energies

Article

Method of Biomass Discrimination for Fast Assessment of Calorific Value

Jarosław Gocławski [1,*], Ewa Korzeniewska [2], Joanna Sekulska-Nalewajko [1], Paweł Kiełbasa [3] and Tomasz Dróżdż [3]

[1] Institute of Applied Computer Science, Lodz University of Technology, 18 Stefanowskiego Street, 90-537 Lodz, Poland; joanna.sekulska-nalewajko@p.lodz.pl
[2] Institute of Electrical Engineering Systems, Lodz University of Technology, 18 Stefanowskiego Street, 90-537 Lodz, Poland; ewa.korzeniewska@p.lodz.pl
[3] Faculty of Production and Power Engineering, University of Agriculture in Krakow, Balicka Av. 116B, 30-149 Cracow, Poland; pawel.kielbasa@urk.edu.pl (P.K.); tomasz.drozdz@urk.edu.pl (T.D.)
* Correspondence: jaroslaw.goclawski@p.lodz.pl

Abstract: Crop byproducts are alternatives to nonrenewable energy resources. Burning biomass results in lower emission of undesirable nitrogen and sulfur oxides and contributes no significant greenhouse effect. There is a diverse range of energy-useful biomass, including in terms of calorific value. This article presents a new method of discriminating biomass, and of determining its calorific value. The method involves extracting the selected texture features on the surface of a briquette from a microscopic image and then classifying them using supervised classification methods. The fractal dimension, local binary pattern (LBP), and Haralick features are computed and then classified by linear discrimination analysis (*LDA*). The discrimination results are compared with the results obtained by random forest (*RF*) and deep neural network (*DNN*) type classifiers. This approach is superior in terms of complexity and operating time to other methods such as, for instance, the calorimetric method or analysis of the chemical composition of elements in a sample. In the normal operation mode, our method identifies the calorific value in the time of about 100 s, i.e., 90 times faster than traditional combustion of material samples. In predicting from a single sample image, the overall average accuracy of 95% was achieved for all tested classifiers. The authors' idea to use ten input images of the same material and then majority voting after classification increases the discrimination system accuracy above 99%.

Keywords: biofuel; biomass; calorific value; image analysis; textural features; random forest; linear discrimination; deep neural network; principal component analysis

check for updates

Citation: Gocławski, J.; Korzeniewska, E.; Sekulska-Nalewajko, J.; Kiełbasa, P.; Dróżdż, T. Method of Biomass Discrimination for Fast Assessment of Calorific Value. *Energies* **2022**, *15*, 2514. https://doi.org/10.3390/en15072514

Academic Editor: Rafał Longwic

Received: 17 January 2022
Accepted: 16 March 2022
Published: 29 March 2022

1. Introduction

On 14 July 2021, the European Commission adopted a package of legislative proposals that set European Union policy goals for climate and energy adjustment. Member states are obliged to cut their net greenhouse gas emissions by at least 55% by 2030 compared to 1990 [1]. Increasingly stringent regulations to protect the environment and air are spurring the search for alternative biofuels, characterized by low emissions of atmospheric pollutants and the highest possible energy efficiency.

Plant biomass is the basis for the production of biofuels, which can substitute or supplement diesel, petrol, and other fossil fuels used for transport, heating, and other applications. There are three categories of biofuels: first generation, second generation, and third generation. First-generation biofuels are produced primarily from plants and plant byproducts, such as vegetable oils, grains, and sugar cane. According to the recommendations of the EC, energy production from these raw materials cannot exceed 7%. Raw materials used for the production of second-generation biofuels include cellulose, crop residues, and forestry residues. Third-generation biofuels are produced by processing autotrophic aquatic organisms, such as algae [2,3]. According to the European Commission, a minimum of 3% of energy should be produced by second- and third- generation biofuels.

Because of their availability and stability of supply, biofuels obtained from crop byproducts should be preferenced [4]. The use of second-generation biofuels also helps to reduce agricultural waste and does not require building expensive production lines, in contrast to other energy sources. One of the second-generation biofuels is biomass obtained from straw. According to cause-effect models, in Poland alone the economic potential for energy production using straw will amount to 5.4 million tons (1.8 Mtoe) in 2030 [3]. Research by Lantz and co-authors [5] shows that greenhouse gas emissions from straw and cereal biofuels are significantly lower than those from fossil fuels, with straw-based fuels being the least harmful. The use of straw as a raw material for biofuel also mitigates climate change via nitrogen fixation [3].

Effective use of straw for energy production requires careful identification of various technical and economic parameters. Among the most important features of fuel are its calorific value and thermal efficiency [6,7]. Calorific value is defined as the amount of heat released during the complete combustion of 1 kg of biomass. Calorific value is the most important determinant of combustibility. It ranges from 8.6 to 44 MJ/kg, depending on the fuel type [8]. Notwithstanding its many benefits compared to other fossil fuels, straw has a relatively low calorific value. Its calorific value depends on the type of straw [9]. Straw used for energy purposes comes most often from wheat, barley, rape, and maize. The dry matter calorific values of these crops are as follows: 17.5–17.7 MJ/kg for wheat straw [10]; 15.7 MJ/kg for barley straw [11]; 15.3 MJ/kg for rapeseed straw [12]; 16.1–20.9 MJ/kg for maize straw [12–14]. The noticeable differences in these values, in particular between straw types, may be caused by sample heterogeneity and lack of consistency in calorimetric procedures [13]. Nonetheless, Pordesimo and co-authors noticed that the energy content of maize straw remains fairly constant, regardless of the fraction and harvest time [13]. Different straws, such as wheat or corn straw, can be mixed with other combustible wastes, such as wood chips or crop-processing residues (grain hull, dry pomace). The resulting mixture can be used for the production of fuel rods or block-shaped fuel. Mixing a certain percentage of a binder in the form of another biomass waste with the crushed straw has been found to increase the calorific value of the briquettes and decrease the energy required for compression. The effects of the binder are related to its type and the percentage ratio of the components [15,16].

The calorific value of highly variable solid fuels in terms of their chemical-physical compositions can be measured directly using a calorimeter bomb [17,18]. In the combustion process, the water vapor resulting from the oxidation of hydrogen is condensed and cooled to the temperature of the main element—a bomb that releases about 600 kcal for every kilogram of condensed water vapor, and which is usually released as vapor in the exhaust gas. Measurement using a calorimetric bomb is complicated and time-consuming. Moreover, it requires configuration procedures, as well as measurements and calculations of partial results. The calorific value can also be estimated from the element composition [19], although that is also a complex process. Conventional analysis to determine the basic composition of fuels requires the use of modern laboratory equipment [20]. To the best knowledge of the authors, no alternative methods have been developed, especially ones based on image analysis, which are not so costly in terms of time and equipment.

Pressed biomaterials are composed of various components that can be distinguished visually to some extent by macro- or microscopic techniques. The material surface presents a set of textual features that can be assessed visually, and which together with color can be subjectively related to the composition and characteristics of the surface morphology (e.g., coarse, rough, and complex surfaces or smooth and uniform surfaces). However, the resulting microstructures are often visually similar; therefore, experts have difficulty recognizing their differences. The material can be subjected to computer analysis, which transforms the texture image into a feature vector using a bank of filters, followed by nonlinearity and smoothing steps for the purposes of further classification. Although there is no generally accepted definition of texture in computer vision, it has been established that a texture image or region obeys certain statistical properties and exhibits repeated structures. Because these properties may be linked to the structure of the studied object, texture is currently being correlated with the chemical and physical properties of biomaterials [21]. Typing and distinguishing objects and materials based on texture is usually possible using linear classifiers [22], provided that the correct set of features is selected, or that a complex set of data is initially provided for further decomposition, as in the case of the PLS-DA classifier [23]. In recent years, there has been growing interest in using image processing techniques employing artificial intelligence classifiers to recognize textures and distinguish between materials with only minor differences in terms of composition and processing [24]. Commercial uses for texture learning models are emerging: smartphone applications that use neural networks have been developed around the world to determine the calorific values of products and meals [25–27].

In this study, we propose an image analysis system for biomass discrimination based on sample textures visible at the microscale. The motivation was to design and build a tool that is capable of recognizing biofuel types for calorific testing purposes. There are no reports in the literature on the use of the method of artificial intelligence to determine the calorific value of biomass prepared in the form of pellets. Due to the innovative approach to the subject, the described method was submitted to the Patent Office in order to protect intellectual property. Detecting textural changes is essential for establishing a successful link between the composition and properties of biofuels. We use a dataset of five biofuel types, which differ in terms of the composition of their crop byproducts and calorific values. Despite the different compositions of the biofuels, they are very similar in macroscopic and microscopic appearance. This allowed us to test the ability of different linear and nonlinear texture classifiers to discern the texture differences between similar materials. The presented approach does not require advanced laboratory equipment and is a much faster method than previously described.

2. Materials and Methods

2.1. Biomass Samples

In the presented method, the calorific values of biofuels are assessed indirectly by texture recognition. The textures of the biomass samples are assigned to specific classes with known heating parameters.

Five different types of materials were prepared from shredded straw and grains. The shredded straw and grains were mixed in appropriate proportions (Table 1) and then pressed using a hydraulic press with a pressure of 3 tonnes. This procedure is equivalent to the process of biomass pelletization used in industry. Thickening is performed to make the properties of straw biomass closer to those of coal, since the energy density of biomass increases as a result of compression. Shapes and geometric dimensions are made uniform, and the moisture content is lowered, which makes this type of processed biomass the basic biofuel for commercial power engineering.

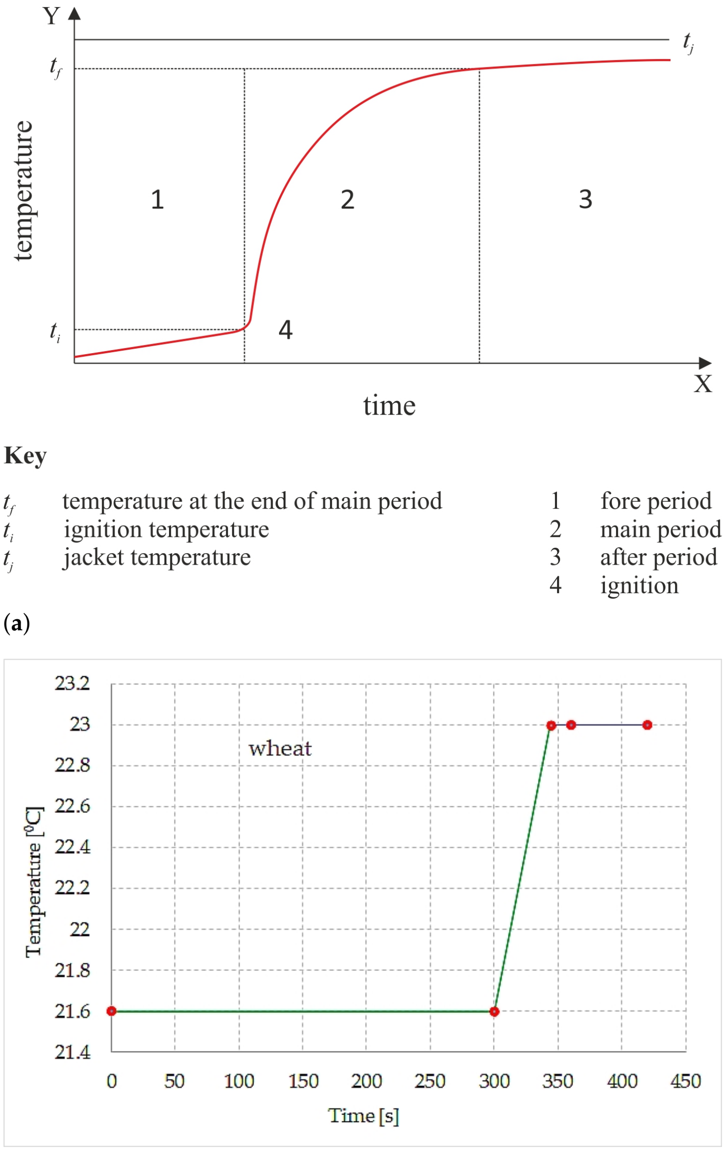
Table 1. Sample data for the tested biofuel materials.

Biofuel Type	Material Components	Number of Images	Heating Value [MJ/kg]	Combustion Heat [MJ/kg]
1	50% wheat, 50% wheat straw	100	15.6	17.15
2	50% wheat, 50% triticale straw	100	15.6	16.90
3	50% wheat straw, 50% triticale straw	100	15.5	16.80
4	100% wheat	100	15.8	17.20
5	100% wheat straw	100	15.4	17.10

The combustion heat of the tested samples was determined with the use of the calorimetric bomb combustion method, according to the EN ISO 18125:2017 Solid biofuels—

Determination of calorific value standard [28]. The Kl-12Mn bomb calorimeter was used, which is capable of determining changes in the temperature of liquid in the calorimetric vessel with an accuracy of 0.001 °C. Due to the methodological requirements and the low density of sawdust, the material was pressed using a hand press to form a pellet before the measurements were taken.

The calorimeter measures and then records automatically three characteristic temperature points and three basic periods in the time-temperature course of the combustion process, as depicted in Figure 1. The measurement cycle of the calorimeter can be completed in approximately 15 min for a single biofuel sample. The results for the calculated heat of combustion are then averaged over 10 measurements, which takes 2.5 h.



(**a**)

(**b**)

Figure 1. Time-temperature curve of the combustion process: (**a**) illustrative model of the time-temperature curve with three characteristic temperature points, t_i, t_j and t_f (explanations on the legend), according to EN ISO 18125:2017-07; (**b**) exemplary time-temperature curve obtained for 100% wheat-pressed material (biofuel type 4).

2.2. Biomass Image Acquisition and Preprocessing

Images of the biofuel samples (Figure 2) were captured using a digital camera mounted on a Delta Optical microscope (Delta Optical, Poland), then transmitted to a Personal

Computer (PC) via a Universal Serial Bus (USB) link and saved as JPEG-type disk files, as shown in Figure 3. In total, 550 sample images representing 110 of each of the 5 classes of biofuel were captured and stored. In our system, it is assumed that entering a single image takes about 10 s to locate sample, focus objective, acquire image and save as a JPEG file. One-time presentation of 100 images of a given biofuel type, necessary to associate the texture of the material with its combustion heat, requires 100 × 10 s ≈ 17 min. After such associations for all materials have been established by the machine learning algorithm, the biofuel identification requires only the input of 10 surface images.

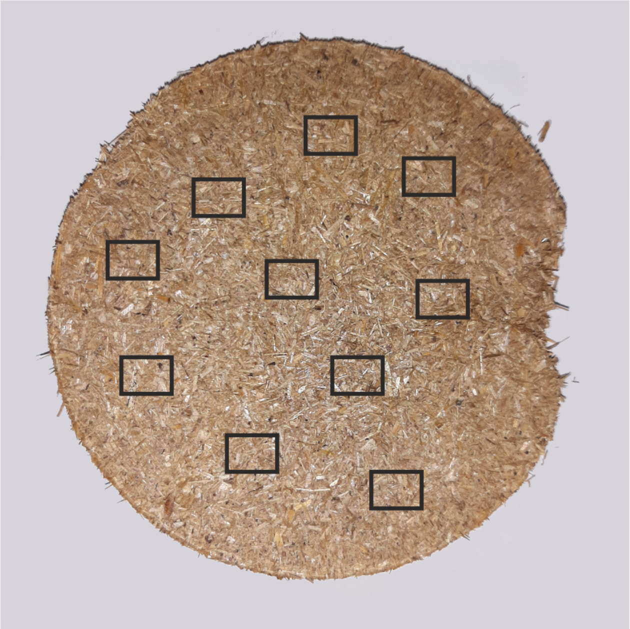


Figure 2. Sample material of 100% wheat straw (biofuel type 5).

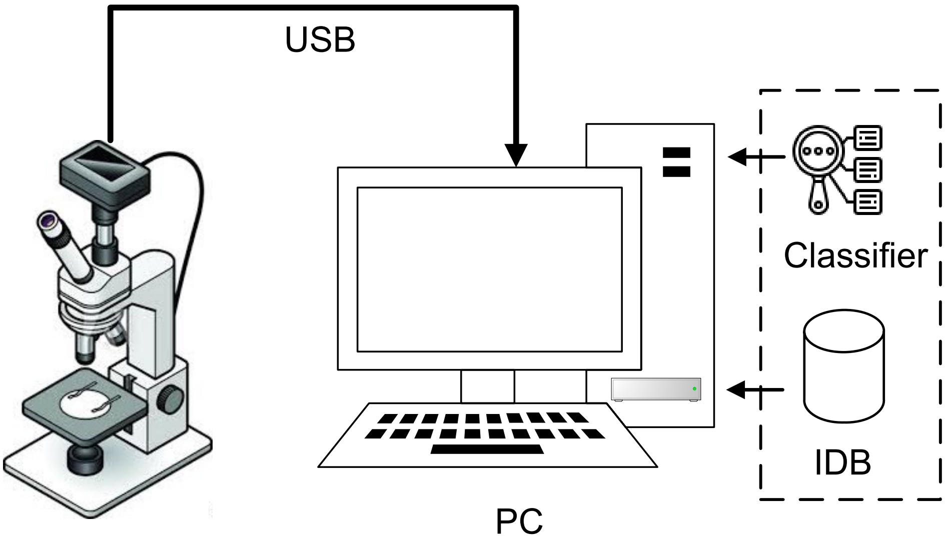


Figure 3. Block diagram of the biofuel material identification system: PC—Personal Computer; IDB—database of biofuel images for training the classifier; USB—serial connection between the digital camera of the microscope and the PC.

The PC module is equipped with an Intel(R) Core(TM) i7-7700HQ Central Processing Unit (CPU) with a 2.80 GHz clock, 16 GB Random Access Memory (RAM), and an NVIDIA GeForce GTX 1050 graphic card with 4 GB fast access RAM as a Graphic Processing Unit (GPU) for use in parallel computations. The images of known biofuel samples are collected in separate folders, composing the Image Database (IDB), and used for training the built-in classifier. The captured images are stored in a raster of 2560 × 1920 pixels, with a resolution of 9600 dpi, which corresponds to an image area of 6.77 × 5.08 mm in physical units. Software implementing the proposed method was designed in the form of a Python script executed in the Windows 10 operating system.

The software runs in two modes: the training mode and the mode of biofuel class prediction, with training required before prediction. The training can be treated as a calibration stage in the proposed biofuel measurement system. Flowcharts of the method algorithm in the two execution modes are shown in Figures 4 and 5. The image preprocessing module is marked as subprocess block 3 in both figures. It incorporates three stages, shown in Figure 6. The first stage consists of scaling down the image raster 8 times to 320 × 240 pixels, with bilinear interpolation. Scaling is performed using the Python *resize* function included in the Python *opencv* library. This operation accelerates the computation of textural features and training of the internal classifier. Equalization of uneven image lighting [29,30] is carried out by subtracting from each image color component I_c its blurred version G_c, which is with the average component intensity $\overline{I}_c$ as given in Equation (1). The blurred component G_c is obtained by Gaussian filtering with the appropriate kernel size and variance.

$$\begin{gathered} I_c' = I_c - G_c + \overline{I}_c, \\ G_c = GaussianBlur(I_c, ksize, sigma), \\ \overline{I}_c = \underset{x,y}{mean}(I_c(x,y)), \end{gathered} \tag{1}$$

where $c \in \{R, G, B\}$, $GaussianBlur$ is a Python function from the *opencv* package, $ksize = (31, 31)$ denotes the limited Gaussian kernel size, and $sigma = 15$ is the assumed Gaussian standard deviation. Conversion of the color image I to grayscale is performed by evaluating its YUV luminance according to the known formula in Equation (2) [30]

$$I_Y = 0.299 I_R + 0.587 I_G + 0.114 I_B. \tag{2}$$

2.3. Biomass Image Textural Feature Computing

To identify the type of biofuel material and therefore its caloric content, the textural features of the pressed cereals are taken into account. The feature selection and extraction modules are shown in Figure 7. To assess the texture of the biofuel, the Haralick textural features $H_1, \ldots, H_{13}$ of the image I_Y are taken into account [31]. These statistical properties, derived from the Gray Level Co-occurrence Matrix (GLCM) [31,32], are related to the intensity differences between pixels of the image I_Y at a fixed distance and thus indirectly refer to the characteristics of the biofuel surface. The Python function $mahotas.features.haralick$ was used, which returns the mean values of all H features in four directions spaced every 45° on the image plane. The function is shown as Equation (3)

$$(H_1, \ldots, H_{13}) = mahotas.features.haralick(I_Y, distance = 2, return\_mean = True), \tag{3}$$

where $distance = 2$ is the assumed fixed distance between compared pixels in all directions. We additionally used the features of fractal dimension, border mean, and border area proposed for the Segmentation-based Fractal Texture Analysis (SFTA) [33]. To compute these features, the gray-level image I_Y must be binarized using one-level Otsu thresholding [34], as described in Equation (4):

$$I_B = T_{OTSU}(I_Y). \tag{4}$$

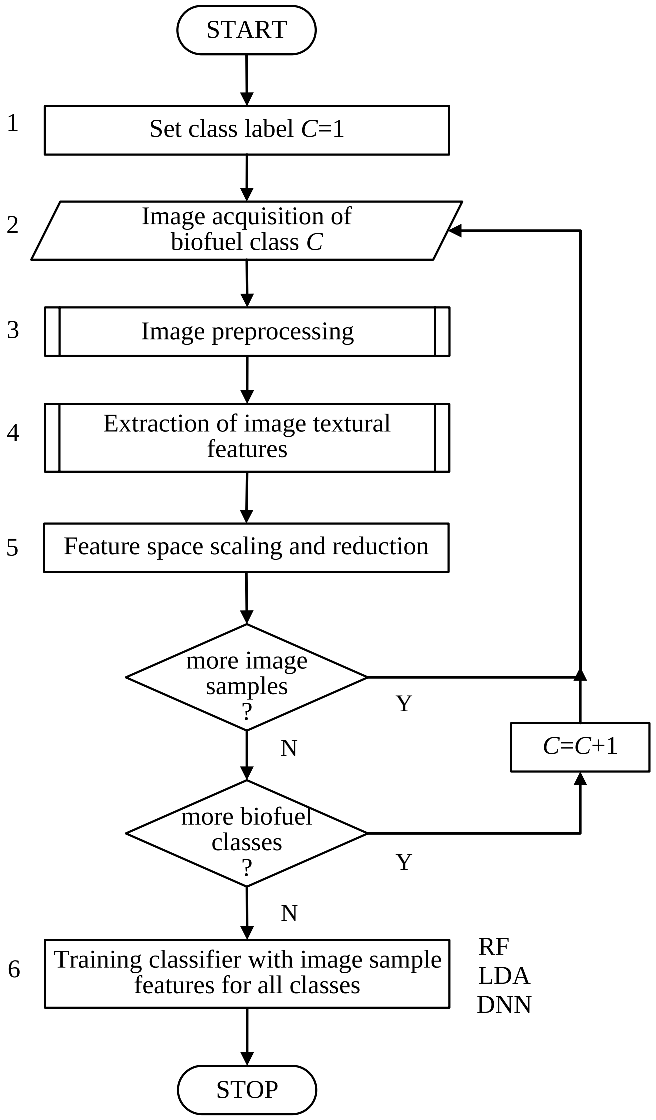

START
1
Set class label *C*=1
2
Image acquisition of biofuel class *C*
3
Image preprocessing
4
Extraction of image textural features
5
Feature space scaling and reduction
more image samples ?
Y
N
C=*C*+1
more biofuel classes ?
Y
N
6
Training classifier with image sample features for all classes
RF
LDA
DNN
STOP

Figure 4. Flowchart of the algorithm in training mode: *RF*—random forest classifier; *LDA*—linear discrimination analysis; *DNN*—deep neural network.

The fractal dimension used in SFTA is based on the box-counting (Minkowski) dimension $dim_{box}(S)$, which is applied according to the formula

$$dim_{box}(S) = \lim_{\epsilon \to 0} \frac{log N(\epsilon)}{log(1/\epsilon)}, \tag{5}$$

where ϵ denotes the side length of the sliding cube, N is the number of cubes, and S is the 2D dataset of image pixels. The dimension D is estimated by the least squares method using the linear equation $y = Dx + A$, where $y = log(N)$ and $x = log(1/\epsilon)$ [35]. To obtain the value of D, gliding boxes of size $[\epsilon_i \times \epsilon_i]$ are used. The number of boxes $N_i = N(\epsilon_i)$ containing at least one bright voxel in the border image $\Delta(I_B)$ is computed for each box of size $\epsilon_i = s, s/2, s/4, \ldots, 2$, starting with an initial size s equal to the smallest image dimension rounded down to the power of 2. The data (x_i, y_i) are computed for each box size ϵ_i, according to Equation (6)

$$y_i = log(N(\epsilon_i)), \quad x_i = log(\epsilon_i), \tag{6}$$

and then the least squares method of regression [36] allows linear approximation of the data. The feature fractal dimension D_f shown in Equation (7) represents the negative slope of the line $y(x)$.

$$D_f = -D, \tag{7}$$

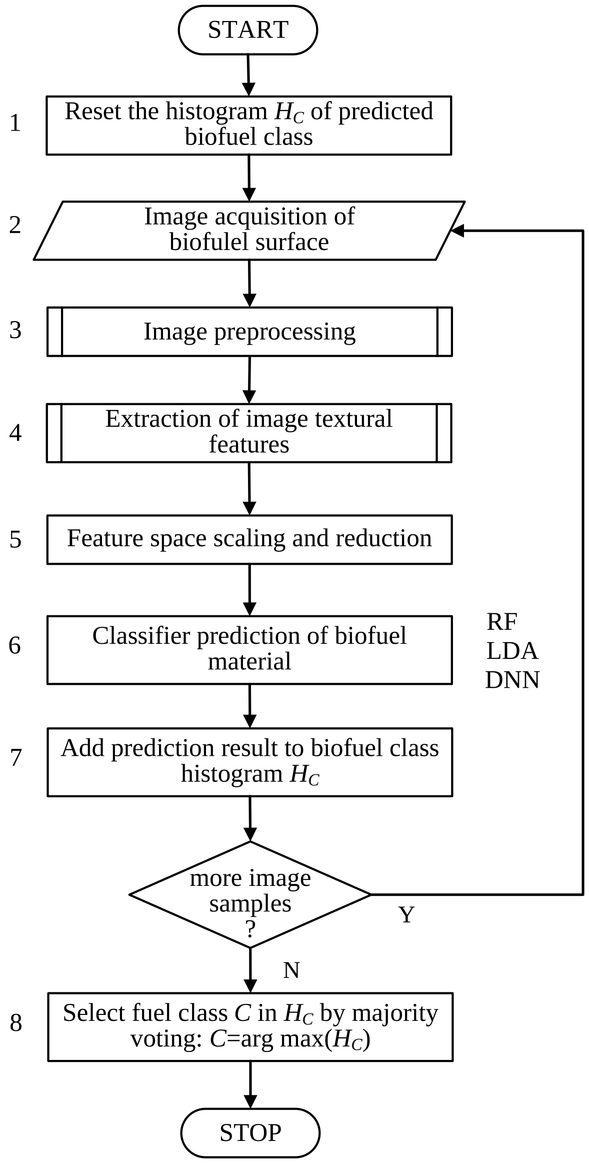

START
1
Reset the histogram H_C of predicted biofuel class
2
Image acquisition of biofulel surface
3
Image preprocessing
4
Extraction of image textural features
5
Feature space scaling and reduction
6
Classifier prediction of biofuel material
RF
LDA
DNN
7
Add prediction result to biofuel class histogram H_C
more image samples ?
Y
N
8
Select fuel class C in H_C by majority voting: C=arg max(H_C)
STOP

Figure 5. Flowchart of the algorithm in prediction mode: *RF*—random forest classifier; *LDA*—linear discrimination analysis; *DNN*—deep neural network.

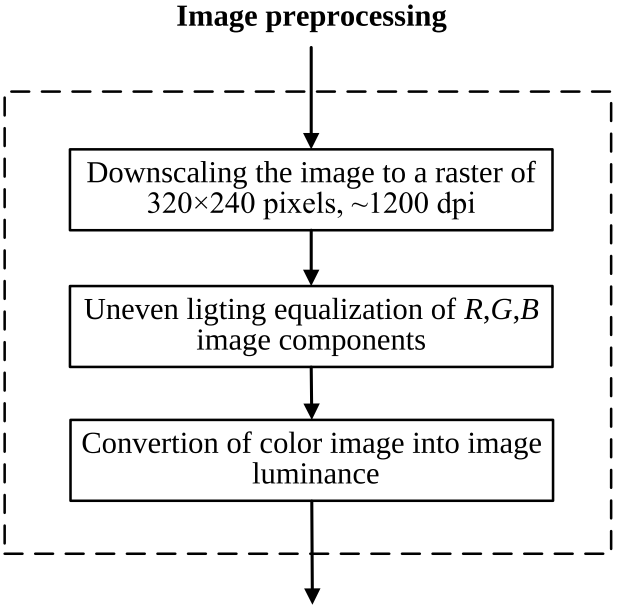


Image preprocessing
Downscaling the image to a raster of 320×240 pixels, ~1200 dpi
Uneven ligting equalization of *R*,*G*,*B* image components
Convertion of color image into image luminance

Figure 6. Flowchart of the image preprocessing stages.

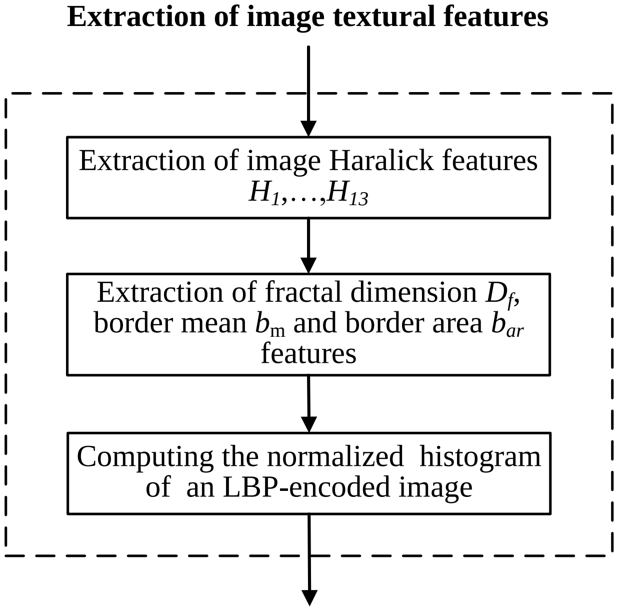

Extraction of image textural features
Extraction of image Haralick features $H_1,\ldots,H_{13}$
Extraction of fractal dimension D_f, border mean b_m and border area b_{ar} features
Computing the normalized histogram of an LBP-encoded image

Figure 7. Flowchart of the textural features selected for biofuel classification.

Other SFTA textural features are the border mean value b_m and the border area b_{ar}. The b_m is evaluated as the average brightness of the image I_Y on the borders of white areas in I_B, as expressed in Equation (8),

$$b_m = \frac{1}{N_\Delta}\sum_{i=1}^{N_\Delta} I_Y \cdot \Delta I_B, \quad N_\Delta = \sum_{x,y} \Delta I_B(x,y), \tag{8}$$

where ΔI_B denotes the border image obtained from I_B and N_Δ is the number of border pixels. The border area defined in Equation (9) is the same as the value N_Δ.

$$b_{ar} = N_\Delta. \tag{9}$$

The image features D_f, b_m and b_{ar} are calculated using appropriate functions developed in the Python environment for the presented method. The last block of textural features included in Figure 7 concerns the Local Binary Pattern (LBP) applied to the biofuel image. The LBP transforms the intensity image I_Y into the image I_{LBP} with pixel values encoded as bit sequences, where each bit is the result of comparing a pixel to each of its P neighbors at a fixed distance of R pixels [37–39]. The values of I_{LBP} are invariant of the I_Y gray-levels. The bit sequence of the I_{LBP} pixel data is rotated to minimize the pixel value. To obtain an LBP image, we used the function $local\_binary\_pattern$ in the Python package $skimage.feature$ given by Equation (10)

$$I_{LBP} = local\_binary\_pattern(I_Y, P, R, method), \tag{10}$$

where $P = 8$, $R = 1$ (explained in the text), and $method =' ror'$ ensures I_{LBP} values invariant of I_Y rotation. The LBP features are prepared as a normalized histogram B of the encoded image I_{LBP}. The histogram function in Equation (11) is available in the Python package $numpy$.

$$B = histogram(I_{LBP}, bins, range, density = True), \tag{11}$$

where $bins = 32$ is the number of histogram bins and $range = [0, 255]$ is the range of image values included in the histogram. All discussed features are interpreted as $N_f = 48$ columns of one data sample row. The N_s sample rows are collected as a single dataframe $\mathbb{X}$ of size $[N_s \times N_f]$, which is submitted to the standardization stage described in the next section. Image acquisition, preprocessing, and textural feature extraction are the initial stages of the proposed classification method, illustrated as blocks 2, 3, and 4 of the algorithm flowchart in Figures 4 and 5.

2.4. Textural Feature Scaling and Reduction

To avoid the influence of the different scales of the textural features on the biofuel assessment, each of the features were standardized individually by centering them (resetting the means) and then scaling to unit variance, as described in Equation (12). This operation can be easily performed with the Python $StandardScaler().fit\_transform(\mathbb{X})$ function in the $sklearn.preprocessing$ package, where $\mathbb{X}[N_s \times N_f]$ is the data frame of the N_f features, and each feature N_s samples long [40,41]:

$$\mathbb{X}'_j = \frac{\mathbb{X}_j - \mu_j}{s_j}, \quad j \in \{1, \ldots, N_f\} \tag{12}$$

where $\mathbb{X}_j$ and $\mathbb{X}'_j$ denote the j-th feature vectors before and after standardization, and μ_j and s_j are its sample mean and standard deviation, respectively. Zero-variance features are eliminated, as they are irrelevant to differentiating the material class. Because the textural features of images are often correlated, many of them add little information to distinguish the biofuels. Therefore, Principal Component Analysis (PCA) is applied to the original N_f-dimensional feature space, to transform it into a new principal component space where the first component direction maximizes the variance of the projected data and all space directions constitute an orthogonal basis [42,43]. The feature data components PC_i, $i = 1, \ldots, N_f$ are uncorrelated and the first components are of the greatest importance for distinguishing between biofuel classes. To carry out the PCA transformation of the feature dataframe $\mathbb{X}$ in the Python environment, the PCA object functions fit and $transform$ included in the package $sklearn.decomposition$ can be used. These functions are implemented as described in Equation (13)

$$X = pca.transform(\mathbb{X}), \quad pca = PCA(N_c).fit(\mathbb{X}_t), \tag{13}$$

where $X[N_s \times N_c]$ denotes the array of PCA feature components for N_s feature samples, N_c is the number of initial components left by the transform, the fit function calculates the pca as a specific transform rule adapted to the feature training data $\mathbb{X}_t[N_t \times N_f]$ of N_t samples, and the $transform$ function converts any input dataframe $\mathbb{X}[N_s \times N_f]$.

2.5. Biomass Classifier Models

Three types of classifiers are considered in this work: a Linear Discrimination Analyzer (*LDA*) [44], a Random Forest (*RF*) classifier [45,46], and a Deep Neural Network (*DNN*) [47,48]. Each of these classifiers is represented as block 6 of the algorithm flowcharts shown in Figures 4 and 5. The *LDA* classifier divides the space X of N_c feature components by hyperplanes positioned to optimally separate component samples from known, different classes. The proposed classifier model is described by the LDA class in the Python package $sklearn.lda$, as described in Equation (14):

$$lda = LDA(), \tag{14}$$

where the *LDA* constructor parameters keep their default values. The lda model object is trained on the component data X_t with the function given in Equation (15), which sets the optimal separating hyperplanes among the classes:

$$lda.fit(X_t, y_t), \tag{15}$$

where $X_t[N_t \times N_c]$ is the array of N_t dataframe samples used for training, consisting of N_c component features, and $y_t[N_s]$ denotes a vector of class labels. The fitted model is later used to predict class labels using the function shown in Equation (16):

$$y_p = lda.predict(X_p), \tag{16}$$

where $X_p[N_p \times N_c]$ contains N_p samples with predictions returned by $y_p[N_p]$. The Random forest fits the number of decision tree classifiers to random dataset subsamples (with replacement) and produces an ensemble prediction from individual tree predictions using majority voting for a biofuel class. The classifier model is defined as an object of the class $RandomForestClassifier$ contained in the Python package $sklearn.ensemble$, with the adjustable parameters shown in Equation (17) [49].

$$\begin{aligned} rf =& RandomForestClassifier(n\_estimators, max\_depth, bootstrap, \\ & oob\_score, criterion), \end{aligned} \tag{17}$$

where $n\_estimators = 200$ is the number of trees in the forest, $max\_depth = None$ denotes tree nodes expanded until all leaves are pure or until all leaves contain fewer than two samples, $bootstrap = True$ denotes that only bootstrap samples are used to build each tree, $oob\_score = True$ uses out-of-bag samples to estimate the generalization score, and $criterion =' gini'$ is the Gini Index criterion of splitting purity at each tree node [50]. The arguments in Equation (17) are adjusted experimentally to ensure high accuracy of biofuel classification.

The rf model object is trained on the basis of the data X_t and the known class answers y_t using the function described by Equation (18):

$$rf.fit(X_t, y_t), \tag{18}$$

where $X_t[N_t \times N_c]$ is the array of N_t dataframe samples, each consisting of N_c textural feature components used for training, and $y_t[N_t]$ is the vector of known biofuel classes corresponding to the samples of X_t. Prediction is made by the function given in Equation (19):

$$y_p = rf.predict(X_p), \tag{19}$$

where $X_p[N_p \times N_c]$ includes N_p samples with predictions returned in the class vector $y_p[N_p]$. An example fragment of a decision tree generated by this model is shown in Figure 8.

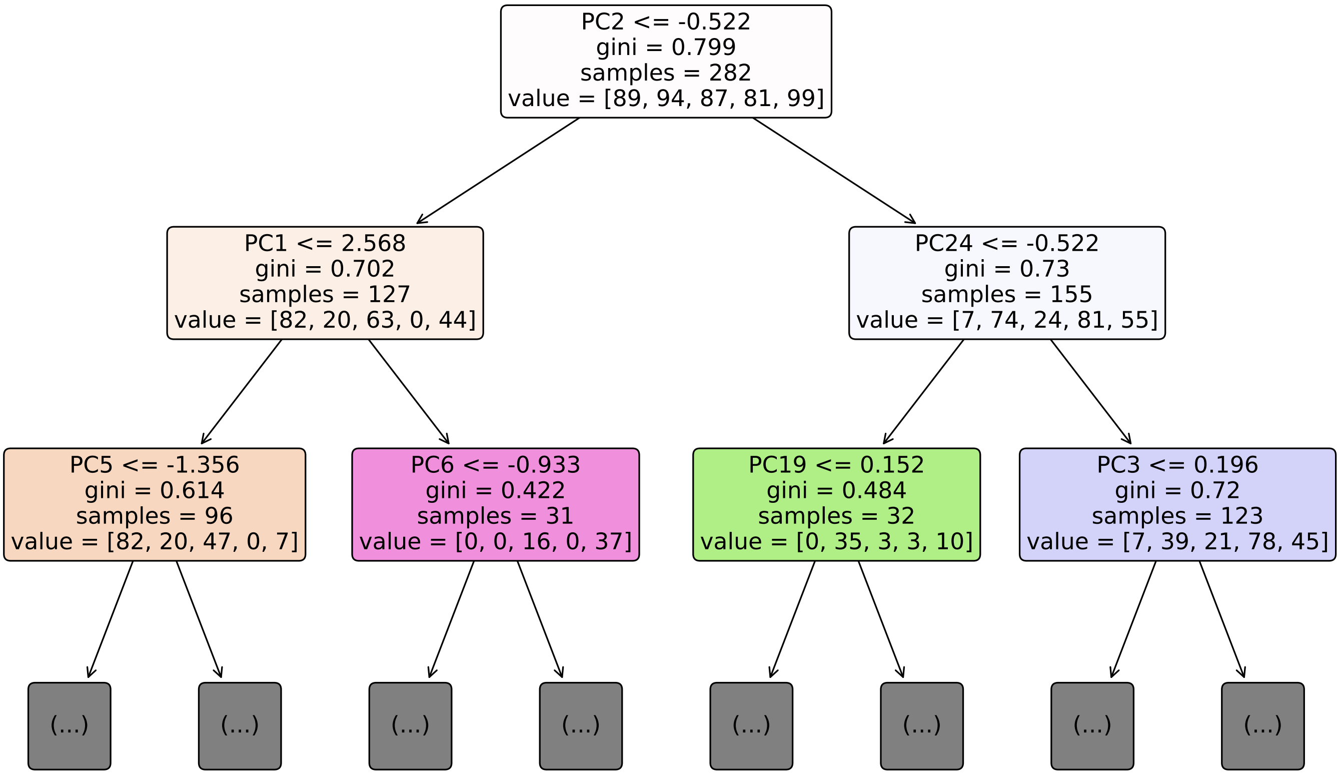


Figure 8. Three initial levels of one of the randomly generated decision trees making up the *RF* classifier. The five elements of *value* in each node correspond to splitting the sample data into five classes. *PCn* denotes the *n*-th PCA component feature.

The proposed *DNN* classifier consists of a sequence of three dense neural layers, where each neuron in every layer computes a weighted sum of activated neuron outputs from the previous layer [47]. The components of the classifier model are visualized in Figure 9. The layers from *layer_1* to *layer_2* implement the Rectifier Linear Unit (*relu*) activation, while the output *layer_3* uses a soft argmax ($softmax$) activator pointing to the output neuron (class) that obtains the maximum value [51]. The *DNN* layers used in the classifier are defined by the function *Dense* available in the Python module $tensorflow.keras.layers$. The function of layers 1 to 3 is shown in Equation (20)

$$layer = Dense(units = S, activation = A), \tag{20}$$

where S and A are set as summarized in Figure 9. S and A refer to the Output Shape and Activation, respectively. Dropout layers 1 and 2 randomly set inputs to 0 with the rates 0.1 and 0.01, respectively, during training time, which helps prevent overfitting [51,52]. Each dropout layer is created by the function *Dropout* included in the Python module $tensorflow.keras.layers$ and illustrated in Equation (21).

$$layer = Dropout(rate = R), \tag{21}$$

where R denotes the proposed rate. The layers are stacked into the model object dnn of the class $tensorflow.keras.Sequential$. The model is then compiled using the method given in Equation (22) to configure it for training:

$$dnn.compile(optimizer = RMSprop(lr, decay, rho)), \tag{22}$$

where $RMSprop$ (Root Mean Square Propagation) is the function of the modern optimization algorithm proposed by Geoff Hinton in the course *Neural Networks for Machine Learning* [53], also discussed in [54], $lr = 5 \times 10^{-4}$ is the initial learning rate of the adaptive algorithm, $rho = 0.9$ is the discounting factor for the gradient calculation, and $decay = 5 \times 10^{-6}$ adjusts lr per optimizer iterations incremented on each batch fit. The dnn model is trained on the data $X_t[N_t \times N_c]$, with the known class answers y_t, using the function in Equation (23):

$$dnn.fit(X_t, y_t, epochs, batch_size), \tag{23}$$

where $epochs = 80$ is the number of passes over the entire feature dataset, and $batch\_size = 5$ is the number of samples propagated through the network in one forward/backward pass. Prediction is performed by the function given in Equation (24):

$$y_p = dnn.predict(X_p), \tag{24}$$

where $X_p[N_p \times N_c]$ includes N_p samples with predictions returned in the class vector $y_p[N_p]$. The classification accuracy in any classifier is defined by Equation (25) as the ratio of the sum of diagonal elements in the confusion matrix to the sum of all its elements.

$$acc = \frac{\sum\limits_{i=1}^{N_C} M_{i,i}}{\sum\limits_{i=1}^{N_C} \sum\limits_{j=1}^{N_C} M_{i,j}}, \tag{25}$$

where $M[N_C \times N_C]$ denotes the confusion matrix and $N_C = 5$ is the number of biofuel classes.

```
Layer (type)                  Output Shape    Activation    Param #
==================================================================
layer_1 (Dense)               (None, 20)      relu          500
__________________________________________________________________
dropout_1 (Dropout) 10%       (None, 20)      –             0
__________________________________________________________________
layer_2 (Dense)               (None, 10)      relu          210
__________________________________________________________________
dropout_1 (Dropout) 1%        (None, 20)      –             0
__________________________________________________________________
layer_3 (Dense)               (None, 5)       softmax       55
==================================================================
Total params: 765
Trainable params: 765
Non–trainable params: 0
__________________________________________________________________
```

Figure 9. Summary of the DNN model proposed for biofuel classification for $N_c = 24$ inputs of the $layer\_1$.

2.6. Prediction Correction Method

Additional correction of classification results is covered in block 7 of the basic algorithm flowchart shown in Figure 5. It is based on the assumption that several images are captured for the same biomaterial and, as a result, several predictions regarding its class can be made. In the case of n independent classification results, the probability of k successes in selecting one correct class can be computed using the binomial distribution with p as the probability of a single success (proper class prediction) [55].

The final biofuel prediction accuracy p_acc is defined as the probability of more than $k = n/2$ successes in n classification trials, which equals the probability of true class prediction when the maximum value location in the histogram of n predictions is selected. This can be computed using the Python binomial cumulative function $scipy.stats.binom.cdf$, as shown in Equation (26):

$$p\_acc = 1 - stats.binom.cdf(k, n, p), \tag{26}$$

where $stats.binom.cdf$ is imported from the $scipy$ package [56], p_acc is the final prediction accuracy, $n = 10$ is the number of independent classification trials, and p is the expected accuracy (probability of true class selection) in a single classification.

3. Results

3.1. Feature Space Scaling and Reduction

The algorithm for the discussed method of biofuel energy class identification was tested for five material classes, listed in Table 1. All the acquired biofuel images used by the image analysis program had the same dimensions of 2560 × 1920 pixels and physical resolution of 2.6 µm/pixel. The training mode requires 500 sample images (100 samples per class). The prediction mode uses 10 sample images of the same, unknown biomaterial, saved separately from the IDB. Figure 10a,c show example images of Class 1 and 2 biofuels, respectively, after scaling down to 320 × 240 pixels. Figure 10b,d refer to the same images after lighting equalization, as described by Equation (1). The results of converting the images into the form of luminance are illustrated in Figure 11.

The original dataframe of textural features in the program training mode, before PCA transform, consists of 500 sample rows (100 per class) as illustrated in Figure 12a, where each row includes $N_f = 48$ textural features and an associated known class label (number). The dataframe in program prediction mode is built of 10 feature rows of the same, unknown biofuel class, as shown in Figure 12b.

On the PCA component plane $(PC1, PC2)$ shown in Figure 13, positive or negative correlations can be observed within several groups of the original textural features—e.g., a positive correlation in the groups H_1, H_5, H_{10} and B_1, B_3, H_{13}. Therefore, prior to classification, it is reasonable to orthogonalize the feature base and further reduce the base dimensionality. After PCA, the sample features are converted to the orthogonal basis of feature components reduced to the first $N_c = 24$ items.

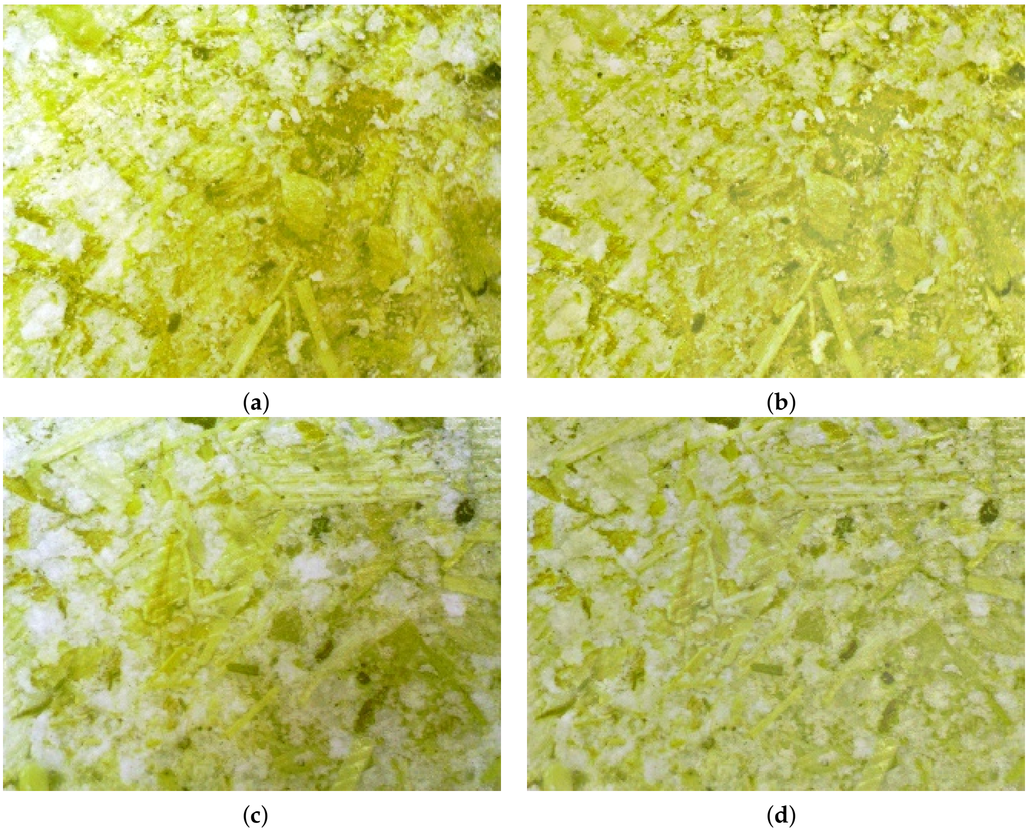


Figure 10. Effect of equalizing the lighting of sample biofuel images, (**a**,**c**)—images from class 1 and 2 with uneven lighting, (**b**,**d**)—the same images after lighting equalization.

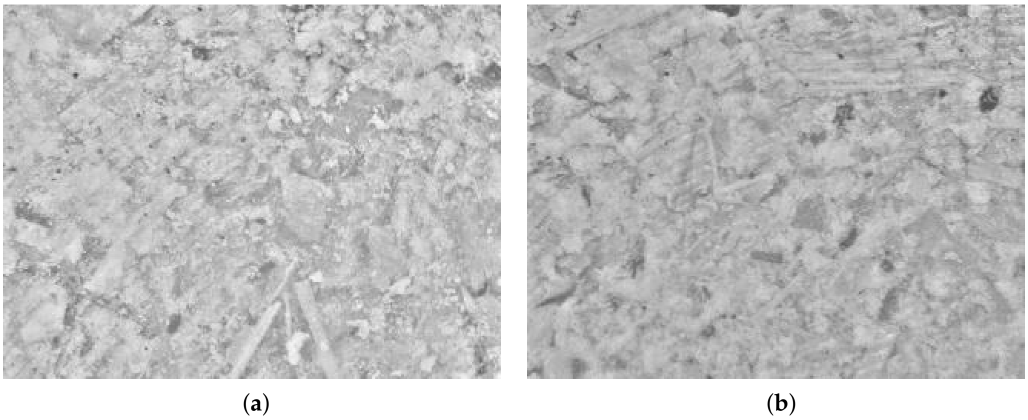
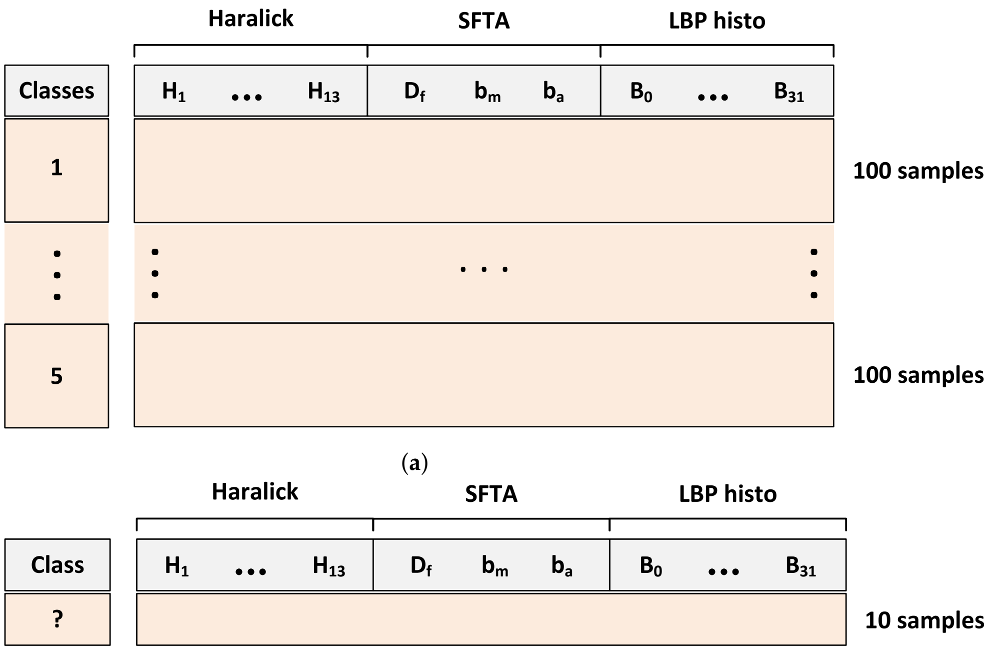


Figure 11. Example images from Figure 10 after conversion to grayscale, (**a**)—luminance of the image from Figure 10b, (**b**)—luminance of the image from Figure 10d



(**a**)

(**b**)

Figure 12. Dataframes of textural features, (**a**) in the training mode, (**b**) in the prediction mode.

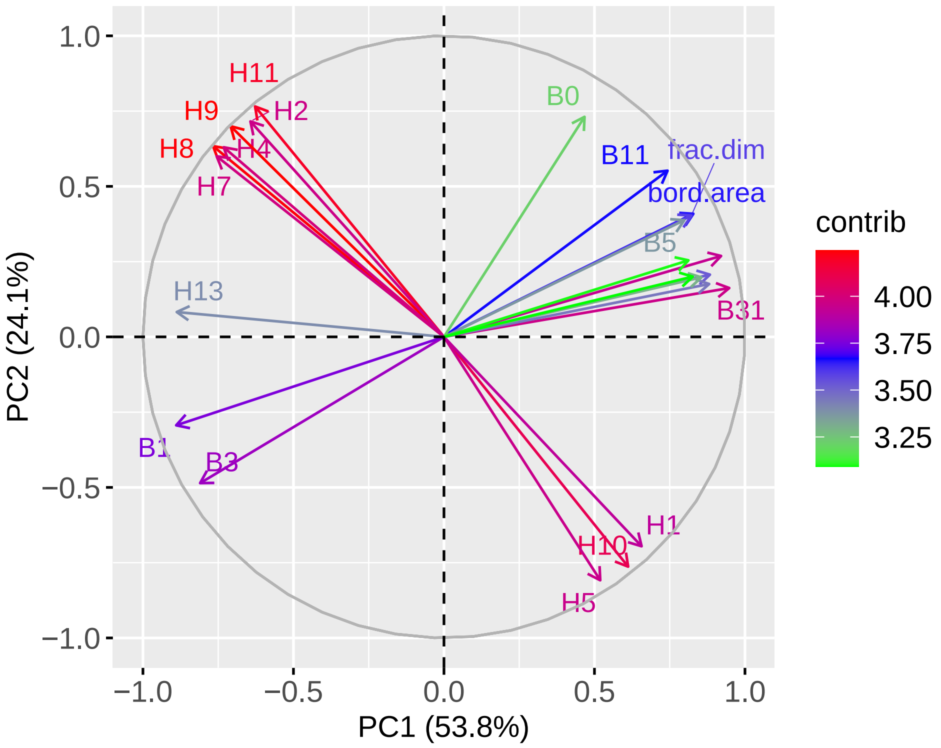
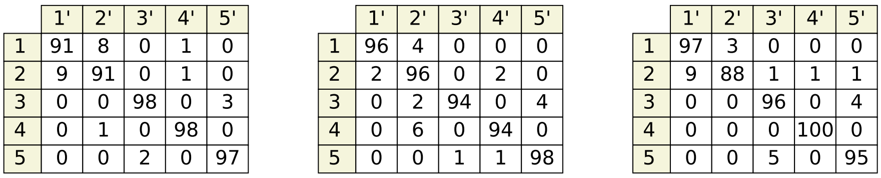


Figure 13. Biplot of textural feature variables on the plane of two major PCA components.

3.2. Classifier Validation and Computing Time

To test the quality of biofuel prediction, 110 feature samples (sets) from each of the 5 biofuel classes were taken and 10 of the samples per class were randomly selected for prediction (total 50 samples). The remaining 100 samples per class were left for classifier training (total 500 samples). After 10 validations had been made on the data split this way, the prediction results were accumulated in confusion matrices obtained for the LDA, RF, and DNN classifiers, as shown in Figure 14a–c, respectively.



	1'	2'	3'	4'	5'
1	91	8	0	1	0
2	9	91	0	1	0
3	0	0	98	0	3
4	0	1	0	98	0
5	0	0	2	0	97

(**a**)

	1'	2'	3'	4'	5'
1	96	4	0	0	0
2	2	96	0	2	0
3	0	2	94	0	4
4	0	6	0	94	0
5	0	0	1	1	98

(**b**)

	1'	2'	3'	4'	5'
1	97	3	0	0	0
2	9	88	1	1	1
3	0	0	96	0	4
4	0	0	0	100	0
5	0	0	5	0	95

(**c**)

Figure 14. Cumulative confusion matrices for 10 validations of 50 feature sets with equal frequency in each class, $\{1, \ldots, 5\}$—true classes, $\{1', \ldots, 5'\}$—predicted classes, (**a**) LDA classifier, (**b**) RF classifier, (**c**) DNN classifier.

The accuracy achieved in the 10 validation tests are presented in Table 2. The average accuracy for the LDA, RF, and DNN classifiers was almost the same, at about 0.95.

Table 2. Prediction accuracy for 10 validations of 50 feature samples (10 per class). *LDA*—Linear Discrimination Analysis, *RF*—Random Forest classifier, *DNN*—Deep Neural Network, *av*—mean accuracy, *sd*—standard deviation.

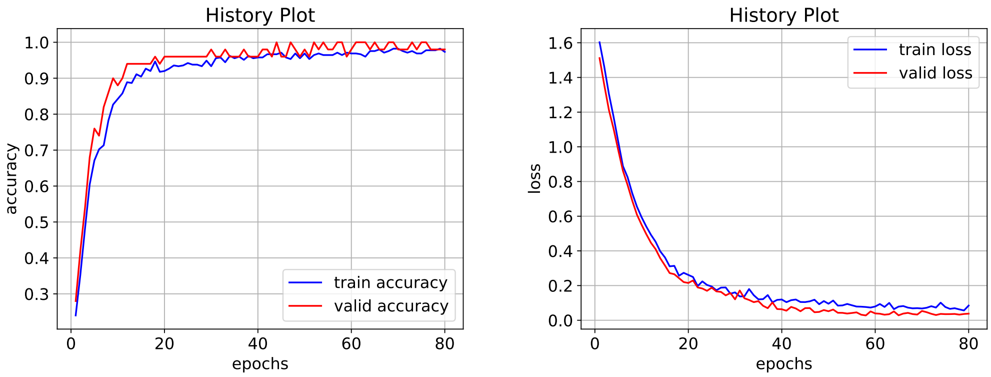
Test		1	2	3	4	5	6	7	8	9	10	*av*	*sd*
acc	*LDA*	0.96	0.88	0.98	0.96	0.96	0.96	0.96	0.98	0.90	0.96	0.95	0.03
	RF	0.96	0.96	0.96	0.94	0.96	0.98	0.98	0.92	0.96	0.94	0.96	0.02
	DNN	0.92	0.96	0.96	1.00	0.90	0.94	0.98	0.98	1.00	0.88	0.95	0.04

Once 10 independent feature samples per class have been predicted, the confusion matrix row contains a histogram of the prediction results for that class. The prediction accuracy of a class indicated by the histogram mode is increased to $p\_acc = 0.9999$, assuming $p = 0.95$ for single prediction success and $n = 10$ independent experiments, as shown in Equation (26).

The loss function of the DNN classifier illustrated in Figure 15b is calculated as the categorical cross-entropy, defined in Equation (27) [57]. This loss is a very good measure of the diversity between two discrete probability distributions, represented by the categorical output vectors y and $\hat{y}$:

$$loss = -\sum_{i=1}^{N_C} y_i \cdot log\, \hat{y}_i, \tag{27}$$

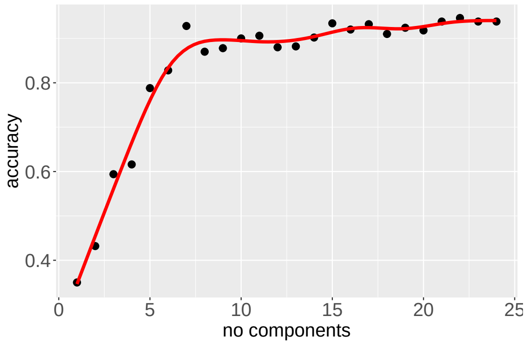
where $\hat{y}_i$ is the i-th scalar value in the model output, and y_i is the i-th element of a ground truth binary output vector corresponding to the class value.



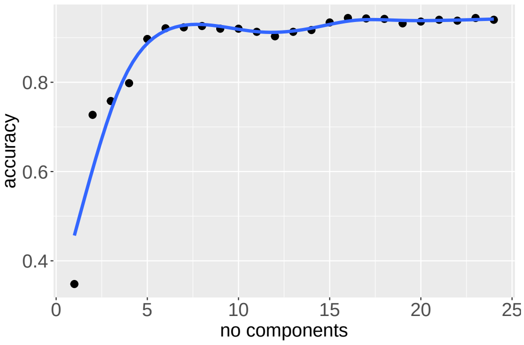
(**a**) (**b**)

Figure 15. Example changes in model metrics during DNN classifier training/validation, (**a**) accuracy (**b**) loss.

The number $N_c = 24$ of first PCA components presented to a classifier was matched to the assumed expected accuracy of the biomaterial classification, which is equal to about 0.95. Exemplary plots of the observed accuracy changes versus the number of components are shown in Figure 16a,b. The execution times of various stages of the proposed algorithm were measured for the computation unit using the hardware and software equipment discussed in Section 2.2.



(**a**)



(**b**)

Figure 16. Observed changes in classification accuracy with the number of initial PCA components used, (**a**) *LDA* classifier, (**b**) *RF* classifier.

The image preprocessing and feature extraction steps represented by block 3 and block 4 both in Figures 4 and 5 require the execution times given in Table 3. The times refer to the analysis of a single image and need to be multiplied by the number of images required, which depends on the program mode. The total time needed for preprocessing and preparing textural features for the built-in classifier does not exceed 2 s in the prediction mode and 100 s in the training mode. Typical times characterizing the work of the proposed classifiers are given in Table 4. They show the definite speed advantage of the *LDA* classifier, which is about 6 times faster at prediction and 45 times faster in training than the proposed *RF* classifier with 200 decision trees. The *RF* classifier is in turn about 100 times faster than the *DNN* classifier during prediction and training, with the *DNN* classifier requiring 80 training epochs.

Table 3. Execution times of image preprocessing and textural feature computing for a single image, *S*1—image rescaling, *S*2—lighting equalization and luminance computing, *S*3, *S*4, and *S*5—stages of Haralick, SFTA, and LBP feature extraction.

Preprocessing		**Feature Extraction**		
S1 **[ms]**	**S2** **[ms]**	**S3** **[ms]**	**S4** **[ms]**	**S5** **[ms]**
126.60	8.06	23.36	13.20	25.86

Table 4. Times of classifier training and prediction for $N_c = 24$ PCA feature components.

Classifier	**Training** **[s]**	**Prediction** **[s]**
LDA	0.009	0.002
RF	0.413	0.012
DNN	39.187	1.615

Nevertheless, even a time of around 40 s for training on 500 images appears to be acceptable. It should be noted that the *DNN* model training function $dnn.fit$ in Equation (23) used GPU GTX 1050 parallel computing. The relatively long duration of the rescaling stage

*S*1 was spent reading from the disk hundreds of JPEG image files with large dimensions of 2560 × 1920 pixels.

3.3. PLSDA Linear Classifier

The classification accuracy of the proposed method was briefly compared with the performance of the Partial Least Squares-Discriminant Analysis (PLSDA) linear classifier. The input data, consisting of original textural feature values after scaling, were transferred to the PLSDA model designed in R language [58,59]. The performance of the model was evaluated using the R function *mixOmics::perf*, with 10-fold cross-validation repeated 100 times [60,61], illustrated in Figure 17 as the classification error rate. This rate is presented as a function of the number of components included in the model. Separate plots are created for the Mahalanobis and maximum distances between clusters in the textural feature space [62]. The error rates do not fall below 9 ÷ 10%, although as many as 24 components from 48 dimensional feature space are used. The expected errors of the PLSDA model are 1.8 ÷ 2 times as large as for the nonlinear classifiers. Balanced errors (BER) are the same as overall for the balanced representation of class samples.

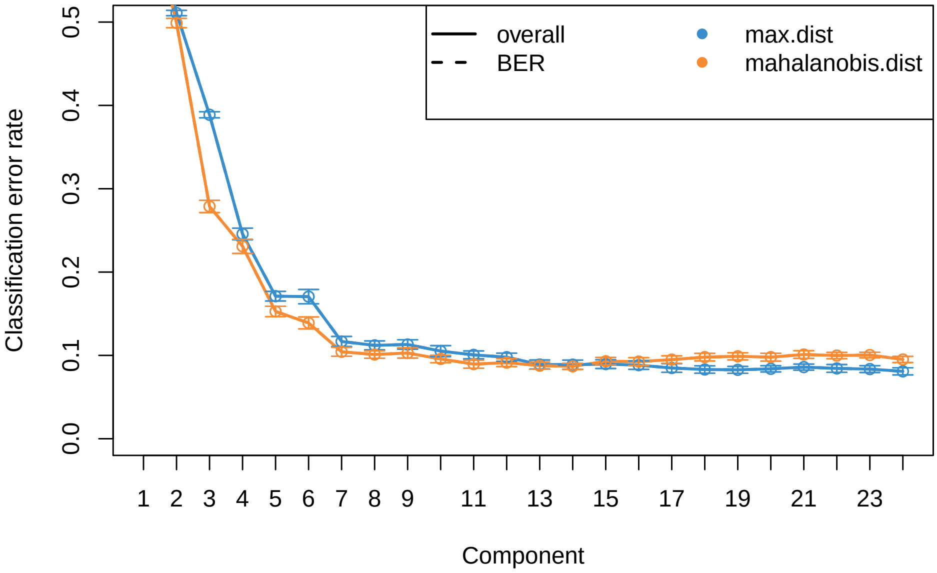


Figure 17. PLSDA classification errors for biofuel materials, *overall*—overall error rate, *BER*—balanced error rate.

4. Discussion

It is easy to notice from Tables 3–5 that the computing time in the prediction mode (2 s) and training mode (100 ÷ 140 s) is negligibly lower than the time for activities related to entering data into the system, that are 100 s and 5000 s respectively. The time of combustion cycles for single material sample according to EN ISO 18125:2017-07 equals 9000 s instead of 100 s in our system prediction mode, i.e., is 90 times slower. Among the several classifiers used to identify the biofuel type, the linear discriminator *LDA* is preferred, as it is the fastest during training and sample assessment. It also provides similar expected accuracy to other methods for material recognition from a single image sample. Nevertheless, when the group of biofuel materials to be recognized is expanded, the *RF* classifier with its nonlinear separation ability may provide better accuracy, at the cost of a slightly extended training and testing time.

Table 5. Times of feature scaling and PCA transform.

Scaling		**PCA**	
Training [ms]	**Prediction [ms]**	**Training [ms]**	**Prediction [ms]**
0.65	0.15	9.20	0.10

So far, the calorimetric method or the study of the elemental composition was used to assess the calorific value of biomass samples. The computer method proposed in the article has not been found in the literature. According to the authors' knowledge, the analysis of carbon surface textures was used only to better understand the behavioural pattern of input material during agglomeration in the pressing chamber of a briquetting machine. The analysis of the images allowed for the determination of statistical significant differences in the particle size distribution of the material subjected to the briquetting process [63]. The image segmentation and classification in the area of fuels was also used for identification of maceral components of coal [64], but there is no information about caloric assessment.

5. Conclusions

The proposed method enables objective differentiation of a specific group of solid biofuel materials, based on their surface textural features. This approach avoids human errors that could appear during visual assessment, because the samples look very similar at both the macro scale and micro scale. The method is designed for quick assessment of biomass calorific properties. It does not require burning all the tested samples in a calorimetric bomb (as recommended by the standard EN ISO-18125), which is also associated with a relatively slow measurement process. In total, the estimated time required to measure the calorific value of one solid biofuel type using the traditional measurement method is about 2.5 h. The method of assessing the calorific value of samples proposed in this article, using the *LDA*, *RF*, or NN classifiers and a prebuilt database, reduces the time needed for assessment by approximately 99%. The estimated time includes image set (80100 items) acquisition, preprocessing, and sample classification. The calorific values need to have been measured by physical methods only once, at the method calibration stage. The presented method enables the identification of a biofuel sample with accuracy above 99%, using feature variations measured in many different areas of a single material sample.

Author Contributions: Conceptualization, J.G., J.S.-N. and E.K.; methodology, J.G., J.S.-N. and E.K.; software, J.G.; validation, J.G., J.S.-N. and E.K.; formal analysis, J.G. and J.S.-N.; investigation, J.G. and J.S.-N.; resources, J.G., J.S.-N. and E.K.; data curation, J.S.-N., E.K., P.K. and T.D.; writing—original draft preparation, J.G., E.K., P.K. and J.S.-N.; writing—review and editing, J.G., E.K., J.S.-N., P.K. and T.D.; visualization, J.G., E.K. and J.S.-N.; supervision, J.G., E.K. and J.S.-N.; project administration, J.G., J.S.-N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

EC	European Commission
IDB	image database
PC	Personal Computer
CPU	Central Processing Unit
RAM	Random Access Memory
GPU	Graphic Processing Unit
GLCM	Gray Level Co-occurrence Matrix
SFTA	Segmentation-based Fractal Texture Analysis
LBP	Local Binary Pattern
PCA	Principal Component Analysis
RF	Random Forest
DNN	Deep Neural Network
LDA	Linear Discrimination Analyzer
PLSDA	Partial Least Squares-Discriminant Analysis
BER	balanced error rate
PC1, PC2	principal components 1, 2

References

1. European Comission. 2030 Climate & Energy Framework. Available online: https://ec.europa.eu/clima/eu-action/climate-strategies-targets/2030-climate-energy-framework_pl (accessed on 14 November 2021).
2. Sarkar, N.; Kumar, S.; Bannerjee, G.; Aikat, K. Bioethanol production from agricultural wastes: An overview. *Renew. Energy* **2012**, *37*, 19–27. [CrossRef]
3. Gradziuk, P.; Gradziuk, B.; Trocewicz, A.; Jendrzejewski, B. Potential of Straw for Energy Purposes in Poland—Forecasts Based on Trend and Causal Models. *Energies* **2020**, *13*, 5054. [CrossRef]
4. Millati, R.; Cahyono, R.; Ariyanto, T.; Azzahrani, I.; Putri, R.; Taherzadeh, M. Agricultural, Industrial, Municipal, and Forest Wastes: An Overview. In *Sustainable Resource Recovery and Zero Waste Approaches*; Elsevier B.V.: Amsterdam, The Netherlands, 2019; pp. 1–22. [CrossRef]
5. Lantz, M.; Prade, T.; Ahlgren, S.; Björnsson, L. Biogas and ethanol from wheat grain or straw: Is there a trade-off between climate impact, avoidance of iLUC and production cost? *Energies* **2018**, *11*, 2633. [CrossRef]
6. Singh, J. Identifying an economic power production system based on agricultural straw on regional basis in India. *Renew. Sustain. Energy Rev.* **2016**, *60*, 1140–1155. [CrossRef]
7. Marks-Bielska, R.; Bielski, S.; Novikova, A.; Romaneckas, K. Straw Stocks as a Source of Renewable Energy. A Case Study of a District in Poland. *Sustainability* **2019**, *11*, 4714. [CrossRef]
8. Lunguleasa, A.; Spirchez, C.; Zeleniuc, O. Evaluation Of The Calorific Values Of Wastes From Some Tropical Wood Species, Maderas. *Cienc. Tecnol.* **2020**, *22*, 269–280. [CrossRef]
9. Logeswaran, J.; Shamsuddin, A.; Silitonga, A.; Mahlia, T. Prospect of using rice straw for power generation: A review. *Environ. Sci. Pollut. Res.* **2020**, *27*, 25956–25969. [CrossRef]
10. Spirchez, C.; Lunguleasa, A.; Ionescu, C.; Croitoru, C. Physical and calorific properties of wheat straw briquettes and pellets. *MATEC Web Conf.* **2019**, *290*, 11011. [CrossRef]
11. Naik, S.; Goud, V.V.; Rout, P.K.; Jacobson, K.; Dalai, A.K. Characterization of Canadian biomass for alternative renewable biofuel. *Renew. Energy* **2010**, *35*, 1624–1631. [CrossRef]
12. Herkowiak, M.; Adamski, M.; Dworecki, Z.; Waliszewska, B.; Pilarski, K.; Witaszek, K.; Niedbała, G.; Piekutowska, M. Analysis of the possibility of obtaining thermal energy from combustion of selected cereal straw species. *J. Res. Appl. Agric. Eng.* **2018**, *63*, 68–72.
13. Pordesimo, L.O.; Hames, B.R.; Sokhansanj, S.; Edens, W.C. Variation in corn stover composition and energy content with crop maturity. *Biomass Bioenergy* **2005**, *28*, 366–374. [CrossRef]
14. Morissette, E.; Savoie, P.; Villeneuve, J. Combustion of Corn Stover Bales in a Small 146-kW Boiler. *Energies* **2011**, *4*, 1102–1111. [CrossRef]
15. Chou, C.; Lin, S.; Lu, W. Preparation and characterization of solid biomass fuel made from rice straw and rice bran. *Fuel Process. Technol.* **2009**, *90*, 980–987. [CrossRef]
16. Chou, C.; Lin, S.; Peng, C.; Lu, W. The optimum conditions for preparing solid fuel briquette of rice straw by a piston-mold process using the Taguchi method. *Fuel Process. Technol.* **2009**, *90*, 1041–1046. [CrossRef]
17. Denisiuk, W. Straw as fuel. *Inżynieria Rolnicza* **2009**, *1*, 83–89.
18. Jach-Nocoń, M.; Pełka, G.; Luboń, W.; Mirowski, T.; Nocoń, A.; Pachytel, P. An Assessment of the Efficiency and Emissions of a Pellet Boiler Combusting Multiple Pellet Types. *Energies* **2021**, *14*, 4465. [CrossRef]

19. Toscano, G.; Pedretti, E. Calorific Value Determination Of Solid Biomassfuel By Simplified Method. *J. Agric. Eng.* **2009**, *3*, 1–6. [CrossRef]
20. Sheng, C.; Azevedoj, L. Estimating the higherheating value of biomass fuels from basic analysis data. *Biomass Bioenergy* **2005**, *28*, 499–507. [CrossRef]
21. Mostaço-Guidolin, L.B.; Ko, A.C.; Wang, F.; Xiang, B.; Hewko, M.; Tian, G.; Major, A.; Shiomi, M.; Sowa, M.G. Collagen morphology and texture analysis: From statistics to classification. *Sci. Rep.* **2013**, *3*, 2190. [CrossRef]
22. Beguet, B.; Guyon, D.; Boukir, S.; Chehata, N. Automated retrival of forest structure variables based on multis-scale texture analysis of VHR satelite imagery. *ISPRS J. Photogramm. Remote Sens.* **2014**, *96*, 164–178. [CrossRef]
23. Lottering, R.T.; Govender, M.; Peerbhay, K.; Lottering, S. Comparing partial least squares (PLS) discriminant analysis and sparse PLS discriminant analysis in detecting and mapping Solanum mauritianum in commercial forest plantations using image texture. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 271–280. [CrossRef]
24. Larmuseau, M.; Sluydts, M.; Theuwissen, K.; Duprez, L.; Dhaene, T.; Cottenier, S. Compact representations of microstructure images using triplet networks. *NPJ Comput. Mater.* **2020**, *6*, 156. [CrossRef]
25. Nurski, M. Sony Wants to Make Sense of Taking Photos of Food. Available online: https://komorkomania.pl/34226,sony-aplikacja-kalorie-zdjecie (accessed on 10 November 2021).
26. Available online: https://apkpure.com/work-performance-plus/biz.sonymobile.wpp (accessed on 14 November 2021).
27. Bite AI, Inc. Bitesnap. The Easier Way to Track What You Eat. Available online: https://getbitesnap.com/ (accessed on 14 November 2021).
28. EN. ISO 18125:2017 Solid Biofuels—Determination of Calorific Value Standard. Available online: https://www.iso.org/standard/61517.html (accessed on 14 November 2021).
29. Dey, N. Uneven illumination correction of digital images: A survey of the state-of-the-art. *Optik* **2019**, *183*, 483–495. [CrossRef]
30. Gonzalez, R.; Woods, R.R.E. *Digital Image Processing*, 4th ed.; Pearson: London, UK, 2017; p. 1168. Available online: http://www.mypearsonstore.com/bookstore/digital-image-processing-9780133356724 (accessed on 14 November 2021).
31. Haralick, R.; Shanmugan, K.; Dinstein, I. Textural Features for Image Classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *3*, 610–621. Available online: http://haralick.org/journals/TexturalFeatures.pdf (accessed on 14 November 2021). [CrossRef]
32. Haralick, R.; Shapiro, L. *Computer and Robot Vision*; Addison-Wesley Pub. Co.: Boston, MA, USA, 1992; p. 459.
33. Costa, A.; Humpire-Mamani, G.; Traina, A. An Efficient Algorithm for Fractal Analysis of Textures. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 22–25 August 2012. [CrossRef]
34. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
35. Iannaccone, P.; Khokha, M. *Fractal Geometry in Biological Systems: An Analytical Approach*; CRC Press: Boca Raton, FL, USA, 1996; p. 368.
36. Yan, X. *Linear Regression Analysis: Theory and Computing*; World Scientific Publishing: Singapore, 2009; p. 348. [CrossRef]
37. Ojala, T.; Pietikäinen, M. Unsupervised Texture Segmentation Using Feature Distributions. *Pattern Recognit.* **1999**, *32*, 477–486. [CrossRef]
38. Ojala, T.; Pietikäinen, M.; Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Recognit.* **2002**, *24*, 971–987. [CrossRef]
39. Pietikäinen, M.; Hadid, A.; Zhao, G.; Ahonen, T. *Computer Vision Using Local Binary Patterns*; Springer: Berlin/Heidelberg, Germany, 2011; p. 207. [CrossRef]
40. Eddie_4072. Feature Scaling Techniques in Python—A Complete Guide. Available online: https://www.analyticsvidhya.com/blog/2021/05/feature-scaling-techniques-in-python-a-complete-guide/ (accessed on 5 November 2021).
41. Scikit-learn Developers. StandardScaler. 2021. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (accessed on 5 November 2021).
42. Jolliffe, I. *Principal Component Analysis,* 2nd ed.; Springer: New York, NY, USA, 2002; p. 493.
43. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [CrossRef]
44. McLachlan, G. *Discriminant Analysis and Statistical Pattern Recognition*; Wiley Interscience: Hoboken, NJ, USA, 2004; p. 526.
45. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
46. Liu, Y.; Wang, Y.; Zhang, J. New machine learning algorithm: Random Forest. In *Information Computing and Applications*; Liu, B., Ma, M., Chang, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; pp. 246–252. [CrossRef]
47. Gulli, A.; Pal, S. *Deep Learning with Keras*; Packt Publishing: Birmingham, UK, 2017; p. 318.
48. Vasilev, I.; Slater, D.; Spacagna, G. *Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with PyTorch, Keras, and TensorFlow*, 2nd ed.; Packt Publishing: Birmingham, UK, 2019; p. 386.
49. Scckit-learn Developers. sklearn.ensemble.RandomForestClassifier. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed on 5 November 2021).
50. Raileanu, L.; Stoffel, K. Theoretical comparison between the Gini index and information gain criteria. *Ann. Math. Artif. Intell.* **2004**, *41*, 77–93. [CrossRef]
51. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; Packt Publishing: Birmingham, UK, 2016; p. 800.
52. Wiki, A. Overfitting vs. Underfitting. Available online: https://docs.paperspace.com/machine-learning/wiki/overfitting-vs-underfitting (accessed on 5 November 2021).

53. Hinton, G. Neural Networks for Machine Learning Online Course. Available online: https://www.coursera.org/learn/neural-networks/home/welcome (accessed on 14 November 2021).
54. Bushaev, V. Understanding RMSprop—Faster Neural Network Learning. Available online: https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a (accessed on 14 November 2021).
55. Westland, J. *Audit Analytics: Data Science for the Accounting Profession*, 1st ed.; Springer: Chicago, IL, USA, 2020; p. 344.
56. The SciPy Community. scipy.stats.binom. Available online: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.binom.html (accessed on 14 November 2021).
57. Gómez, R. Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and All Those Confusing Names. Available online: https://gombru.github.io/2018/05/23/cross_entropy_loss/ (accessed on 5 November 2021).
58. Lee, L.; Liong, C.; Jemain, A. Partial least squares discriminant analysis (PLSDA) for classification of high-dimensional (HD) data: A review of contemporary practice strategies and knowledge gaps. *Analyst* **2018**, *143*, 3526–3539. [CrossRef] [PubMed]
59. Lantz, B. *Machine Learning with R—Third Edition: Expert Techniques for Predictive Modeling*; Packt Publishing: Birmingham, UK, 2019; p. 458.
60. Biocondictor. mixOmics. Available online: http://www.bioconductor.org/packages/release/bioc/manuals/mixOmics/man/mixOmics.pdf (accessed on 5 November 2021).
61. Rohart, F.; Gautier, B.; Singh, A.; Lê Cao, K.A. mixOmics: An R package for 'omics feature selection and multiple data integration. *PLoS Comput. Biol.* **2017**, *13*, e1005752. [CrossRef] [PubMed]
62. Prabhakaran, S. Mahalanobis Distance–Understanding the Math with Examples (Python). 2019. Available online: https://www.machinelearningplus.com/statistics/mahalanobis-distance/ (accessed on 5 November 2021).
63. Chaloupková, V.; Ivanova, T.; Ekrt, O.; Kabutey, A.; Herák, D. Determination of Particle Size and Distribution through Image-Based Macroscopic Analysis of the Structure of Biomass Briquettes. *Energies* **2018**, *11*, 331. [CrossRef]
64. Wang, H.; Lei, M.; Chen, Y.; Li, M.; Zou, L. Intelligent Identification of Maceral Components of Coal Based on Image Segmentation and Classification. *Appl. Sci.* **2019**, *9*, 3245. [CrossRef]