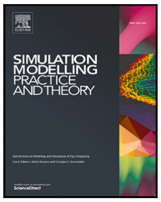
Simulation Modelling Practice and Theory 114 (2022) 102400

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat



A comparison of 2DCNN network architectures and boosting techniques for regression-based textile whiteness estimation

Thanasis Vafeiadis [a,*], Nikolaos Kolokas [a], Nikolaos Dimitriou [a], Angeliki Zacharaki [a], Murat Yildirim [b], Habibe Gülben Selvi [b], Dimosthenis Ioannidis [a], Dimitrios Tzovaras [a]

[a] *CERTH/ITI (Information Technologies Institute, Center for Research and Technology), Hellas, 57001, Thessaloniki, Greece*
[b] *Zorluteks Textile Trading and Production CO., Lüleburgaz, 39750, Kırklareli, Turkey*

ARTICLE INFO

Keywords:
Industry 4.0
2DCNN
Regression
Machine learning
Random Forest
XGBoost
Deep learning
Textile whiteness

ABSTRACT

This paper presents a comparative assessment of two-dimensional convolutional neural networks (2DCNN) and boosting methods for regression-based textile whiteness estimation, applied to high resolution images of textiles of an industrial cotton textiles producer, labeled with whiteness values, thus enabling supervised learning. The images were taken under various lighting conditions. Concerning the machine learning methods, Random Forest and XGBoost were the selected and tested boosting techniques on which model hyper-parameter tuning was applied, whereas regarding the 2DCNN architectures, the known from literature ColorNet architecture was selected and a more shallow one, called WERegNet, was introduced. Data augmentation was applied during pre-processing, due to the limited amount of available data. Based on the simulation results, the WERegNet architecture surpasses ColorNet and XGBoost in terms of performance, while it is comparable with Random Forest on test set, based on model selection measure Normalized Root Mean Squared Error.

1. Introduction

Colorimetry is the science and technology used to quantify and describe physically the human color perception [1]. In colorimetry, whiteness is defined as a measure of how closely a surface matches the properties of a perfect reflecting diffuser, i.e. an ideal reflecting surface that neither absorbs nor transmits light, but reflects it at equal intensities in all directions. For the purposes of this standard, the color of such a surface is known as preferred white. Whiteness index (WI) is a measure which correlates the visual ratings of whiteness for certain white and near-white surfaces [2].

Bleaching is a process performed by the textile industry so that fabric achieves the required whiteness, which has to be of high level in some cases [3,4]. It is included in the finishing stage of textile manufacturing [5], along with other processes.

There is a number of different indices available, as discussed below. However, many of them are based on the standard definition of the International Commission on Illumination (Commission internationale de l'éclairage, CIE), which is the international authority on color, color spaces, light and illumination. It was founded in 1913 as a successor to the Commission Internationale de Photométrie, that was established in 1900, and currently is based in Vienna, Austria [6]. CIE Standard Illuminant D65 is a common standard illuminant, belonging to the D series of illuminants, which aim at portraying standard illumination conditions under natural light in different parts of the world [7]. Particularly, D65 approximates the average midday light in Western/Northern Europe, and it is characterized by CIE as the standard daylight illuminant [8]. D65 light is generated only by simulators, the quality of which

[*] Corresponding author.
E-mail address: thanvaf@iti.gr (T. Vafeiadis).

https://doi.org/10.1016/j.simpat.2021.102400
Received 26 May 2021; Received in revised form 24 August 2021; Accepted 25 August 2021
Available online 4 September 2021

is evaluated by the CIE Metamerism Index [9,10]. CIE has provided whiteness and tint formulas useful to compare wheteness of samples evaluated for D65 [11].

On the other hand, in recent years deep learning and convolutional networks have gained a lot of traction in quality control for Industry 4.0 [12] with several methods proposed for micro-electronics [13,14], metal parts manufacturing [15], visual inspection process in the printing [16], visual quality control automation on the assembly line [17], spectrophotometric color correction [18] and geometric deep lean learning to support sustainable organizational growth [19] among others.

The motivation behind this study is, firstly, to evaluate the suitability and performance of deep learning architectures and state-of-the-art machine learning methods on the regression problem of textile whiteness estimation on fabrics and, secondly, to decide for a robust regression model to be used in an industry-specific evaluation toolkit embedded in a sensor. This work constitutes a comparison between deep learning architectures and widely known and used machine learning techniques using high resolution images from fabrics as input. In particular, the performance of 2D convolutional neural network (2DCNN) architectures, one known from scientific literature and a proposed one that is a more lightweight network, along with Random Forest and XGBoost boosting machine learning techniques, is compared using normalized root mean square error so as to overcome the problem of comparing models with different scales. Boosting machine learning approaches were selected compared to non-boosting version because they are more robust and effective learning algorithms and have been consequently utilized in many domains such as telecommunications [20], civil engineering [21–23] and econometrics [24]. The input dataset of machine learning models is based on images of fabrics of a cotton home textiles producer, with eleven distinct values of Berger whiteness index. Due to the limited amount of available training samples, a data augmentation technique based on width-shift of initial images has been employed before training.

The remainder of the paper is organized as follows. In Section 2, whiteness estimation methods based on measurements from specialized instruments and without use of machine learning are presented. In Section 3, the use case and data are described in more detail, whereas in Section 4 selected and tested machine learning approaches are presented. Section 5 includes the parameters settings of machine and deep learning algorithms, while Section 6 provides a brief description of simulation results. Finally, in Section 7, a discussion about the results and other aspects is provided and in Section 8 our conclusions are drawn.

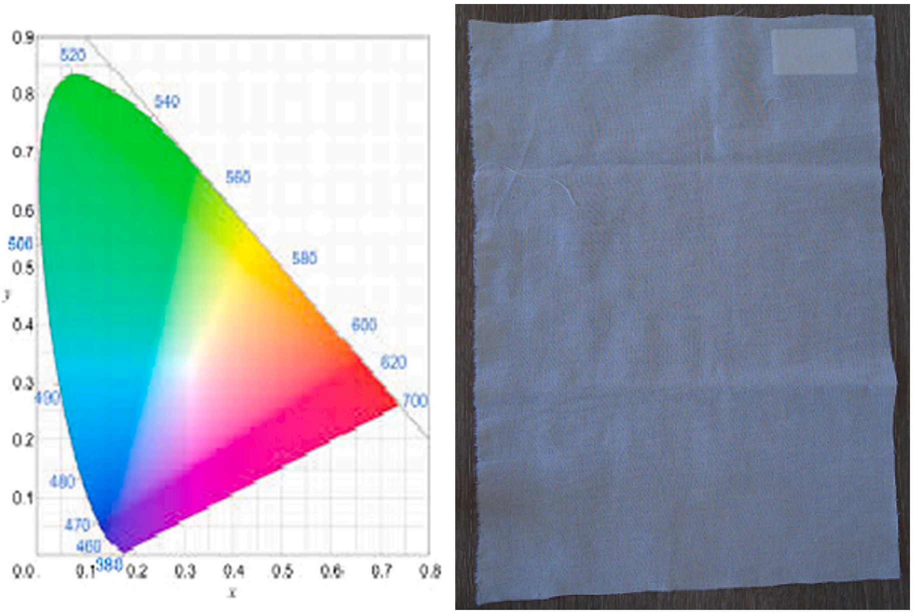
2. Whiteness estimation without machine learning

This section includes bibliographic references related to whiteness estimation, starting from those not (exclusively) related to textiles, and continuing with others, pertinent to fabrics. The references in each paragraph have been cited in chronological order.

In the technique of [25], which determines the dominant wavelength of corrected fluorescent spectra, color difference formulas for non-fluorescent materials are used. The work in [26], which points out the difficulty to use a universal whiteness formula, since the ideal whiteness is subjective and depends on the material type (e.g. textile), presents a spectrocolorimetric technique to instrumentally estimate whiteness. In [27], a new whiteness formula is introduced, which adapts the traditional formula of the International Commission on Illumination (CIE) to address tint and excitation purity. The recent paper [28] defined two new formulas, the first (which was mostly proposed) based on CIECAM02, and the second combining the CAT02 chromatic adaption transform with the CIE formula. In [29], similar authors as in the previous paper evaluate several innovative whiteness formulas facing the limitations of the CIE formula, mentioning the whiteness formula with the CAT02 chromatic adaptation transform as generally the best. Finally, the work in [30] deals with whiteness of teeth, assessed with the whiteness index for dentistry.

Continuing with bibliography about textiles only, in [31], textile whiteness is computed by lightness, redness and yellowness, while the impact of fluorescence is also considered. In [32], a more recent standard method is used, based on reflection coefficient. In [33], fabric whiteness is measured by a Gretag Macbeth SpectroEye spectrophotometer using the common D65 illuminate and 10° additional normal observer function, to evaluate its deterioration due to exposure to light. In [34], Hunter L, a, b system is described, whereas [35] concludes that the CIE formula surpasses that of Uchida for fabric whiteness estimation. The first found reference which uses the Berger whiteness index (considered in the current work) is [36]. In [37], textile whiteness is computed by the Datacolor apparatus, whereas [38] uses the CIE whiteness and whiteness tint formulas to analyze optically whitened woven or knitted cellulosic textiles, irradiated with xenon arc light. The work in [39] investigates the correlation of whiteness and yellowness depending on hue, and [40] estimates whiteness of hydrogen peroxide bleached cotton textile according to the American Society of Testing Methods (ASTM) Method E31373, related to CIE. Whiteness estimation of the cotton fabric samples in [41] is based on the American Association of Textile Chemists and Colorists (AATCC), particularly test method 110. In [42], CIE whiteness of cotton textile is assessed from tristimulus values using spectrophotometer. Hunter's formula was used also in [43] to evaluate the effect of environmentally-friendly scouring ("bioscouring") of fibers, using spectrophotometer for the relevant measurements. Bioscouring of cotton textiles is also discussed in [44], where the Berger whiteness index has been used and expressed as a quadratic function of pectate lyase, lipase, protease and xylanase. In [45], the ASTM-E313-00 method evaluates the whiteness of woven textile after desizing, scouring and bleaching. In [46], the CIE technique is used for D65 and 10° observer, to evaluate the effect of two softeners on cotton textile treated with butane tetracarboxylic acid (BTCA). Instead, in [47], the treatment of fabric refers to the application of polyaldehyde trehalose (OTr), and whiteness is computed with the AATCC Testing Method 11-2005.

To the best of our knowledge, there is no related work devoted exclusively to whiteness estimation using machine learning. However, some machine learning algorithms useful for color recognition are described in Section 4, along with the alternative approach proposed in this paper.



(a)

(b)

Fig. 1. (a) CIE 1931 color space, the first defined quantitative link between distributions of wavelengths in the electromagnetic visible spectrum and physiologically perceived colors in human color vision. (b) Example textile with Berger 66.

3. Use case and data set description

3.1. Problem definition

The Berger WI, which is used in this work, is specified for illuminant C and 2-degree observer functions only. However, the equation is commonly used with other illuminants and observer functions, therefore the value shown will depend on the primary illuminant and the observer function you have chosen. The formula is: $WI = Y + 0.3018Z - 3.831X$. In this formula, the X, Y, Z values represent 3 different colors (red/green/blue) on a 3D space. The value of 100 Berger stands for perfect whiteness, which is located at the center of the XYZ tristimulus given in Fig. 1(a).

The aim is to produce fabric with whiteness of 60 Berger at the end of the bleaching process, and for this, a fast, autonomous and reliable Artificial Intelligence (AI) algorithm for whiteness estimation, embedded in a sensor, is needed.

3.2. Data set description

The data set contains high-resolution images with graphic's display resolution of 4096 × 3000 or $4K \times 3K$. Eleven different types of WI textile fabrics are available (with WI equal to 51, 56, 58, 61, 62, 66, 68, 70, 74, 77 and 78), and for each type, twenty high-resolution images (ten from the front and ten from the rear) where captured, so overall 220 images were created (see Fig. 1(b)). WI for the training samples were measured in laboratory with specialized equipment, and for each sample illumination was changed at predetermined levels. Obviously, the number of annotated samples is very small for a 2DCNN network, therefore a data augmentation step has been implemented, which will be described in a subsection below.

4. Machine learning approaches

As mentioned above, this article has as main objectives to evaluate known boosting machine learning techniques along with deep learning architectures, on the annotated data sample of textile fabrics, to address the problem of assessing the whiteness of fabrics as a regression and not as a classification problem. For that, Random Forest and XGBoost machine learning techniques are selected among others, because are less computationally expensive and do not require GPU resources neither for training and searching the proper set of hyper-parameters, nor for testing. As for deep learning, an architecture, known from literature, for color classification is selected, denoted hereafter as ColorNet, along with a proposed architecture, more sallow compared to ColorNet if we consider the depth of the network, but effective, denoted hereafter as WERegNet. An extensive description of all tested methodologies is provided below.

4.1. Random Forest

Random Forest, also known as random decision forest, is an ensemble of decision trees with each one constructed by using a random subset of the training data, while the output class is the mode of the classes decided by each decision tree [48]. The Random Forest machine learning technique is easy-to-use, flexible and simple, and provides great results most of the time with minimum time spent on hyper-parameter tuning. This machine learning technique has gained popularity due to the fact that it can be used for both classification and regression problems. The fundamental idea behind Random Forest methodology is that it manages to combine the predictions made by many, single, decision trees into an overall model, and uses averaging to improve the predictive accuracy and control over-fitting. Predictions made by individual decision trees may not be that accurate as in a whole forest but combined together are closer to the desire value, on average.

Random Forest has a wide range of hyper-parameters available for tuning, so that predictions are as accurate as possible. Thus, in this work an exhaustive grid search, a brute force approach that scans the whole grid of hyper-parameters combinations, is performed regarding the random forest hyper-parameters given below: (a) number of trees in the forest (*n_estimators*), (b) maximum number of features considered for splitting a node (*max_features*), (c) maximum number of levels in each decision tree (*max_depth*), (d) minimum number of data points placed in a node before the node is split (*min_samples_split*), (e) minimum number of data points allowed in a leaf node (*min_samples_leaf*) and (f) bootstrap method for sampling data points, with or without replacement (*bootstrap*).

4.2. XGBoost

XGBoost (Extreme Gradient Boosting) is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework [49]. The gradient boosting framework is optimized through parallel processing, tree-pruning, handling missing values and regularization to avoid over-fitting or bias. It is a supervised learning algorithm that has the ability to develop fast and high performance gradient boosting tree models. It has been very popular in recent years due to its versatility, scalability and efficiency [49]. The XGBoost learning algorithm is extremely fast, due to its parallel computation ability, high efficiency, and the fact that it can be used for classification and regression problems, as well as to extract variable importance without requiring initial feature engineering on the data, such as missing values imputation, scaling and normalization. Like Random Forest, XGBoost has a wide range of hyper-parameters and their proper tuning could also lead to accurate predictive results. Thus, an exhaustive grid search regarding the XGBoost hyper-parameters also took place. The hyper-parameters of the XGBoost algorithm tuned in this work are given below: (a) number of trees (or rounds) in an XGBoost model (*n_estimators*), (b) learning in the gradient boosting model, also called shrinkage or eta in XGBoost documentation (*learning_rate*), (c) maximum depth of a tree (*max_depth*), (d) subsample ratio of the training instances (*subsample*) and (e) minimum number of instances needed to be in each node (in regression tasks) (*min_child_weight*).

4.3. Color Recognition Neural Network (ColorNet)

Color Recognition Neural Network ColorNet, is a vehicle color recognition method using convolutional neural network (CNN). CNN is a type of neural network but instead of using fully connected layer, CNN uses convolutional layer [50] to extract features from data. The training mechanism is very similar to normal neural network and uses stochastic gradient descent as training algorithm [51]. The input image is converted into two different color spaces, HSV and CIE Lab, and the ColorNet architecture consists of 2 base networks and 8 layers for each base network with total 16 layers. ColorNet is a deep network architecture with 44.091.457 trainable parameters and 704 non-trainable parameters.

4.4. Whiteness Estimation Regression Neural Network (WERegNet)

The architecture of the proposed network WERegNet is shown in Fig. 2. It is a 2DCNN with an input image with 3 channels with 150,528-dimensional, or with 224 × 224 × 3 resolution. There are five convolutional layers blocks consisting of sequences of 2D convolutional, ReLU activation function and max pooling layers that progressively drop the spatial dimensions of the input tensor while increasing the number of channels. Concretely, the first convolutional block has 64 filters, with kernel size and stride pixels to be (11,11) and (4,4), respectively. The following convolutional layer consists of two blocks that have 64 filters, with kernel size and stride pixels to be (3,3) and (1,1), respectively. The third convolutional layer has 128 filters, with kernel size and strides pixels to be (3,3) and (1,1), respectively. The fourth convolutional layer has same architecture with the second, except that the filters are set at 128. The fifth convolutional layer is same as the fourth except that filters drop in half. The max pooling pool size is set at (3,3), whereas stride pixels are set at (2,2) and employed in the first, second and fifth convolutional layers. The dropout layer is set at 20% and employed in the first, second, fourth and fifth convolutional layers. The network architecture of WERegNet is not that deep as that of ColorNet and has 24.089.281 trainable parameters and 0 non-trainable parameters. A flattening and 3 dense layers, with 4.096 neurons the first two and 2.018 neurons the third one, are applied at the end of the network, so as to estimate whiteness index of the input image.

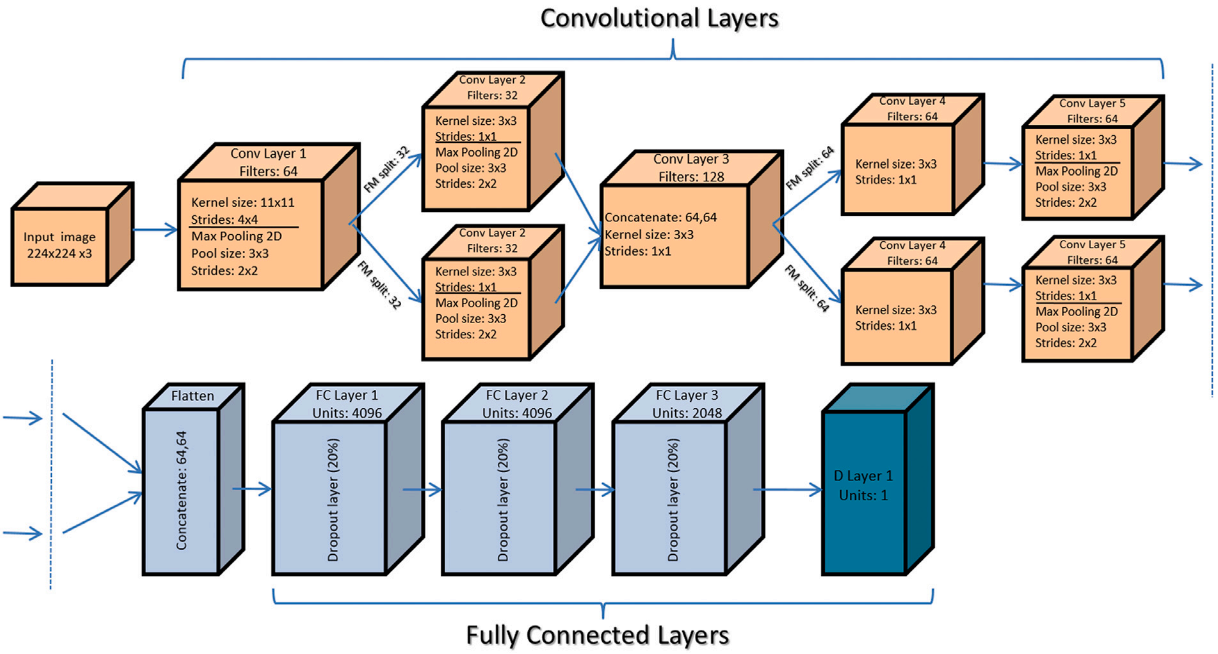


Fig. 2. The WERegNet architecture. In each block, filter size, kernel size, strides and max pooling internal parameters are reported.

4.5. Evaluation measures

In statistical analysis and particularly in regression models, a common way of measuring the quality of the fit of a certain model is the Root Mean Square Error (RMSE) [52], also called Root Mean Square Deviation, given by:

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

where y_i is the i-th actual value, $\hat{y}$ is the predicted value by the selected model and n is the size of the actual test set. If the predicted values are close enough to the actual one's the RMSE will be small, while in case where predicted and actual are substantially different (at least for some cases) the RMSE will be large. A value of zero would indicate a perfect fit to the data [53–55]. Despite the fact that, in this work, the models that were compared do not have different response variables and modifications, such as sqrt, standardized or log-transformed, it is decided to calculate and use the Normalized Root Mean Square Error (NRMSE), as the main evaluation measure, because NRMSE has the ability to be further categorized into different performance levels. Thus, from the different available approaches to calculate the NRMSE from RMSE, utilizing the mean, the difference between maximum and minimum, the inter-quartile range and the standard deviation, the *std*-NRMSE [52], given below:

$$std - NRMSE = \frac{RMSE}{std(y)} \tag{2}$$

is the best option out of four versions referred above, because it presents the ratio between the variation not explained by the regression vs the overall variation in the actual value y. If the regression explains all of the variation in actual value y, nothing gets unexplained and the *std*-NRMSE is zero. If the regression explains some part and leaves some other unexplained, which is at a similar scale with the overall variation, the ratio will be around 1. Any value of the *std*-NRMSE beyond 1 will indicate a much greater variation or noise than in the variable itself and consequently a low predictability. Thus, the desired value of *std*-NRMSE is below 1 and as much as possible near zero [53].

5. Simulation setup

Below, a brief presentation of the parameters of each regression methodology used in the simulation is given. The combination of all the values of parameters for each methodology is provided within the description of each subsection below. To that end, all simulations were performed using the Python programming language.

Table 1
MSE, RMSE, and *std*-NRMSE for validation and test set of Random Forest for optimal hyper-parameters set.

Data augmentation: not applied			
	MSE	RMSE	*std*-NRMSE
Validation set	0.0489	0.2211	0.0197
Test set	23.8	4.8785	0.8246
Data augmentation: applied			
	MSE	RMSE	*std*-NRMSE
Validation set	15.8684	3.9835	0.4205
Test set	15.9251	3.9906	0.6745

5.1. Random Forest

The definition of simulation parameters for Random Forest varies depending on internal values that are chosen. Thus, for the *n_estimators* parameter 100 different numbers of forest trees from 200 to 1000 are chosen, the *max_features* parameter varies as: (‘auto’, ‘sqrt’), the *max_depth* parameter varies from 10 to 110 with step 10 (overall 11 values), the *min_samples_split* parameter varies as: (2, 5, 10), the *min_samples_leaf* parameter varies as: (1, 2, 4), and finally the *bootstrap* parameter varies as: (‘True’, ‘False’). The combination of all the values of each hyper-parameter for Random Forest gives overall 36000 cases. Training time for Random Forest was up to 16 for the cases described above and the whole training session was performed on two Tesla K40m GPUs.

5.2. XGBoost

The definition of simulation parameters for XGBoost varies depending on internal values chosen. Thus, for the *n_estimators* parameter 100 different numbers of trees (or rounds) from 200 to 1000 are chosen, for the *learning_rate* parameter 10 random values from uniform distribution from 0.01 to 0.6 are chosen, the *max_depth* parameter varies from 3 to 9 with step 1 (overall 7 values), for the *subsample* parameter 10 random values from uniform distribution from 0.3 to 0.9 are chosen, and finally, the *min_child_weight* parameter varies as: (2, 3, 4). The combination of all the values of each hyper-parameter for XGBoost gives overall 210000 cases. Training time for XGBoost was slightly larger compare to Random Forest for the cases described above.

5.3. ColorNet - WERegNet

The architecture of ColorNet remained untouched except for the final layer, that is changed from *softmax* to *linear* activation function so as the network has the ability to provide regression predictions. Also, the input shape of the images was adapted to 224 × 224 × 3 resolution, same as WERegNet. Both deep learning networks are tested on initial and augmented data. For the initial data, the re-scale parameter is 1/255 and the validation split is set at 20% (training 80% - validation 20%). For data augmentation, the width-shift (or horizontal shift) approach is chosen, with the internal parameter value set at 0.25, and with re-scale and validation split to be same as in the initial case. Width-shift and validation split are re-calculated for each epoch so as to provide more robust models.

Convolutional neural networks are able to learn features that are invariant to their location in the image. Nevertheless, augmentation can further aid in this transformation invariant approach to learning and can aid the model in learning features that are also invariant to transforms such as left-to-right to top-to-bottom ordering, illumination levels in photographs, and more. Data augmentation is only applied on the training set and not on the validation and test sets [56].

The ColorNet and WERegNet networks are trained for 1,000 epochs, with the callback that the best model is saved when it appears. Training time was up to 18 h for ColorNet and around 15 h for WERegNet for the cases described above.

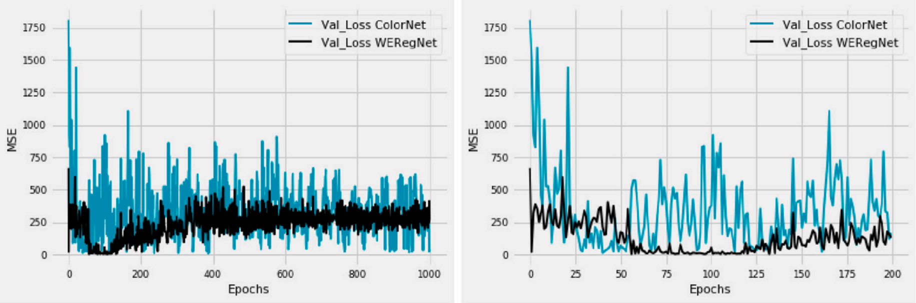
6. Simulation results

6.1. Random Forest

The optimal hyper-parameters for the case where no data augmentation took place are given below: (a) number of trees in the forest: *668*, (b) maximum number of features considered for splitting a node: *‘auto’*, (c) maximum number of levels in each decision tree: *110*, (d) minimum number of data points placed in a node before the node is split: *2*, (e) minimum number of data points allowed in a leaf node: *2* and (f) bootstrap method for sampling data points, with or without replacement: *‘False’*. Moreover, the optimal hyper-parameters for the case where data augmentation took place are given below: (a) number of trees in the forest: *878*, (b) maximum number of features considered for splitting a node: *‘auto’*, (c) maximum number of levels in each decision tree: *70*, (d) minimum number of data points placed in a node before the node is split: *2*, (e) minimum number of data points allowed in a leaf node: *5* and (f) bootstrap method for sampling data points, with or without replacement: *‘True’*. Table 1 provides the simulation results of MSE, RMSE and *std*-NRMSE for validation and test set for the optimum hyper-parameters selected by the grid search training session.

Table 2
MSE, RMSE, and *std*-NRMSE for validation and test set of XGBoost for optimal hyper-parameters set.

Data augmentation: not applied			
	MSE	RMSE	*std*-NRMSE
Validation set	0.1427	0.3778	0.0337
Test set	48.3125	6.9507	1.1749
Data augmentation: applied			
	MSE	RMSE	*std*-NRMSE
Validation set	9.9444	3.1535	0.3329
Test set	25.9625	5.0953	0.8613



(a) (b)

Fig. 3. Validation loss for: (a) 1000 epochs for ColorNet and WEReNet and (b) for the first 200 epochs where the best model loss is achieved for both architectures.

Table 3
MSE, RMSE, and *std*-NRMSE for validation and test set of ColorNet for best model.

Data augmentation: not applied			
	MSE	RMSE	*std*-NRMSE
Validation set	0.0873	0.2955	0.0264
Test set	28.87	5.0862	0.8597
Data augmentation: applied			
	MSE	RMSE	*std*-NRMSE
Validation set	27.6485	5.2582	0.4695
Test set	20.483	4.5258	0.7650

6.2. XGBoost

The optimal hyper-parameters for the case where no data augmentation took place are given below: (a) number of trees (or rounds) in an XGBoost model: *283*, (b) learning in the gradient boosting model, also called shrinkage or eta in XGBoost documentation: *0.0412*, (c) maximum depth of a tree: *6*, (d) subsample ratio of the training instances: *0.8819* and (e) minimum number of instances needed to be in each node (in regression tasks): *3*. Moreover, the optimal hyper-parameters for the case where data augmentation took place are given below: (a) number of trees (or rounds) in an XGBoost model: *645*, (b) learning in the gradient boosting model, also called shrinkage or eta in XGBoost documentation: *0.1341*, (c) maximum depth of a tree: *3*, (d) subsample ratio of the training instances: *0.6036* and (e) minimum number of instances needed to be in each node (in regression tasks): *4*. Table 2 provides the simulation results of MSE, RMSE and *std*-NRMSE for validation and test set for the optimal hyper-parameters selected by the grid search training session.

6.3. ColorNet - WERegNet

Fig. 3 shows the validation loss (MSE) of ColorNet and WERegNet for 1000 (see Fig. 3(a)) and first 200 (see Fig. 3(b)) epochs of training, when data augmentation took place. One can see that both architectures have converging paths and the minimum validation loss is achieved almost in the beginning of the training. For the record, the validation loss of the best model for the ColorNet is 6.95 and for WERegNet is 4.00. The same hold for the case when data augmentation did not take place.

Tables 3 and 4 provide the simulation results of MSE, RMSE and *std*-NRMSE for validation and test set for the best model selected after the training session of 1000 epochs, for ColorNet and WERegNet, respectively.

Table 4
MSE, RMSE, and *std*-NRMSE for validation and test set of WERegNet for best model.

Data augmentation: not applied			
	MSE	RMSE	*std*-NRMSE
Validation set	0.0789	0.2810	0.0251
Test set	24.62	4.9618	0.8387
Data augmentation: applied			
	MSE	RMSE	*std*-NRMSE
Validation set	26.2568	5.1241	0.4576
Test set	17.597	4.1949	0.7090

7. Discussion

Below, simulation results are presented and a comparison of the four methods performance is made for the cases when data augmentation (a) is not applied and (b) is applied on initial dataset and only for the training set. Except from the performance of the four methods on textile whiteness estimation problem, our aim is to show the usefulness of data augmentation on the analysis of problems with low-volume data. Thus, for the case where data augmentation was not applied on training set, *std*-NRMSE for the validation set is: 0.0197, 0.0251, 0.0264 and 0.0337 for Random Forest, WERegNet, ColorNet and XGBoost, and for the test set the value of *std*-NRMSE is: 0.8246, 0.8387, 0.8597 and 1.1749 for Random Forest, WERegNet, ColorNet and XGBoost, respectively. However, in the case where data augmentation was applied on training set, *std*-NRMSE for the validation set is: 0.3329, 0.4205, 0.4576 and 0.4695 for XGBoost, Random Forest, WERegNet and ColorNet, but for the test set the value of *std*-NRMSE is: 0.6745, 0.7090, 0.7650 and 0.8613 for Random Forest, WERegNet and ColorNet and XGBoost, respectively. From the results presented above, it is clear that the application of data augmentation on the training set improved the performance of all approaches on the test set. Taking under consideration the results of deep learning architectures and boosting techniques, it is clear that Random Forest outperforms all the other approaches regarding whiteness estimation on test set, with only WERegNet following closely. It is worth mentioning that deep learning architectures and boosting machine learning approaches tested for textile whiteness estimation have been evaluated for classification as well. The ranking of the four approaches based on F-measure, regardless of the use or not of data augmentation is similar as in the regression case.

8. Conclusion and summary

This work provides evidence that supervised machine learning regressors are capable of successfully estimating textile whiteness under various lighting conditions, thus enabling the timely anomaly detection in the quality of the produced textiles. The comparative analysis of boosting machine learning techniques, Random Forest and XGBoost, and 2DCNN architectures, ColorNet and the proposed WERegNet, shows that WERegNet has better performance on test set compared to ColorNet and XGBoost and it is close enough to Random Forest, based on the standard deviation normalized root mean square error (*std*-NRMSE) evaluation metric. The paper also demonstrated the remarkable reduction of the test error thanks to data augmentation during pre-processing. To the best of our knowledge, this is the first work explicitly estimating textile whiteness using machine learning methodologies. Regarding the performance of the regression models on textile whiteness estimation, in future work there is a plan to explore additional hyper-parameters settings on WERegNet so as to improve further the performance of the proposed WERegNet by a more thorough optimization of its hyper-parameters. Two approaches will be used: the grid search that exhaustively considers all parameter combinations from a parameter space and the randomized search that samples a given number of candidate parameters from a parameter space with a specified distribution. Moreover, computational complexity is comparable among all architectures, taking into account the number of tested hyper-parameter combinations. Thus, considering the promising results of Random forest and WERegNet, these approaches are going to be applied also to other industrial problems, such as defect detection and quality monitoring, using e.g. thermal images as input.

CRediT authorship contribution statement

Thanasis Vafeiadis: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Changes and corrections before the final submission. **Nikolaos Kolokas:** Conceptualization, Methodology, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Changes and corrections before the final submission. **Nikolaos Dimitriou:** Conceptualization, Methodology, Formal analysis, Data curation, Writing – review & editing, Visualization, Changes and corrections before the final submission. **Angeliki Zacharaki:** Conceptualization, Writing – review & editing, Changes and corrections before the final submission. **Murat Yildirim:** Resources, Writing – review & editing, Changes and corrections before the final submission. **Habibe Gülben Selvi:** Resources, writing – review & editing, Changes and corrections before the final submission. **Dimosthenis Ioannidis:** Supervision, Funding acquisition, Changes and corrections before the final submission. **Dimitrios Tzovaras:** Project administration, Funding acquisition, Changes and corrections before the final submission.

Acknowledgment

This work has been partially supported by the European Commission through the project HORIZON 2020 - INNOVATION ACTIONS (IA)-869884-RECLAIM. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

References

[1] Y. Ohno, CIE Fundamentals for Color Measurements, IS & T NIP16 Conference, 2000.
[2] https://www.xrite.com/de/service-support/whiteness_index.
[3] A. Athalye, Bleach clean up, 2014, https://www.fibre2fashion.com/industry-article/7349/bleach-clean-up.
[4] A.M. Collier, A Handbook of Textiles, Pergamon Press, 1970.
[5] A. Kumar, R. Choudhury, Principles of Textile Finishing, Elsevier Science, 2017.
[6] http://cie.co.at/about-cie.
[7] J. Schanda, 3. CIE colorimetry, in: Colorimetry: UnderstandIng the CIE System, Wiley, 2007, pp. 25–78, http://dx.doi.org/10.1002/9780470175637.ch3.
[8] K.D. Mielenz, J.J. Hsia, J.R. Moore, A.R. Robertson, H. Terstiege, J.F. Verrill, ISO 11664-2:2007(en) Colorimetry — Part 2: CIE standard illuminants, 2007, https://www.iso.org/obp/ui/#iso:std:52496:en.
[9] E. Allen, D.H. Alman, C.S. McCamy, Y. Nayatani, N. Ohta, J. Schanda, F.T. Simon, D. Strocka, H. Terstiege, R. Hirschler, J.T.C. van Kemenade, J.C. Zwinkels, A Method for Assessing the Quality of Daylight Simulators for Colorimetry, CIE, ISBN: 978-92-9034-051-5, 1999.
[10] Y.-M. Lam, J.H. Xin, Evaluation of the quality of different D65 simulators for visual assessment, Color Res. Appl. 27 (4) (2002) 243–251, http://dx.doi.org/10.1002/col.10061.
[11] E.C. Carter, J.D. Schanda, R. Hirschler, S. Jost, M.R. Luo, M. Melgosa, Y. Ohno, M.R. Pointer, D.C. Rich, F. Viénot, L. Whitehead, J.H. Wold, Colorimetry, fourth ed., CIE, 2018, http://dx.doi.org/10.25039/TR.015.2018.
[12] https://www.ibm.com/blogs/client-voices/quality-control-automation-leads-to-industry-4-0/.
[13] N. Dimitriou, L. Leontaris, T. Vafeiadis, D. Ioannidis, T. Wotherspoon, G. Tinker, D. Tzovaras, Fault diagnosis in microelectronics attachment via deep learning analysis of 3D laser scans, IEEE Trans. Ind. Electron. 67 (7) (2019) 5748–5757, http://dx.doi.org/10.1109/TIE.2019.2931220.
[14] N. Dimitriou, L. Leontaris, T. Vafeiadis, D. Ioannidis, T. Wotherspoon, G. Tinker, D. Tzovaras, A deep learning framework for simulation and defect prediction in industrial processes, Simul. Model. Pract. Theory 100 (2020) 102063, http://dx.doi.org/10.1016/j.simpat.2019.102063.
[15] T. Kotsiopoulos, L. Leontaris, N. Dimitriou, D. Ioannidis, F. Oliveira, J. Sacramento, S. Amanatiadis, G. Karagiannis, K. Votis, D. Tzovaras, P. Sarigaiannidis, Deep multi-sensorial data analysis for production monitoring in hard metal industry, Int. J. Adv. Manuf. Technol. 100 (2020) 102063, http://dx.doi.org/10.1007/s00170-020-06173-1.
[16] J. Villalba-Diez, D. Schmidt, R. Gevers, J. Ordieres-Mere, M. Buchwitz, W. Wellbrock, Deep learning for industrial computer vision quality control in the printing industry 4.0, Sensors (2019) http://dx.doi.org/10.3390/s19183987.
[17] R. Ozdemir, M. Koc, A quality control application on a smart factory prototype using deep learning methods, in: 14th IEEE International Conference on Computer Sciences and Information Technologies, 2019, http://dx.doi.org/10.1109/STC-CSIT.2019.8929734.
[18] K. Papachristou, N. Dimitriou, A. Drosou, G. Karagiannis, D. Tzovaras, Realistic texture reconstruction incorporating spectrophotometric color correction, in: 25th IEEE International Conference on Image Processing, 2018, http://dx.doi.org/10.1109/ICIP.2018.8451323.
[19] J. Villalba-Diez, M. Molina, J. Ordieres-Mere, S. Sun, D. Schmidt, D. Tzovaras, Geometric deep lean learning: Deep learning in industry 4.0 cyber–physical complex networks, Sensors 20 (3) (2020) 763, http://dx.doi.org/10.3390/s20030763.
[20] T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, K.Ch. Chatzisavvas, A comparison of machine learning techniques for customer churn prediction, Simul. Model. Pract. Theory 55 (2015) 1–9, http://dx.doi.org/10.1016/j.simpat.2015.03.003.
[21] Y. Shin, Application of boosting regression trees to preliminary cost estimation in building construction projects, Comput. Intell. Neurosci. 2015 (2015) http://dx.doi.org/10.1155/2015/149702.
[22] D. Arditi, T. Pulket, Predicting the outcome of construction litigation using boosted decision trees, J. Comput. Civ. Eng. 19 (4) (2005) 387–393.
[23] Y. Shin, T. Kim, H. Cho, K.I. Kang, A formwork method selection model based on boosted decision trees in tall building construction, Autom. Constr. 23 (2012) 47–54.
[24] V.V. Gavrishchaka, Boosting-based frameworks in financial modeling: Application to symbolic volatility forecasting, in: Advances in Econometrics, Emerald Publishing Limited, UK, 2006.
[25] G. Anders, Control of whites: The cibanoid white scale, J. Society Dye. Colour. 84 (2) (1968) 125–132, http://dx.doi.org/10.1111/j.1478-4408.1968.tb02807.x.
[26] T.N. Ezhova, G.A. Reshetnjak, Spectrocolorimetric method for instrumental estimation of whiteness, in: Proc. SPIE 2161, CIS Selected Papers: Photometry, 1993, http://dx.doi.org/10.1117/12.166378.
[27] H. Uchida, A New Whiteness Formula, 23, (4) Wiley Online Library, 1998, pp. 202–209, http://dx.doi.org/10.1002/(SICI)1520-6378(199808)23:4<202::AID-COL4>3.0.CO;2-S.
[28] X. Lv, Y.Z. Wang, M. Wei, M.R. Luo, New metrics for evaluating whiteness of fluorescent samples, color and imaging conference, in: 27th Color and Imaging Conference Final Program and Proceedings, 2019, pp. 247–251, http://dx.doi.org/10.2352/issn.2169-2629.2019.27.44.
[29] Y. Wang, X. Lv, M.R. Luo, Testing performance of whiteness formulas, in: Advances in Graphic Communication, Printing and Packaging - Lecture Notes in Electrical Engineering, vol. 543, 2019, pp. 3–8, http://dx.doi.org/10.1007/978-981-13-3663-8_1.
[30] M.M. Pérez, L.J. Herrera, F. Carrillo, O.E. Pecho, D. Dudea, C. Gasparik, R. Ghinea, A.D. Bona, Whiteness difference thresholds in dentistry, Dent. Mater. 35 (2) (2019) 292–297, http://dx.doi.org/10.1016/j.dental.2018.11.022.
[31] K.J. Nieuwenhuis, Whiteness and fluorescence of fabrics, J. Amer. Oil Chem. Soc. 45 (1) (1968) 37–42, http://dx.doi.org/10.1007/BF02679044.
[32] G.P. Shesternina, I.Kh. Raskina, L.I. Belen'kii, Use of new soviet colorimetric equipment in textile-material quality evaluation, 1985, https://www.google.com/search?client=firefox-b-d&q=Use+of+new+soviet+colorimetric+equipment+in+textile-material+quality+evaluation.
[33] W. Czajkowski, J. Paluszkiewicz, R. Stolarski, M. Kaźmierska, E. Grzesiak, Synthesis of reactive UV absorbers, derivatives of monochlorotriazine, for improvement in protecting properties of cellulose fabrics, Dye. Pigment. 71 (3) (2006) 224–230, http://dx.doi.org/10.1016/j.dyepig.2005.07.004.
[34] A.K.R. Choudhury, Textile Preparation and Dyeing, Science Publishers, 2006.
[35] R. Jafari, S.H. Amirshahi, A comparison of the CIE and uchida whiteness formulae as predictor of average visual whiteness evaluation of textiles, Text. Res. J. 77 (10) (2007) 756–763, http://dx.doi.org/10.1177/0040517507080688.
[36] S. Kalantzi, D. Mamma, P. Christakopoulos, D. Kekos, Effect of pectate lyase bioscouring on physical, chemical and low-stress mechanical properties of cotton fabrics, Bioresour. Technol. 99 (17) (2008) 8185–8192, http://dx.doi.org/10.1016/j.biortech.2008.03.020.
[37] S. Kalantzi, D. Mamma, E. Kalogeris, D. Kekos, Improved properties of cotton fabrics treated with lipase and its combination with pectinase, Fibres Text. East. Eur. 18 (2010) 86–92, [5(82)].

[38] M. Tutak, O. Demiryürek, Ş. Bulut, D. Haroğlu, Analysis of the CIE whiteness and whiteness tint of optically whitened cellulosic fabrics, Text. Res. J. 81 (1) (2011) 58–66, http://dx.doi.org/10.1177/0040517510380111.

[39] H. Jung, T. Sato, Comparison between the color properties of whiteness index and yellowness index on the CIELAB, Text. Color. Finish. 25 (4) (2013) 241–246, http://dx.doi.org/10.5764/TCF.2013.25.4.241.

[40] E.S. Abdel-Halim, S.S. Al-Deyab, One-step bleaching process for cotton fabrics using activated hydrogen peroxide, Carbohydr. Polymers 92 (2) (2013) 1844–1849, http://dx.doi.org/10.1016/j.carbpol.2012.11.045.

[41] L.G.M. Silva, F.C. Moreira, A.A.U. Souza, S.M.A.G.U. Souza, R.A.R. Boaventura, V.J.P. Vilar, Chemical and electrochemical advanced oxidation processes as a polishing step for textile wastewater treatment: A study regarding the discharge into the environment and the reuse in the textile industry, J. Cleaner Prod. 198 (2018) 430–442, http://dx.doi.org/10.1016/j.jclepro.2018.07.001.

[42] T. Hareem, F. Arooj, S. ur R. Kashif, Z. Farooq, Economic viability of pilot-scale application of ozone in cotton bleaching with multiple reuse of water, Ozone: Sci. Eng. 41 (2) (2018) 197–203, http://dx.doi.org/10.1080/01919512.2018.1509204.

[43] A. Singh, L.M. Varghese, B. Battan, A.K. Patra, R.P. Mandhan, R. Mahajan, Eco-friendly scouring of ramie fibers using crude xylano-pectinolytic enzymes for textile purpose, Environ. Sci. Pollut. Res. 27 (2019) 6701–6710, http://dx.doi.org/10.1007/s11356-019-07424-9.

[44] S. Kalantzi, D. Kekos, D. Mamma, Bioscouring of cotton fabrics by multienzyme combinations: application of Box–Behnken design and desirability function, Cellulose 26 (2019) 2771–2790, http://dx.doi.org/10.1007/s10570-019-02272-9.

[45] S. Udhayamarthandan, J. Srinivasan, Integrated enzymatic and chemical treatment for single-stage preparation of cotton fabrics, Text. Res. J. 89 (19–20) (2019) 3937–3948, http://dx.doi.org/10.1177/0040517518824845.

[46] A. Mahapatra, S. Patil, A. Arputharaj, V.D. Gotmare, P.G. Patil, Effect of textile softeners on BTCA treated cotton fabric, Indian J. Fibre Text. Res. (IJFTR) 45 (1) (2020) 96–101.

[47] J. Lou, J. Yuan, Q. Wang, P. Wang, J. Xu, Y. Yu, X. Fan, Preparation and application of polyaldehyde trehalose as a new hydrophilic anti-crease finishing agent for cotton fabric, Text. Res. J. (2020) 1–12, http://dx.doi.org/10.1177/0040517520942964.

[48] L. Breiman, Random forests - random features, Berkeley: Statistics Department, University of California, Retrieved from ftp://ftp.stat.berkeley.edu/pub/users/breiman.

[49] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[50] https://cs231n.github.io/convolutional-networks/.

[51] R.F. Rachmadi, I.K.E. Purnama†, Vehicle Color Recognition using Convolutional Neural Network, arXiv:1510.07391.

[52] S.A. Otto, How to normalize the RMSE [Blog post], 2019, https://www.marinedatascience.co/blog/2019/01/07/normalizing-the-rmse/.

[53] S.A. Otto, M. Kadin, M. Casini, M.A. Torres, T. Blenckner, A quantitative framework for selecting and validating food web indicators, Ecol. Indic. 84 (2018) 619–631, http://dx.doi.org/10.1016/j.ecolind.2017.05.045.

[54] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction To Statistical Learning - with Applications in R, Springer, New York, 2013.

[55] T.J. Hastie, R.J. Tibshirani, Generalized Additive Models, Chapman & Hall / CRC, Boca Raton, 1990.

[56] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html.