **(b) Worst**

Figure 7. The results with error rates of best and worst accuracy.

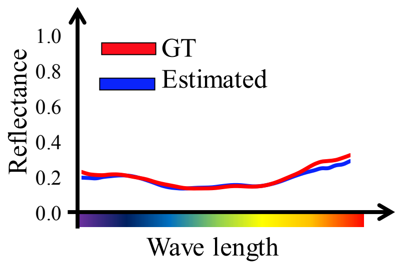


Figure 8. The original reflectance and the reflectance obtained by applying Kubelka-Munk theory to the estimated pigments concentration.

Figures 7(b) and 8 show that, although the accuracy is considerably low in terms of estimating the concentration of the pigment, the reflectance obtained from the concentration is almost the same. Also, looking at the reflectance in Figure 8, it is considered that the waveform represents a color close to black. In the case of mixing multiple colors to make black, there are innumerable combinations, and thus it is an irreversible problem to estimate the concentration of the pigment from the reflectance, which caused the estimation to fail.

5. Comparison with conventional method

We compared the proposed method with the conventional method in terms of executing speed. In general, the technique such as deep learning takes long time, however once it finishes training process, it's possible to estimate the desired value quickly. Especially, in such cases where you need to repeat estimating, and the speed is important more than the accuracy, deep learning technique is advantageous.

We measured the time taken by our method to estimate the value and the time by the error minimization for-loop. The spec of the PC we used is as follows. The CPU is AMD Ryzen 5 2600 Six-Core Processor 3.40GHz. The memory capacity is 16 GB. The result is shown in Table 2. The difference is obvious, and it shows our method is much more efficient. The point here is the fact that our method doesn't require any iterating process that is time consuming rather than the time itself. We note that in estimating by simple optimization, we assumed the combination of pigments is known so it takes more time in actual situation in which you don't kwon the pigment combination. Also, the step used in for-loop was 1%, so it will be longer if you want higher accuracy.

Table 2 the result of comparison in speed.

	Our method	Simple optimization
Time in training	51 minutes	
Time in estimating	0.00006 seconds	24 minutes

6. Conclusion and future work

In this study, we proposed the method to estimate the pigment concentration using the auto-encoder model of neural network. We showed that this method can estimate the most suitable combination of pigments and its concentration. It's applicable even if the number of possible combinations is more than 800. One of the advantages is that the model can learn some features from training data and interpolate what is not included in training data. In this time, we found just 5% steps of concentration values is enough to learn the overall trend contained in the data.

In future work, we plan to apply the proposed method to an actual painting and see how accurately it accomplishes to estimate the values in a practical situation. Moreover, it is needed to find an application of our method. Finally, it is also our task to consider the neural network model, which makes it possible to train and estimate faster and more efficiently.

References

[1] Kirchner E, Lans I, Ligterink F, Hendriks E, Delaney J. Digitally reconstructing van Gogh's field with irises near Arles. Part 1: varnish. Color Res Appl. 2017;43:150-157

[2] Kirchner E, Ivd L, Ligterink F, et al. Digitally reconstructing van Gogh's field with irises near Arles. Part 2: pigment concentration maps. Color Res Appl. 2017;43:158-176.Part 1: varnish. Color Res Appl. 2017;43:150-157

[3] Kirchner E, Lans I, Ligterink F, et al. Digitally reconstructing van Gogh's field with irises near Arles. Part 3: determining the original colors. Color Res Appl. 2017;43:311-327.

[4] Berns, Roy S. Billmeyer and Saltzman's principles of color technology. Wiley, 2019.

[5] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507

[6] Shi, Liang, et al. "Deep multispectral painting reproduction via multi-layer, custom-ink printing." SIGGRAPH Asia 2018 Technical Papers. ACM, 2018

[7] Okumura, Y. 2005. M.S. Thesis: Developing a spectral and colorimetric database of artist paint materials. Edited by RIT Scholar Works: Rochester Institute of Technology.

[8] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).