Journal of Theoretical and Applied Information Technology

15th November 2023. Vol.101. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL DEEP LEARNING MODEL FOR RAISIN GRAINS CLASSIFICATION

TAHA M. MOHAMED[1,2], SARAH N. ABDULKADER[1,3]

[1]Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt

[2]College of Business, University of Jeddah, Jeddah, Kingdom of Saudi Arabia (KSA)

[3]Faculty of Computer Studies, Arab Open University, Cairo, Egypt

E-mail: dr.tahamahdy@fci.helwan.edu.eg, tmmohamed@uj.edu.sa

sarah_abdullah@fci.helwan.edu.eg, sara.nabil@aou.edu.eg

ABSTRACT

Automatic goods classification is important to facilitate the operations performed at customs. Raisin grains are one of these important goods that need accurate classification. The performances of the algorithms available in the literature are not satisfactory. So, this paper presents a high performance deep-learning model for raisin grains classification. The proposed model performs the necessary preprocessing steps on a publicly available dataset. The architecture of the proposed model is the standard architecture which is more appropriate for the tabular dataset used. Many preprocessing tasks are evaluated before constructing the deep network. Moreover, the features of the dataset are analyzed and preprocessed by using feature normalization and principal component analysis (PCA). The performance of the deep network is evaluated using different network configuration for optimal dataset modeling. Moreover, the proposed model evaluates the validity of some network regularization techniques to maximize the performance of the proposed model. The proposed results show that the proposed model outperforms other works in the literature when applied on the same dataset. The proposed accuracy and F-Score exceed 91% in high correlation dataset features. Comparing to previous works, at least 5% improvements are achieved using the proposed model. An important conclusion from the obtained results is that dimension reduction methods are not effective on all datasets. The results also support the importance of the preprocessing steps in enhancing the deep network performance.

Keywords— *Deep Learning; Raisin Grains Classification; Hyper-Parameters Tuning; Features Normalization.*

1. INTRODUCTION

Supply chain management (SCM) aims to deliver and classify products and goods in a timely manner with a good accuracy [1]. The fourth industrial revolution (Industry 4.0) technologies, such as machine learning (ML) [2], may help in solving SCM challenges referred to as SCM 4.0. Many challenges face SCM. For example, in ports, serious challenges face extensive container inspection when examined by customs to verify goods compliance, origin, type, and value of goods [3]. Additional challenges face agricultural products inspection [4]. Often, customs inspectors analyze agricultural products manually to take the appropriate decision [5]. However, this manual work is inefficient and heavily dependent on the experience of the customs inspectors [3], [5]. The complexity of the inspection process increases when dealing with goods sensitive to delay or long-run storage [6]. For example, vegetables, fruits, snacks, and many other foods may spoil if stored for an extended period in ports. In this case, automatic, accurate, and rapid classification of these goods is required [7]. One of the agricultural products is the raisin grains. The raisins are dried grapes rich in dietary fiber and vitamins. Raisins are produced in many regions of the world in various sizes and colors [4], [8]. In 2017, the production of raisins reached 1.22 million tons worldwide [8]. Different raisins have different properties and different commercial values. Fig. 1 shows three different colors of raisins.

However, fake raisin has seriously harmed the market of consumers and enterprises [9]. Also, manual grading of raisins includes more labor requirements, time-consuming, low-quality work, and inaccuracy [4]. So, raisin classification is one of the main challenges facing producers and buyers

[8], [9]. Automatic raisin classification may use some raisin features such as color, texture, quality, and size [10].

Black raisin Golden raisin Green raisin

Figure 1: Three Different Types of Raisin Grains

Deep learning (DL) is the core of machine learning techniques that rely on deep neural networks. These networks may be used in efficient classification, clustering, and regression. DL outperforms classical machine learning techniques such as random forest, support vector machines (SVM), decision trees, and Bayesian classifiers [11]. DL could be used in automated classification of raisin grains if proper raisin features and proper classification models are chosen. In this case, the automatic classification performed by DL can speed up custom inspections. However, the classes of raisins may not be discriminated from each other's. This is clear from the works proposed in [10] and [12] where only 86% accuracy was achieved. Also, the raisin features are highly correlated. The authors in [10] uses some classifiers to achieve this accuracy. According to their results, SVM achieves the highest accuracy. So, there is a need for more robust model to classify raisin grains with a higher accuracy.

This paper presents a novel deep-learning model for raisin grains classification. This classification aims to classify raisin grains into two classes: Kecimen and Besni. We used the public dataset of raisin grains proposed in [10], [12] for performance comparison. The architecture of the proposed model is the standard architecture which is more appropriate for the tabular dataset used. The standard architecture consists of one input layer representing numeric features values, fully connected hidden layers, and one output layer for classification labeling.

This proposed work begins by investigating and analyzing the features of the dataset and use some preprocessing methods to decrease features correlation. So, a novel deep neural network model is proposed. Additionally, the proposed model is tuned by adjusting the hyper-parameters. The paper tests many scenarios and many configurations to choose the best model. The performance of the proposed model is evaluated by using many performance graphs and performance metrics. The main target of this proposed work is to achieve a significant improvement in model performance compared to other works in the literature. Overall, the proposed results outperform other methods in the literature. So, the main contributions of this paper are as follows:

- Features analysis of raisin dataset proposed in [10] and [12].
- Evaluation of the different preprocessing techniques on the dataset
- Designing a standard deep neural network architecture for classifying raisin grains
- Developing a robust raisin grains classifier
- Evaluation of the deep network regularization
- Tuning the deep network hyperparameters
- Evaluation of the model performance
- Results discussions and comparison with other related work
- Discussing the results and presenting future directions

The rest of this paper is organized as follows; Section 2 presents the related work. Section 3 presents the necessary background of the deep neural networks and their configuration. Section 4 presents the proposed model. The experimental results are shown in Section 5. Finally, the paper is concluded in section 6.

2. RELATED WORK

Manual classification of agricultural products is very costly and unreliable. The decisions made by human are often slow, subjective, and inconsistent [13]. So, Machine learning could be used in automatic classification for accuracy and consistency. As an example of agricultural products classification, the authors in [13] use machine learning techniques to classify paddy rice seeds. The authors achieve the best accuracy by using deep learning techniques which exceeds 95%. Also, in [7], the authors use machine learning models for tea classification. The authors in [14] use machine learning techniques to determine four aspects of snacks including production technology, raw material, frying oil, and place of origin [14]. Regarding raisin grains classification, the authors in [10] present a machine vision system to classify two types of raisins grains: Kecimen and Besni. They

extract the features of 900 records of raisin grains. Additionally, they propose seven features to be used in the classification process. However, all chosen features are morphological features. They used three different classifiers. Among these classifiers, SVM achieves the highest accuracy; 86.44% only. The same dataset proposed in [10] and [12] has been tested by Kaggle community [15] from developers and researchers. To our knowledge, the maximum accuracy achieved in [15] is below 89%.

In [16], the authors extend the raisin classification process by investigating raisin quality using image processing techniques. Their work examines 750 raisins images of both good and bad raisins. In contrast to the work in [10], the authors in [16] use different texture features in the classification process. Again, their results indicate that SVM achieves the highest classification results. The results indicate that automated systems can accurately classify a bulk of raisin grains. Another aspect of raisin grain classification is presented in [8]. The authors in [8] combine near-infrared (NIR) spectroscopy and ML to identify different raisins varieties. Also, they used the principal component analysis (PCA) to extract the best features. They use many classifiers to evaluate performance [8]. Similar work is proposed in [9], where a combination of NIR and ML is used to classify three raisin types. In their work, the spectral data is normalized after collection. Then, the PCA is applied for dimension reduction.

Furthermore, the authors in [17] present an automatic grading system for grading raisin grains. The grading relies on the raisin's color and size features. Their method converts the color image to a binary image after some pre-processing. They use 19 features for raisin classification. The color feature is the most important feature in raisin classification and sorting in their work. The raisins sorting is performed according to the Red Green Blue (RGB) color system [17]. To summarize the previous work regarding raisins classification, Table 1 summarizes the main merits and drawbacks of the literature. From the previous discussion, the raisin grains classification is still a challenging problem due to lack of robust features and robust classification models. Also, the datasets available are limited. The performances of the existing algorithms are still limited. The accuracies of these algorithms are still below 90%. This is due to some challenges. The challenges are mainly related to the choice of the proper features, pre-processing algorithms, and classifiers. However, DL models are not heavily adopted in raisin grains classification despite their adoption in other similar fields. However, adopting DL algorithms in raisin grains classification could enhance classification accuracy as shown from the proposed results. The previous work leads to designing a robust deep learning model to deal with the raisins classification challenges. So, this paper presents an efficient deep-learning model for raisin grains classification. The details of this model are shown in sections 4 and 5.

Table 1 : Comparison of Related Work of Raisin Classification

Ref No.	Merits	Drawbacks
[8]	Use NIR combined with ML	Types of raisins used are different from the proposed
[9]	Used color feature	Color feature is not available unless images are used
[10]	Binary classification of Raisin, Morphological features used	Only 86% accuracy
[15]	Many classification methods are used	Accuracy of all below 89%
[16]	Classification of good and bad raisin grains Texture features used	Only classify good and bad raisins not raisin types.
[17]	Automatic grading of raisin	Color feature is not always available

3. DEEP NEURAL NETWORKS

Deep neural networks are extensions of traditional neural networks. Due to the importance of these networks, they are used in deep learning models. These models prove high efficiency in machine learning tasks such as classification, clustering, and regression [18]. Classification is often the most common machine learning task. Deep networks differ from traditional neural networks because deep networks have many hidden layers and many neurons aiming to model the input dataset perfectly. Deep learning uses backpropagation methods to adapt the weights of the deep network. On each neuron of the deep network, a decision is taken using a nonlinear activation function. Many activation functions, such as the Sigmoid, ReLU, and Tanh functions, may be used. The Sigmoid function is given by Eq. 1 as [19]:

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (1)$$

Also, the Tanh function is described by Eq. 2 as:

$$\mathrm{Tanh}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (2)$$

There is a relationship between both Tanh and Sigmoid functions. The Tanh could be derived from the Sigmoid function using Eq. 3 as [19]:

$$\text{Tanh}(x) = 2\sigma(2x) - 1 \quad (3)$$

However, both Tanh and Sigmoid suffer from the vanishing gradient problem when used in the hidden layers of the deep network. Alternatively, the ReLU function is the most appropriate function that may be used in the hidden layers of the neural network. The ReLU function could be described in Eq. 4 as:

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (4)$$

For binary classification problems that include only two classes, such as the problem we address here (raisin grain classification), the loss function used is the binary cross entropy (BCE) function. It is given by Eq. 5 as:

$$BCE = -\frac{1}{n} + \sum_{i=1}^{n} (y_i \log(\hat{y}) + (1 - y_i)(1 - \log(\hat{y}))) \quad (5)$$

Where y_i is the desired result, while $\hat{y}$ is the actual result. During network training, the loss is used to update the weights of the neural networks in the backpropagation process using the gradient descent methods. The gradient descent methods are popular optimization methods (optimizers) used in deep networks [20].

Some important parameters are used during the training process; for example, the training dataset is divided into batches. The batch size controls the number of training samples used before updating the internal parameters of the model. Also, the number of epochs determines the number of iterations performed on the dataset during the training process. Optimizer is an essential factor in deep network performance. The most efficient stochastic gradient descent algorithm is called Adam. Adam has good results in many problems, is easy to implement, requires little memory, and is computationally efficient.

Some other hyper-parameters are user-controlled, such as the number of hidden layers, the number of neurons, the activation function, and the loss function used. These parameters highly affect the performance of the deep network [18], [19]. There are many architectures of deep networks. The choice of the deep architecture depends on the form of the dataset. The most common deep network architecture is the standard architecture which consists of an input layer, many hidden layers, and an output layer. This network is suitable for tabular datasets such as the dataset addressed in this paper. For spatial datasets, such as images, the convolutional neural networks (CNNs) are more appropriate as it detects spatial correlations between pixels. However, the recurrent neural networks (RNNs) are more appropriate for text processing and time series analysis.

Due to the tabular representation of the raisin grains dataset, this paper proposes a standard deep learning architecture for raisin grains classification. The proposed work tests different network design scenarios with different hyper-parameters configurations suggesting the best model for solving the raisin grain classification problem.

4. THE PROPOSED MODEL

This section presents the proposed deep learning model. This model aims to classify the raisin grains into two classes of raisin grains: Besni and Kecimen. The classification is carried out using a standard deep neural network architecture. The proposed model is illustrated in Fig. 2. In the proposed model, some numeric raisin features are preprocessed and analyzed by using correlation and principal component analysis (PCA).

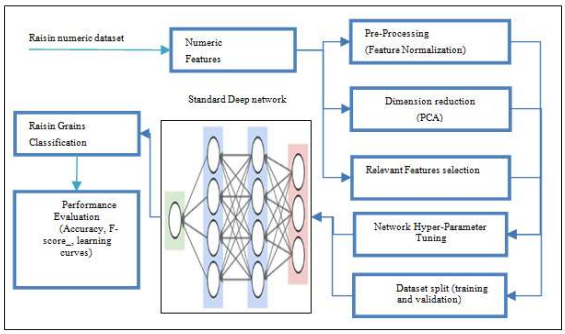

Figure 2: the structure of the proposed classification model

Then, the preprocessed features are normalized to balance the weights of the features. The preprocessed features are fed to a standard fully connected deep network architecture. For classification. This network is regularized and adjusted to model the input data. Also, the deep network hyperparameters are configured and adjusted including the number of hidden layers and the number of neurons. Also, some performance metrics are evaluated and compared with other works in the literature. The details of the proposed model are explained in the following subsections.

4.1. Dataset

The dataset used in this paper is developed by [10]. It is publicly available at [12]. The dataset

consists of a total of 900 records of raisin grains. The dataset represents the two types of raisin grains: Besni and Kecimen, with an equal number of records. Initially, the images of raisin grains were captured, preprocessed, and segmented [10]. Then, seven morphological features are extracted from the images [10]. These features are Area, Perimeter, Major-Axis-Length, Minor-Axis-Length, Eccentricity, Convex-Area, and Extent [10]. The dataset features distributions are analyzed for better understanding of the dataset. Fig. 3 shows the visualization of each feature of the dataset.

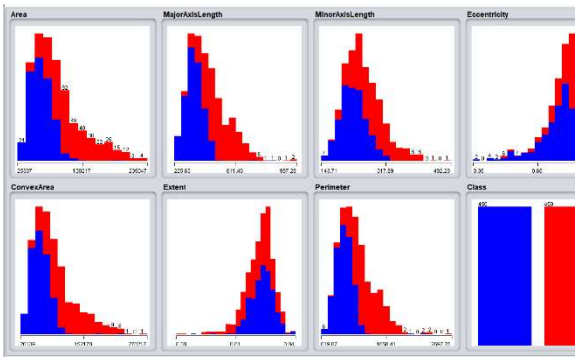

Figure 3: Visualization of the dataset

It is clear from Fig. 3 that the features proposed by [10] are non-discriminative. This issue has a negative effect on the classification accuracy. Another drawback of the public dataset [10], [12] is that; the authors in [10], [12] converts the original color raisins images to binary images. However, the color of a raisin is an important feature that should be considered in the feature extraction process. Also, the curvature of the raisin grains should be extracted. These drawbacks of the dataset [10] negatively affect the classification performance.

4.2. Preprocessing

Preprocessing aims to prepare the dataset to the classification process. Fig. 3 shows overlaps of the dataset features. A further analysis of the features shows a significant variation in features averages and standard deviation. Again, this problem decreases the classification accuracy. For example, the average of the Area feature is 87804, while the average of the Extent feature is only 0.7. These results suggest an additional preprocessing of the dataset features. So, this paper performs an essential preprocessing operation before classification. The proposed preprocessing is the normalization of all features. The normalization of features is given by:

$$z = \frac{x-\mu}{\sigma} \quad (6)$$

Where x is the original feature value, μ is the average feature value, z is the normalized value, and σ is the standard deviation of the feature. The interpretation of this step is that the neural networks work well with the normally distributed values N (0, 1). Additionally, features correlations are examined in the dataset. Table 2 shows the correlation matrix of the features used in the dataset. For simplicity, the features are named from F1 to F7, representing the seven previously mentioned features. It Is clear from Table 2 that there are very high correlations between most of the features. For example, there is a very high correlation between the first, third, fifth, and seventh features. This high correlation may negatively affect classifiers' performances. A possible solution to this problem may include removing some highly correlated features. Another solution is to perform a dimension reduction technique, such as the principal component analysis (PCA), to decrease the number of dimensions used. Also, the sequential feature selection may be used to evaluate the worth of a subset of attributes by considering their predictive ability.

Table 2: The correlation matrix of the Raisin dataset

	F1	F2	F3	F4	F5	F6	F7
F1	1						
F2	0.93	1					
F3	0.90	0.72	1				
F4	0.33	0.58	-0.02	1			
F5	0.99	0.94	0.89	0.34	1		
F6	-0.01	-0.20	0.14	-0.36	-0.05	1	
F7	0.96	0.97	0.82	0.44	0.97	-0.17	1

Fig. 4 shows the boxplot of all dataset features. It is clear from the figure that; the number of outliers is minimal in all features, which means that

the features are internally consistent. Also, for each feature, the median of the feature is almost near the average of the feature. Also, the features are symmetric, not skewed, and not dispersed. These results support the importance of the preprocessing before classification to enhance the classification accuracy.

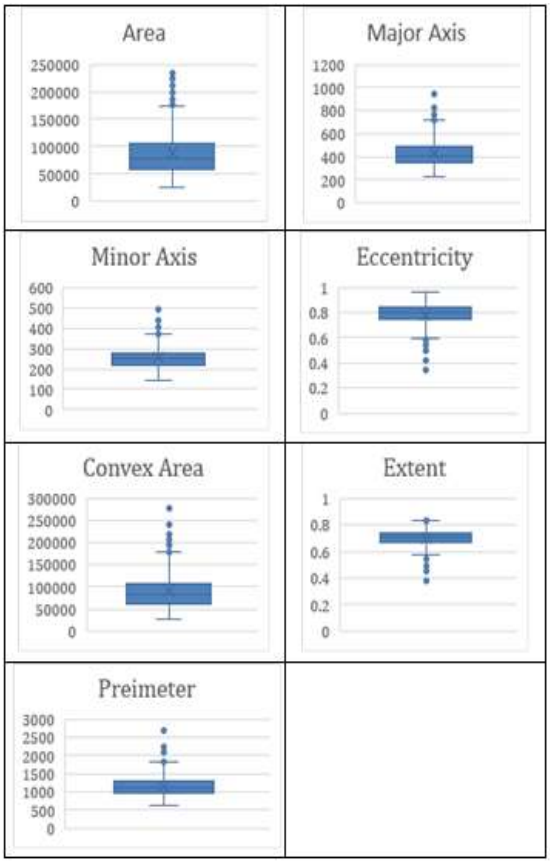

Figure 4: Boxplot of the features in the dataset

4.3. Classification

In this paper, the problem addressed is the raisin grains classification. This problem is very important in logistics as automatic and accurate classifications are required. The performances of the algorithms in the previous works are not satisfactory. Also, deep networks were not investigated for solving this problem. So, the paper proposes a standard deep neural network architecture designed for raisin grains classification. The raisins dataset used is given in a tabular form representing raisins features in columns and raisin instances in rows. So, the choice of the standard architecture is more appropriate for this dataset. The standard deep network consists of one input layer with seven inputs representing the seven raisin features. These features are firstly preprocessed before feeding them to the network. The deep network includes more than one hidden layer. These hidden layers are fully connected. The standard deep network has only one output layer with one output. The proposed deep network is tuned and optimized to give the optimal performance. Moreover, the loss function, optimizer, number of hidden layers, number of neurons in each layer, and the type of activation function are tuned before building the network.

In this paper, the stochastic gradient descent algorithm (Adam) is used as an optimizer. The loss function used is the binary cross entropy (BCE). The ReLU function is used as an activation function in all hidden layers. The Sigmoid function is used in the output layer. The initial deep network configuration is shown in Fig. 5. In this initial configuration, two hidden layers are specified. Each hidden layer includes 32 neurons. However, these configurations are only initial configuration. These configurations are changing during testing processes to determine the optimal configuration.

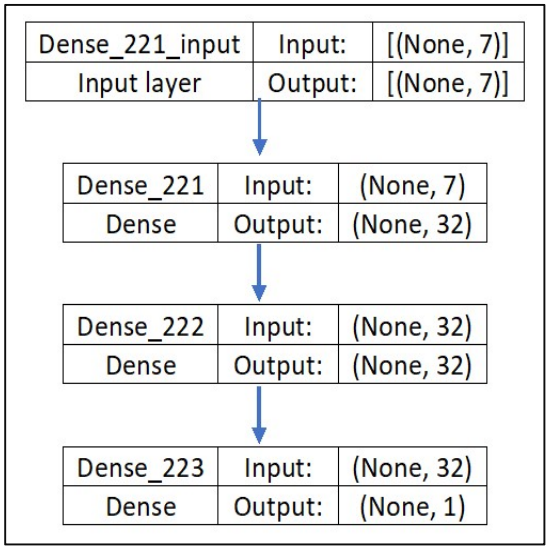

Figure 5: Initial configuration of the deep neural network

In the testing process, the dataset is divided into two sets; the training set and the validation set. So, the learning curves are depicted and analyzed. The learning curves graph is a plot that shows time on the x-axis and learning on the y-axis. This paper uses two types of learning curves: optimization learning curves and performance learning curves. Optimization learning curves show the learning loss, while the performance learning curves show the model accuracy [21]. These curves illustrate the tested model status: under-fitting, good fitting, or overfitting [18]. Underfitting occurs when a model cannot learn from the training dataset [18]. Overfitting occurs when a model learns a statistical

noise from the training dataset [18], [21]. A good fitting exists between overfitting and underfitting. Good fitting is the goal of the learning algorithms which satisfies a minimal gap between two final loss values. This gap is called the generalization gap [18], [21]. Precision, recall, F-Score, and accuracy are used for model performance evaluation. These metrics include precision, recall, F-Score, and accuracy. These metrics are given by Eq. 7-10 as:

$$\text{precision} = tp / (tp + fp) \quad (7)$$

$$\text{recall} = tp / (tp + fn) \quad (8)$$

$$F - \text{score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall}) \quad (9)$$

$$\text{accur} = (tp + tn) / (tp + tn + fp + fn) \quad (10)$$

TP is the true positive, FP is the false positive, TN is the true negative, and FN is the false negative. F-Score could be computed from both precision and recall. Since the importance of the two raisin classes is the same, F-Score is only considered in the experimental results instead of using both precision and recall. The proposed work computes accuracy beside F-Score for performance evaluation and comparison.

5. EXPERIMENTS AND RESULTS

This section presents the experimental results. Python 3 Jupiter Notebook is used for testing and graphing. The effective deep learning Keras library is used. Also, Waikato Environment for Knowledge Analysis (WEKA) and Microsoft (MS) Excel are used to visualize the dataset features and the computation of Pearson correlation coefficient. All experiments use the ReLU activation function in all hidden layers. For the output layer, the Sigmoid activation function is used. Also, the number of epochs is fixed to 100 as the convergence occurred in all trials using this number. The batch size equals 32. The model's accuracy and F-Score are computed and compared with previous work. The loss function used is cross-binary Entropy. The optimizer used is Adam. However, both the number of hidden layers and the number of neurons are considered hyper-parameters. The dataset is divided into training, validation, and testing. Table 3 shows an example of the hyper-parameters used in the deep network during the learning process.

Table 3: Default hyper-parameters configuration of the deep neural network.

Layer (type)	Output Shape	Param#
1	(None, 32)	256
2	(None, 32)	1056
3	(None, 1)	33
Total params: 1,345 Trainable params: 1,345 Non-trainable params: 0		

It is clear from the table that; the first hidden layer has 256 parameters, the second hidden layer has 1056 parameters, and the output layer has 33 parameters. So, the total number of parameters is 1345. These parameters are all updated during the learning process to construct the model. The following subsections detail the obtained results and discuss the learning curves of the proposed model in many experiments. Also, the recommended configuration of the proposed model is illustrated and explained.

5.1. Performance Testing Without Feature Normalization

Fig. 6 shows the first experiment of the deep network. In this experiment, the dataset is fed to the network without preprocessing (i.e., without normalization of the features). In this case, the average accuracy of the model is only 81.48%. The F-Score is only 78.63%. This low performance occurs due to the high variations in the features averages and standard deviations. From Fig. 6 although the model loss has decreased significantly for both training and validation, the model suffers from underfitting, as indicated by the accuracy curve of both the training and the validation set. Many fluctuations in the curve show this underfitting. In this case, the model cannot learn from the training dataset.

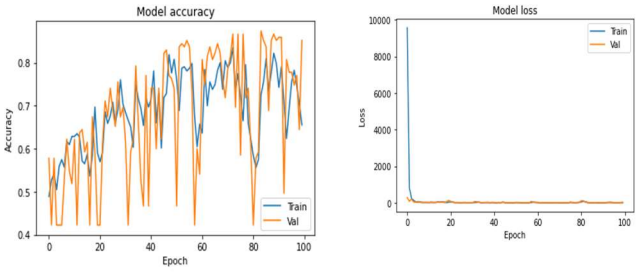

Figure 6: Accuracy and loss of the model without preprocessing; Acc:81.48%, F-Score :78.63%

5.2. Performance Testing Using PCA

To deal with the correlated features problem, two experiments are performed using principal component analysis (PCA) to decrease the dimensions of the features before using the deep network. The first PCA experiment performs features normalization followed by performing PCA. The results of this experiment are shown in Fig. 7. The accuracy reaches 91.85%, while the F-Score reaches 90.26%. Although this is a good performance, however, from the loss curve shown in Fig. 7, the model suffers from overfitting.

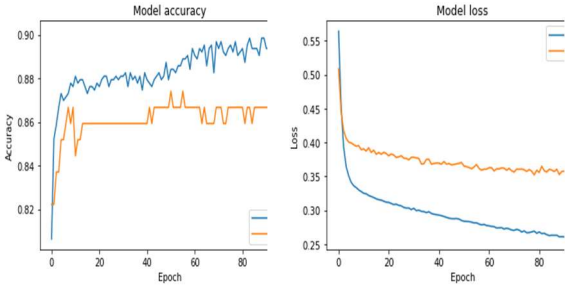

Figure 7: Experiments using PCA and normalization;Acc=91.85%, F-Score =90.26%

There is a large gap between the training curve and the validation curve. Also, the training curve decays rapidly compared to the validation curve. Also, the accuracy curve in the accuracy graph shows no more additional learning on the validation set. So, the accuracy of the validation dataset is 86% on average. Fig. 8 shows the results of the second PCA experiment. In this experiment, PCA is performed without normalization of the dataset. Here, the accuracy is only 86.66%, while the F-Score equals 86.36%. Again, an overfitting is observed from the accuracy graph as the training accuracy curve decays while there is no learning enhancement in the validation curve. Also, many fluctuations appear in the accuracy curve of the validation set, illustrated by noisy movements in curves. So, PCA has not an optimal performance as concluded from the two previous experiments. Occasionally, we try removing some dimensions of the transformed dimensions using PCA. However, there are no significant improvements in all PCA cases. So, this paper does not consider PCA in the following experiments.

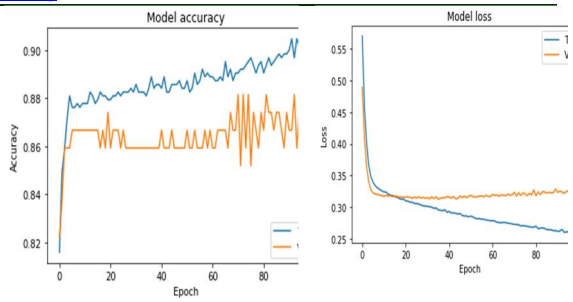

Figure 8: Experiments using PCA without normalization;Acc=86.66%, F-Score =86.36%

To deal with the features correlation problem, this paper tests the sequential feature selection algorithm [22], [23]. WEKA suggests four candidate features using this algorithm: Major-Axis-Length, Eccentricity, Convex-Area, and Perimeter [22], [24]. However, when using these candidate features only in the classification, the accuracy decreased to 85% on average. The F-Score is only 83% on average. These results lead to consider all features of the dataset despite the high correlation between them.

5.3. Performance Testing Using Network Regularization

Model complexity is a possible one cause of overfitting. Regularization may solve the overfitting problem of the deep network [18]. This complexity could be simplified using neuron dropout [11]. Dropout means that a neuron may be dropped during training with some probability P. In this case, the neural network becomes simpler. However, a balance is required in the regularization process to avoid underfitting. Fig. 9 shows an experiment to deal with overfitting. In this experiment, the deep network has four hidden layers, each hidden layer containing 1000 neurons. Lot of P values are tested ranging from 0.2 to 0.9. However, the performances of all trials are not good, as shown by Fig. 9. The figure shows an experiment using P=0.5 as an example. In this case, the accuracy only equals 84.44%, while the F-Score equals 82.92%. The interpretation of this result is that, due to the small size of the dataset, simpler networks may be appropriate to model this dataset. So, regularization is not always an appropriate solution for overfitting especially when using datasets with small sizes.

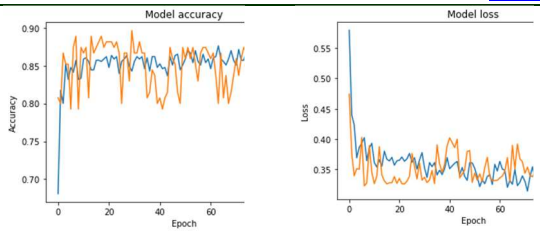

Figure 9: Regularization using dropout, P=0.5;Acc=84.44%, F-Score =82.92%

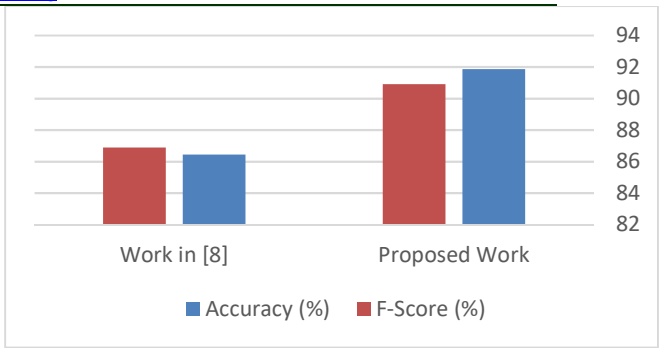

Figure 10: Performance comparison of the proposed work and work in [10]

5.4. Performance of The Recommended Model

The previous results lead to concluding that features normalization is an important step that improves classification accuracy. Also, applying feature selection and PCA are not appropriate on this dataset. So, the recommended model proposed by this paper is to performs features normalization without performing dimension reduction. Also, the recommended model determines the appropriate network configuration and hyper-parameters. So, this work analyzes different scenarios to specify the optimal hyper-parameters to achieve the optimal performance.

Table 4 shows the tested different hyper-parameters and their resulting performance. The first column of the table shows the number of hidden layers tested. The second column shows the number of neurons in each hidden layer. Both accuracy and F-Score [25] are shown in the third and fourth columns, respectively. From the table, the highest accuracy equals 91.85%. This accuracy was achieved when using two hidden layers in the network. Each hidden layer contains 20 neurons. Also, the highest F-Score is achieved in this case which equals 90.90%. Interestingly, other tabulated results are very encouraging.

Comparing these results with the previous work described in [10], the authors in [10] achieved an accuracy of only 86.44%. The accuracy of the proposed algorithm exceeds 91%. Also, the F-Score of the proposed algorithm outperforms that presented in [10]. The F-Score shown in [10] is only 86.88%. Fig. 10 shows a performance comparison of the proposed model and the work proposed in [10]. It is clear from the figure the superiority of the proposed model compared to the work proposed in [10].

Additionally, the authors in [10] evaluate three classifiers: Linear Regression (LR), Multilayer Perceptron (MLP), and SVM. SVM achieves the best accuracy and the best F-Score as indicated from their results. Also, SVM outperforms MLP in their proposed model. This indicates that the MLP they used should be tuned better. Although the authors in [10] state the number of hidden layers in their model, the number of neurons is not stated, which is an important hyper-parameter.

An example of one performance curve of the recommended configuration is shown in Fig. 11. It is clear from the figure that, besides the significant accuracy and F-Score values achieved, the learning curves are also very smooth. From the model accuracy graph, the accuracies of the training and the validation sets are almost semi-identical. It is also clear from the loss curve that the losses of both training and validation are minimal, and the generalization gap is also minimal. So, this paper achieves a good-fit model. Thanks to the preprocessing step and to the hyper-parameter tuning step.

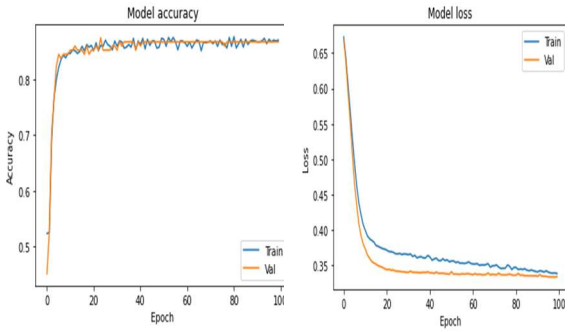

Figure 11: The good-fit proposed model;Acc=89.62%, F-Score =90.14%

Table 4: Hyper-parameters tuning of the deep network

#Hidden	Neurons/Layer	Accuracy	F-Score

Layers		(%)	(%)
2	32	89.70	89.97
	20	91.85	90.90
	10	85.92	85.27
3	32	88.88	87.39
	20	88.14	88.40
	10	85.18	86.95
4	32	81.48	82.26
	20	89.62	90.00
	10	85.18	84.61
5	32	88.14	85.96
	20	87.40	88.74
	10	88.88	87.80

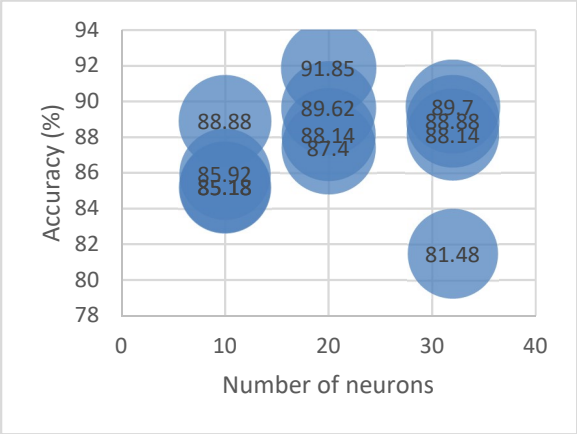

Figure 12: Model Accuracy Based on The Number of Neurons

Finally, Fig. 12 shows the accuracy of the recommended model considering the number of neurons used. It is clear from the figure that the best accuracy was achieved when using a slightly middle number of neurons in each hidden layer (i.e., 20 neurons in this case).

5.5. Discussion

Agricultural product classification is one of the challenging problems in machine learning unless suitable features and classifiers are used [7]. This paper proposes a standard deep network model for raisin grains classification. The proposed work uses the dataset proposed in [10], [12] for classifying raisins into two classes: Besni and Kecimen. The dataset is a tabular dataset consists of numeric values representing different raisins features. The dataset used consists of seven highly correlated features. All features are derived from pixel counting approaches such as area and eccentricity.

The distribution of the individual features shows that these features are not discriminative features of the two classes. Also, the correlation matrix shows that all features in the dataset are highly correlated. Possible solutions to the correlated features problem are feature selection and dimension reduction algorithms. Results obtained by testing feature selection algorithms indicate that the model performance is not satisfactory. This means that using highly correlated features may enhance classifier performance. Also, using PCA is not optimal on this dataset. By using PCA, some fluctuations occur in the accuracy curve despite removing many combinations of PCA dimensions. This may be explained as the number of features in the dataset is quietly small. That is, PCA may be useful in datasets having large number of features. Also, features distribution differences suggest feature normalization as a pre-processing step before feeding the normalized features to the deep network. The previous results lead to recommend a standard deep model that uses normalized features as input without performing any dimension reduction technique. The recommended deep model also uses all features of the dataset without any feature selection process. Also, the optimal hyperparameters of the model are determined.

The results of the proposed recommended model show enormous enhancements to the model performance, and a good fit model is achieved. Also, the results show that tuning the model hyper-parameters essentially enhances the deep network performance. Also, the proposed results indicate that regularization may not an appropriate solution to model overfitting, especially with small-size datasets. In this case, a small deep network is more appropriate. The proposed results also show that the proposed algorithm outperforms other algorithms in the literature using different testing scenarios. The proposed accuracy exceeds 91% in highly correlated and non-discriminative features. Besides the considerable performance enhancement achieved, the learning curves are also very smooth. The generalization gap is minimal. Also, the loss is minimal, whereas the accuracy is maximal in both training and validation sets, as indicated by the performance graphs. Regarding the number of hidden layers, the proposed recommended model suggests using 2 - 4 hidden layers on this dataset. The best accuracy was achieved using a middle number of neurons in each hidden layer (i.e., 20 neurons in each hidden layer). However, the performance results may be enhanced if more appropriate features are added to the dataset from the raisin images. For example, the color and curvature of the raisin grains are essential features that should be added to the dataset. In this case, some of the current, non-discriminating, features

may be removed. So, the model performance may be enhanced.

6. CONCLUSION

This paper proposes an efficient deep learning model for raisin grains classification. The proposed model uses feature normalization as a pre-processing to deal with the highly correlated, and non-discriminative, dataset features. The proposed model tunes the hyper-parameters of the deep network to achieve the optimal design of the deep network. The proposed model outperforms others in the literature using many testing scenarios. The proposed accuracy and F-Score exceed 91% using the non-discriminative dataset features. To our knowledge, the proposed model achieves the highest performance on the same dataset compared to other works. There are some important conclusions from the experiments conducted. First, using PCA or feature selection does not enhance the model performance on this dataset. Second, neural network regularization may not be an effective solution for overfitting, especially for small size datasets. Third, features normalization is a crucial pre-processing step for this dataset. Finally, the choice of optimal network configurations deeply affects the model performance. Future work may include evaluation of the proposed model complexity. Also, additional new features, such as color and texture features, may be evaluated to enhance the model performance. Although the standard deep models are more appropriate for tabular datasets, other deep models such as CNNs may be evaluated in future work. In this case, tabular data may be treated as images. However, the limited availability of robust and large raisin datasets is a main challenge facing current raisin classification research.

REFERENCES

[1] A. Rana, "Supply chain optimization using machine learning," *International Research Journal of Engineering and Technology*, vol. 7, no. 6, pp. 7110-7114, June 2020.

[2] E. Tirkolaee, S. Sadeghi, F. Mooseloo, H. Vandchali and S. Aeini, "Application of machine learning in supply chain management: a comprehensive overview of the main areas," *Mathematical Problems in Engineering*, Hindawi, Vol. 2021, no. 2, 2021.

[3] R. Konyaitė, A. Brazinskaitė, I. Šaltisc and T. Krilavičius, "Machine learning approaches for customs fraud detection," in the *26th Information Society and University Studies Conference*, Kaunas, Lithuania, pp. 54-63, April 2021.

[4] V. Kumar and S. Veerappa, "Raisins grade detection using image processing technology," *International Journal of Science and Research*, vol. 6, no. 5, pp. 2319-7064, 2017.

[5] J. Che, Y. Xing and L. Zhang, "A comprehensive solution for deep-learning based cargo inspection to discriminate goods in containers," in the proceeding of the *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, USA, pp. 1287-12877, 2018.

[6] M. Altalak, M. Ammaduddin, A. Alajmi and A. Rizg, "Smart agriculture applications using deep learning technologies: a survey," *Applied Sciences*, MDPI vol. 12, no. 12, pp. 5919- 5939, 2022.

[7] D. Batool, M. Shahbaz, H. Asif, K. Shaukat, T. Alam *et al*., "A hybrid approach to tea crop yield prediction using simulation models and machine learning," *Plants*, MDPI, vol. 11, no. 25, pp. 1925-1945, 2022.

[8] Y. Zhang, Y. Yang, C. Ma and L. Jiang, "Identification of multiple raisins by feature fusion combined with NIR spectroscopy," *PLoS One*, vol. 17, no. 7, 2022.

[9] J. Guo, C. Chen, C. Chen, E. Zuo, B. Dong *et al.*, "Near-infrared spectroscopy combined with pattern recognition algorithms to quickly classify raisins," *Scientific Reports*, Nature, vol. 12, no. 1, 2022.

[10] I. Cinar, M. Koklu and S. Tasdemir, "Classification of raisin grains using machine vision and artificial intelligence methods," *Gazi Journal of Engineering Sciences*, Vol. 6, no. 3, pp. 200-209, Dec. 2020.

[11] M. Aamir, Z. Rahman, W. Abro and M. Tahir, S. Ahmed, "An optimized architecture of image classification using convolutional neural network", *International Journal of Image, Graphics and Signal Processing(IJIGSP)*, vol. 11, no. 10, pp. 30-39, 2019.

[12] Machine Learning Repository, https://archive.ics.uci.edu/ml/machine-learning-databases/00617/, Last accessed 10 Oct. 2022.

[13] K. Kiratiratanapruk, P. Temniranrat, W. Sinthupinyo, P. Prempree, K. Chaitavon *et Al.*, "Development of paddy rice seed classification process using machine learning techniques for automatic grading

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

machine", *Journal of Sensors*, Hindawi, vol. 2020, 2020.

[14] E. Benes, D. Bajusz, A. Gere, M. Fodor and A. Rácz, "Comprehensive chemometric classification of snack products based on their near infrared spectra", *LWT - Food Science and Technology*, vol. 133, no. 1, 2020.

[15] Kaggle Raisin Competition, https://www.kaggle.com/datasets/muratkokludataset/raisin-dataset/code, Last accessed 10 Oct. 2022.

[16] M. Khojastehnazhand and H. Ramezani, "Machine vision system for classification of bulk raisins using texture features," *Journal of Food Engineering*, Elsevier, vol. 271, no. 1, April 2020.

[17] K. Shinde and B. Patil, "Sorting of raisins using computer vision approach," *International Research Journal of Engineering and Technology*, vol. 4, no. 6, pp. 2540-2545, 2017.

[18] I. Goodfellow, Y. Bengio and A. Courville, "Deep learning," Illustrated Edition, *Adaptive Computation and Machine Learning series*, USA, MIT Press, Chapter 5-7, pp. 95-263, 2016.

[19] C. Higham and D. Higham, "Deep learning: an introduction for applied mathematicians," *Siam Review*, vol. 61, no. 4, pp. 860–891, 2019.

[20] H. Choi, C. Jung, T. Kang, H. Kim and I. Kwak, "Explainable time-series prediction using a residual network and gradient-based methods," *IEEE Access*, IEEE, vol. 10, pp. 108469 - 108482, 2022.

[21] J. Brownlee, "Deep learning with python develop deep learning models on theano and tensorflow using keras," 1st Edition, *Machine Learning Mastery*, Electronic Book, Chapter 8, pp. 51-55, 1st edition, 2017.

[22] T. Mohamed, "Efficient breast cancer detection using sequential feature selection techniques," in the proceeding of IEEE 7th *International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, pp. 458-464, 2015.

[23] T. Trang, H. Cam,V. Nalluri, L. Chen and Y. Yang, "IS-DT: A new feature selection method for determining the important features in programmatic buying," *Big Data Cognitive Computing*, MDPI, vol. 6, no. 4, pp. 118-137, 2022.

[24] K. ElDahshan, A. AlHabshy and L. Mohammed, "Filter and embedded feature selection methods to meet big data visualization challenges," Computers, Materials & Continua (CMC), vol. 74, no.1, pp. 817–839, 2023.

[25] Samy S. Abu-naser, Basem S. Abunasser, "The Miracle of Deep Learning in the Holy Quran", Journal of Theoretical and Applied Information Technology, Sept. 2023 -- vol. 101. no. 17-- 2023