Predicting Colour Reflectance with Gradient Boosting and Deep Learning *

Asei Akanuma[1,2], Daniel Stamate[1,2], and J. Mark Bishop[2]

[1] Data Science & Soft Computing Lab, London,
[2] Department of Computing, Goldsmiths College,
University of London, UK
d.stamate@gold.ac.uk

Abstract. Colour matching remains to be a labour-intensive task which requires a combination of the colourist's skills and a time consuming trial-and-error process even when employing the standard analytical model for colour prediction called Kubelka-Munk. The goal of this study is to develop a system which can perform an accurate prediction of spectral reflectance for variations of recipes of colourant concentration values, which could be used to assist the colour matching process. In this study we use a dataset of paint recipes which includes over 10,000 colour samples that are mixed from more than 40 different colourants. The framework we propose here is based on a novel hybrid approach combining an analytical model and a Machine Learning model, where a Machine Learning algorithm is used to correct the spectral reflectance predictions made by the Kubelka-Munk analytical model. To identify the optimal Machine Learning method for our hybrid approach, we evaluate several optimised models including Elastic Net, eXtreme Gradient Boosting and Deep Learning. The performance stability of the models are studied by performing computationally intensive Monte Carlo validation. In this work we demonstrate that our hybrid approach based on an eXtreme Gradient Boosting regressor can achieve superior performance in colour predictions, with good stability and performance error rates as low as 0.48 for average dE_{CMC} and 1.06 for RMSE.

Keywords: Colour Reflectance Prediction, Paints, Coatings, AI-Machine Learning, eXtreme Gradient Boosting, Deep Learning, Elastic Net, Monte Carlo Validation

1 Introduction

Colour matching is a specialised task which requires a colourist to mix a set of colourants in different proportions until a suitable visual match is achieved between the mix and a desired colour. The mix that is produced is a formula called the 'recipe' which is a list of mixing ratio of the colourants used to obtain the

* This work was part funded by InnovateUK through a Knowledge Transfer Partnership.

colour match. Colour matching is therefore a considerably laborious task and is achieved through a combination of the colourist's skills and trial-and-error. To assist the colourists and to speed up the process of discovering recipes, Computer Colourant Formulations are often used which utilises various colour prediction software. These software allow for the prediction of colour given a recipe which are often based on analytical models. A relatively common method of Computer Colourant Formulation is to implement an analytical model based on the radiation transfer theory known as Kubelka-Munk (K-M) [1] [2] which was originally proposed in 1931 [1]. Briefly, the K-M models allows for the prediction of the spectral reflectance for a mixture of colourants by characterising them according to two coefficients which are absorption (K) and scattering (S). Though the K-M model is favoured for their simplicity and ease of use, in many cases their approximations alone are not sufficiently accurate for real world applications. Subsequently, to overcome some of the limitations of the modelling assumptions made by the K-M models, multi-flux models [3] have been proposed as alternatives. However, the improvements in prediction accuracy from these models have been relatively limited despite their added modelling complexity.

Thus, there have been numerous previous works on the application of Machine Learning methods for colour recipe predictions. In the works by Bishop et al. [4] [5] neural network models were used to predict recipes for dye concentrations from CIELAB coordinates. Bezerra and Hawkayard [6] used a neural network to predict the concentrations of florescent dyes from spectral reflectance values. On the other hand, Westland et al. employed a neural network model to predict spectral reflectance for mixtures of inks printed on cards [7]. Furferi and Governi [8] proposed a neural network based approach to correcting the spectral reflectance from an analytical model, to estimate spectrophotometer readings for carded fiber. In the work by Hung et al. [9] a neural network was trained to predict colour properties of cotton fabrics. Jawahar et al. [10] devised a neural network model to predict the tri-stimulus values for leather dyeing, which was shown to perform better than a K–M model. Hemingray and Westland [11] used a number of several separate neural networks to predict the spectral reflectance at each wavelength for fibre blends, while Pan et al. used a similar neural network for the transformation between two Spectral Spaces [12]. On the other hand, Furferi et al. [13] proposed a hybrid K-M-neural network method to predict the reflectance values of a blend. Shen et al. [14] introduced a hybrid model based on the Stearns–Noechel and a neural network model. Also, Zhang et al. [15] proposed a method to improve predictions of K-M using a hybrid of least squares and grid search method, for spectrophotometric colour matching for fiber blend. In [27] the authors of this work proposed a neural network approach to estimating colour reflectance with product independent models.

Recent works have also attempted to solve the inverse problem of spectral reflection prediction. In the works by Tarasov et al. [16], [17] and [18], a feed-forward neural network was used to predict the colourant recipe values from the observations of spectral data. The work by Zhang et al. proposed a method for

dye recipe prediction from colour measurements for cotton fabrics, using a novel recurrent neural network [19].

Our present work focuses on the prediction of spectral reflectance from colourant concentration values for paints recipes. We design a system which is able to perform predictions of spectral reflectance for variations of recipes. However, rather than attempting to model the relationship of the colourants to spectral reflectance directly, we propose a unique solution for colour prediction which is based on a combined approach of the K-M analytical model and Machine Learning. More specifically, our approach uses the Machine Learning models to correct the spectral reflectance approximations made by the K-M analytical models *thereby overcoming some limitations of the traditional analytical models.* Thus, it is a hybrid approach of the analytical and Machine Learning methods which always provides a better predictive performance than the K-M analytical model alone.

In addition, our work includes a comparison of several different varieties of Machine Learning algorithms, including Elastic Net, Extreme Gradient Boosting and Deep Learning. Interestingly, our experimental results demonstrates that the Extreme Gradient Boosting algorithm provides the most stable and best performance among them.

The remainder of the paper is organised as follows. In Section 2 we describe the methodology, the data that is used, and the Machine Learning predictive models used for this study. Section 3 discusses the experimental results. Finally Section 4 provides the conclusions.

2 Methodology

2.1 Dataset

The data used in this study originates from a commercial database of paint colour recipes used for coatings. The database includes recipes for more than 10,000 colour samples which are mixed from more than 40 different colourants in defined proportions with each having a corresponding spectral reflectance curve measured for their colour by a Spectrophotometer. The measurements of the reflectance spectra include 31 wavelengths of the visual spectrum in the range of 400 to 700 nanometres at each 10 nanometre step intervals under the illuminant D65, measured at 45 degree face angle. The reflectance values are therefore vectors of length 31 which correspond to each wavelength for the measured range of the spectral curve. All variables in the data are numerical and continuous, and thus this work is considered a regression task of predicting 31 target variables.

Data cleansing: For the data cleansing process we removed samples with incomplete paint recipes and incomplete measurements. Additionally, as a precautionary measure, we excluded some of the data by ranking the samples which have the highest dE_{CMC} [26] errors between the measured colour and a predicted colour from the K-M analytical model, by removing 2 percent of those

top ranked samples. The assumption here is that some of the samples may contain errors which may originate from the data collection or from the process of entering the data in the database. Using the K-M analytical model also as a proxy for assessing sample quality, we find this data cleansing process is able to assure better quality of data, resulting in a dataset with a population of reliable and consistent samples.

2.2 A hybrid approach to colour prediction

The aim of this work is to perform a reliable prediction of the spectral reflectance for any given recipe. Our approach distinguishes itself from most colour prediction approaches in the sense that we combine the analytical K-M model with Machine Learning methods. This can be described as a two step process of (1) making an initial prediction of the spectral reflectance by using a K-M model for the recipe from the colourant concentration values, followed by (2) using the Machine Learning model to correct the initial prediction made in step (1) such that the final estimates are closer to the actual spectral reflectance (measured by the spectrophotometer). This is achieved by optimising the Machine Learning models to predict the residuals of the K-M model's predictions of the measured spectral reflectance values. For the inference of new datapoints, the K-M model's predictions are summed up with the Machine Learning model's predictions.

The following gives a description of the input and output variables for the Machine Learning predictive models used in this work.

Descriptions of input variables:

- The paint recipe which are the concentrations of the colourants given by percentage composition values - a sparse vector where each element represents the concentration amount of a colourant (where a value of 0 indicates that the colourant is not in use).
- The spectral reflectance of the K-M model's predictions - a vector of length 31 for each value in the spectral reflectance curve.

Descriptions of the target variables: The target variables in this study are the residuals between the measured spectral reflectance values of the 31 wavelengths and the predictions made initially by the K-M model.

2.3 Predictive models and fine tuning

In this study we developed several optimised predictive models in order to compare performances among them. The predictive models that we explored in this work are the Elastic Net, two types of eXtreme Gradient Boosting (linear and tree based models) and Deep Learning.

Elastic Net: Firstly, an Elastic Net model [20] was tuned for each of the 31 target variables individually. Elastic Net is a type of regularized regression model which provides a middle ground between Ridge regression and Lasso regression. Elastic Net uses a regularisation term which is a simple mix or both Ridge and Lasso which is shown in the Equation 1 below.

$$J(\theta) = MSE(\theta) + r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^{n} \theta_i^2 \tag{1}$$

The r hyperparameter is a regularisation mix ratio; when $r = 0$, Elastic Net is equivalent to Ridge regression, whereas when $r = 1$, it is equivalent to Lasso regression. The α hyperparameter represents the strength of the regularisation. The Elastic Net was tuned by searching for the optimal hyperparameter values by a using a grid search with a 3 fold-cross validation on the training set in the search ranges shown in Table 1. A final model was then fitted on the entire training set data.

Table 1. Hyperparameter search grid for Elastic Net algorithm

ENet hyperparameters	Hyperarameter Search Range
α	[0.001 - 100]
r	[0.1 - 1]

eXtreme Gradient Boosting with linear models: Secondly, we applied the eXtreme Gradient Boosting [21] regressor algorithm based on linear models, with the squared error objective. For this implementation, 31 individual eXtreme Gradient Boosting models were tuned with a grid search to find the optimal configurations for the *learning rate, estimator numbers, lambda and alpha* hyperparameters using a 3 fold-cross validation on the training set. After the optimal hyperparameters were found, the final models were fitted on the entire training set data. Table 2 shows the table of the searched hyperparameter ranges.

Table 2. Hyperparameter search grid for eXtreme Gradient Boosting (linear based) algorithm

XGBoost (linear) hyperparameters	**Hyperparameter Search Range**
learning rate	[0.05 - 0.2]
n estimators	[500 - 2500]
lambda	[0.1 - 100]
alpha	[0.1 - 100]

eXtreme Gradient Boosting with Tree models: Thirdly, we tuned the eXtreme Gradient Boosting regressor based on tree models, with the squared error objective. 31 individual eXtreme Gradient Boosting models were tuned with a grid search to find the optimal configurations for the hyperparameters for *learning rate, max depth, subsample rate, min child weight, number of estimators, column sample by node, column sample by tree* for the search ranges shown in Table 3. Once the optimal hyperparameters were found, the final model was fitted on the entire training set.

Table 3. Hyperparameter search grid for eXtreme Gradient Boosting (tree based) algorithm

XGBoost (Tree) hyperparameters	**Hyperparameter Search Range**
learning rate	[0.05 - 0.2]
max depth	[5 - 20]
subsample	[0.5 - 1]
min child weight	[0.5 - 1]
n estimators	[500 - 2500]
colsample bynode	[0.33 - 1]
colsample by tree	[0.33-1]

Deep Learning: For our Deep Learning approach we experimented with various Multilayer Perceptron (MLP) model architectures including a fully-connected feed-forward model, a Resnet [22] like model (with skip-connections), and a wide-and-deep-learning [23] like model. In all the experiments, the network architecture is a single model which has a fixed number of input nodes which is equal to the number of the variables used, and has 31 output nodes. Different configurations were searched for the optimal number of layers, hidden node numbers, and different hidden node activations such as:

- Logistic Sigmoid:

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$

- Hyperbolic tangent:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- ReLU (Rectified Linear Unit):

$$\text{ReLU(x)} = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

– SELU and ELU:

$$\text{SELU}_{\alpha,\lambda}(x) = \lambda \begin{cases} \alpha(\ \exp(x) - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

where α and λ are constants 1.6732 and 1.0507 respectively for standard scaled inputs., and

$$\text{ELU}_{\alpha}(x) = \begin{cases} \alpha(\ \exp(x) - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

where value for alpha is picked typically between 0.1 and 0.

– Softplus:

$$\text{Softplus}(a) = \ln(1 + e^a),$$

The Deep Learning model was tuned using the Adam optimiser [25] to minimise various loss functions such as the *MSE*, *MAE* and *Huber* loss. The Huber loss L_δ, as defined below, was used in particular to prevent the potential impact of any remaining outliers in the data:

$$L_\delta = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta \\ \delta(|y - f(x)| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

The network was trained up to 2000 epochs to minimise the loss. In addition, to control for model overfitting, the L2 regularisation was applied to the hidden layer weights, and an early stopping criteria was used to find the appropriate number of epochs, by observing if a validation loss did not improve within the next 100 epochs based on a validation set which was randomly selected from 10 percent of the training set population. Table 4 shows the searched ranges for hyperparameters values in the Deep Learning algorithm.

Table 4. Hyperarameter search grid for Deep Learning algorithm

Deep Learning params	**Hyperparameter Search Range**
number of layers	[1 -5]
hidden node size	[50 - 500]
loss functions	[MSE, MAE, Huber]
learning rate	[0.001 - 0.1]
L2 regularization	[0.0001 - 0.1]
hidden activations	[sigmoid, tanh, relu, elu, selu, softplus]

2.4 Hardware and software

Due to the significant computational requirements of the Monte Carlo validation that we use in our framework, the implementation of this work was performed on two servers with Xeon 6-cores processors and 96GB of RAM each, and one server with a Ryzen 16-cores processor, 128GB of RAM and an RTX 3080 GPU. The experiments were carried out with Python 3 and Numpy, Pandas, scikit-learn, XGBoost, Tensorflow and Keras packages.

3 Experiments and Results

3.1 Evaluating performances

From the dataset, 90 percent was used as the training set for optimising and building the predictive models while the remaining 10 percent was used as testing set to evaluate the performance of the modelling.

The main concern of this work is to minimise the visual difference between the spectrophotometer's measured colour and the predicted colour of the recipe samples. Therefore the dE_{CMC} colour distance [26] in the Equation 2 below is appropriate for measuring the performances of the proposed methods as it takes the colour sensitivities of the human visual system into consideration. For each of the compared models, the performance was measured as the calculated averages of dE_{CMC} for the test set predictions. Additionally, in the Results subsection below we also provide the root mean squared error RMSE for the prediction results.

$$dE_{CMC} = \sqrt{\left(\frac{dL}{lS_L}\right)^2 + \left(\frac{dC}{cS_C}\right)^2 + \left(\frac{dH}{S_H}\right)^2} \tag{2}$$

3.2 Monte Carlo Validation

To investigate stability of the performance of the models we further evaluate results by running a Monte Carlo validation which consists of repeating the process of randomly splitting the dataset into training and testing sets as described previously, rebuilding the models on the training sets and evaluating them on the test sets. This allows for the variation in model performances to be assessed. For the Monte Carlo validation, the average and the standard deviation (SD) of the performances are provided for each performance measure. Our Monte Carlo validation consists of 30 runs, i.e. 30 data splits with model training and evaluation.

3.3 Results

A summary of the results for the best performances, measured as dE_{CMC} and $RMSE$ errors (smaller figures are better for both measures), can be found in

Table 5. Evaluation of performances for different algorithms for average dE_{CMC} and $RMSE$

Machine Learning Methods	**Avg.** dE_{CMC}	$RMSE$
Elastic Net	0.675	1.366
Extreme Gradient Boosting (Linear)	0.674	1.326
Extreme Gradient Boosting (Trees)	0.479	1.057
Multilayer Perceptrons	0.539	1.111

Table 6. Evaluation of performances of different algorithms for average dE_{CMC} and $RMSE$ with Monte Carlo validation based on 30 runs

Machine Learning Methods	**Avg.** dE_{CMC}	$RMSE$
Elastic Net	0.708 ± 0.009	1.400 ± 0.045
Extreme Gradient Boosting (Linear)	0.677 ± 0.018	1.403 ± 0.062
Extreme Gradient Boosting (Trees)	0.477 ± 0.010	1.089 ± 0.047
Multilayer Perceptrons	0.547 ± 0.014	1.188 ± 0.061

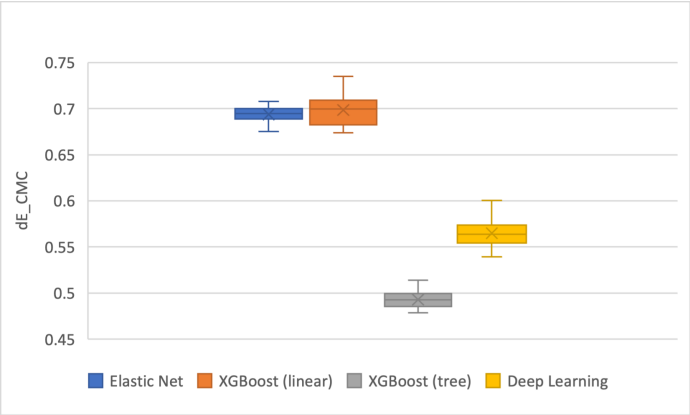

Fig. 1. Box plots for average dE_{CMC} performance results for Monte Carlo validation based on 30 runs

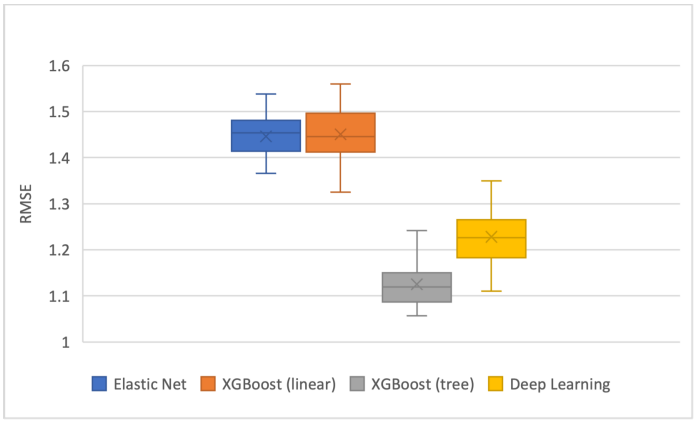

Fig. 2. Box plots for $RMSE$ performance results for Monte Carlo validation based on 30 runs

Table 5. Of the 4 compared Machine Learning methods, the best model was obtained using the eXtreme Gradient Boosting with trees, which achieved the best performances of 0.479 and 1.057 for average dE_{CMC} and $RMSE$, respectively. This was followed by the second best model obtained using the Deep Learning method. The latter achieved 0.539 and 1.11 for average dE_{CMC} and $RMSE$, respectively. Elastic Net and the Extreme Gradient Boosting linear based methods however, achieved the two worst performances across the 4 Machine Learning methods.

A similar ranking of performances can be observed from the average results of 30 runs in the Monte Carlo validation, with eXtreme Gradient Boosting based on trees, being on top and followed by Deep Learning, as shown in Table 6. Moreover it should be noted that the most stable models are obtained with the eXtreme Gradient Boosting based on trees, and with Elastic Net methods, since they led to the lowest standard deviations (SD) in the Monte Carlo validation procedure, as shown in the same Table 6. The performance ranking and stability conclusions above are also visually reconfirmed by the boxplots shown for dE_{CMC} and $RMSE$ in Fig. 1 and Fig. 2, respectively (lower positioned boxplot is better for performance, and smaller boxplot is better for stability).

4 Discussions and Conclusions

Currently, colour matching is still a labour-intensive task which requires a combination of the colourist's skills and a time consuming trial-and-error process even when employing one of the most used analytical models for colour prediction, namely Kubelka-Munk (K-M). The goal of this study was to develop a system which can perform an accurate prediction of spectral reflectance for variations of recipes of colourant concentration values, which could be used to assist the colour matching process.

This work explored the prediction of spectral reflectance from colourant concentration values using a dataset of paints recipes, and proposed a combined approach based on the K-M conventionally-employed analytical method and a selection of Machine Learning methods. This hybrid approach involves using optimised Machine Learning models to correct the initial predictions made by a K-M model trained on the data. In particular, the Machine Learning models were used to predict the residuals of the K-M model, computed as the difference between the measured spectral reflectance values of the colour recipes and the K-M predictions. In order to obtain the final reflectance predictions, the Machine Learning predictions were then added to the K-M predictions. We explored 4 Machine Learning methods for our proposed approach, including Elastic Net, two types of eXtreme Gradient Boosting methods (linear and tree based), and Deep Learning. The Machine Learning methods were assessed for their performances (for which lower is better) in terms of the error measures of average dE_{CMC} and $RMSE$. The experiments demonstrated that the best performing model was the tree based eXtreme Gradient Boosting which outperformed all other models tested, achieving the smallest error measures of dE_{CMC} and $RMSE$. Moreover, the same model was also the most stable, together with Elastic Net which in turn was among the two worst performing models with respect to dE_{CMC} and $RMSE$.

References

1. Kubelka P, Munk F. Ein Beitrag zur Optik der Farbanstriche. Zurich: Technische Physik; 1931.
2. J.H. Nobbs, "Kubelka-Munk theory and the prediction of reflectance" (1985), Review of Progress in Coloration (SDC), 15, pp. 66-75.
3. Mudgett, P. S., and L. W. Richards. "Multiple scattering calculations for technology." Applied Optics 10.7 (1971): 1485-1502.
4. Bishop, J.M., Bushnell, M.J. & Westland, S., (1991), The Application of Neural Networks to Computer Recipe Prediction, Color, 16 (1), pp. 3-9.
5. Bishop, J.M., Bushnell, M.J., Usher, A. & Westland, S., (1991), Neural Networks in the Colour Industry, Proc. AIENG 1991, pp. 423-434, Oxford, UK.
6. Clovis de M Bezerra and CJ Hawkyard. Computer match prediction for fluorescent dyes by neural networks. Coloration Technology, 116 (5-6):163 169, 2000. 6, 7
7. Stephen Westland, Laura Iovine, and John M Bishop. Kubelka-Munk or neural networks for computer colorant formulation?, Proceedings of the SPIE, Volume 4421, p. 745-748 (2002)
8. Rocco Furferi and Lapo Governi. Prediction of the spectrophotometric response of a carded fibre composed by different kinds of coloured raw materials: An artificial neural network-based approach. Color Research and Application, 36:179191, 2011. 5, 7.
9. Hung, O. N., et al. "Using artificial neural network to predict colour properties of laser-treated 100% cotton fabric." Fibers and polymers 12.8 (2011): 1069-1076.
10. Malathy Jawahar, Narasimhan Kannan, Chandra Babu, and Mehta Kondamudi Manobhai. Artificial neural networks for colour prediction in leather dyeing on the basis of a tristimulus system. Coloration Technology, 131(1):48 57, 2015. 5, 7.

11. Hemingray, Caroline, and Stephen Westland. "A novel approach to using neural networks to predict the colour of fibre blends." Coloration Technology 132.4 (2016): 297-303.
12. Pan, Q., P. Katemake, and S. Westland. "Neural Networks for Transformation to Spectral Spaces." ACA 2016 China, Color Driving Power, Proceedings. ACA, CFCA, CAC, 2016.
13. Furferi, R., Governi, L., & Volpe, Y. (2016). Color matching of fabric blends: Hybrid Kubelka-Munk+ artificial neural network based method. Journal of Electronic Imaging, 25(6), 061402
14. Shen, J., Zhou, X., Ma, H., & Chen, W. (2017). Spectrophotometric prediction of pre-colored fiber blends with a hybrid model based on artificial neural network and Stearns–Noechel model. Textile Research Journal, 87(3), 296-304.
15. Zhang, G., Pan, R., Zhou, J., Wang, L., & Gao, W. (2021). Spectrophotometric color matching for pre-colored fiber blends based on a hybrid of least squares and grid search method. Textile Research Journal
16. Tarasov, Dmitry, Oleg Milder, and Andrey Tyagunov. "Inverse problem of spectral reflection prediction by artificial neural networks: Preliminary results." 2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). IEEE, 2018.
17. Tarasov, Dmitry, and Oleg Milder. "The Inverse Problem of Spectral Reflection Prediction by Artificial Neural Networks: Neugebauers Primaries vs. Recipes." 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). IEEE, 2019.
18. Tarasov, Dmitry A., and Oleg B. Milder. "The inverse problem of spectral reflection prediction: Problems of framework selection." AIP Conference Proceedings. Vol. 2293. No. 1. AIP Publishing LLC, 2020.
19. Zhang, Jianxin, et al. "Dyeing recipe prediction of cotton fabric based on hyperspectral colour measurement and an improved recurrent neural network." Coloration Technology 137.2 (2021): 166-180.
20. Zou, Hui; Hastie, Trevor (2005). "Regularization and Variable Selection via the Elastic Net". Journal of the Royal Statistical Society, Series B. 67 (2): 301–320
21. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.
22. He, K., Zhang, X., Ren, S., and Sun, J. 2016. Identity mappings in deep residual networks. In European conference on computer vision (pp. 630- 645). Springer, Cham.
23. Cheng, Heng-Tze, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson et al. "Wide & deep learning for recommender systems." In Proceedings of the 1st workshop on deep learning for recommender systems, pp. 7-10. 2016.
24. Géron, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc., 2019.
25. Kingma, Diederik P., and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
26. Robertson AR. "CIE guidelines for coordinated research on colour difference evaluation". Color Res Appl 1978;3:149 –151.
27. Akanuma, Asei and Stamate, Daniel. A Neural Network Approach to Estimating Color Reflectance with Product Independent Models. 31st International Conference on Artificial Neural Networks – ICANN 2022, LNCS vol 13531, Springer