SELCUK UNIVERSITY JOURNAL OF ENGINEERING SCIENCES

Journal homepage: http://sujes.selcuk.edu.tr

ISSN: 2757-8828

International Open Access

Volume 21
Issue 02

August, 2022

Research Article

Identification of Chicken Eimeria Species from Microscopic Images by Using Convolutional Neural Network Method

***Zeki Kucukkara** [a,*], **Ilker Ali Ozkan** [a], **Sakir Tasdemir** [a]*

[a]*Selcuk University, Faculty of Technology, Department of Computer Engineering, Konya, Turkey*

ARTICLE INFO

Article history:
Received 11 May 2022
Accepted 07 August 2022

Keywords:
Convolutional Neural Nets
Deep learning
Disease Detection
Image Classification

ABSTRACT

Eimeria is a parasite that lives in the intestinal, bile duct, and liver tissues of various domestic animals such as rabbits, chickens, geese, ducks, cattle, pigs, cats, and dogs. Due to these conditions, these parasites can spread rapidly, negatively affect animal productivity, and lead to deadly results. For this reason, it is vital to determine the disease early and prevent its spread at an early age. Because of these parasites' diversity, complexity, and similarity, a system automatically analyzes them using microscopic images is needed. A model was developed to address this problem using Convolutional Neural Networks to predict seven different types of noise on microscopic images. In the developed methodology, the average accuracy rate was 93.85%. This model developed to detect seven different types of parasites has shown that it can be used successfully.

1. Introduction

Detection of poultry diseases and accordingly preventing diseases is essential in preventing commercial losses. Automatic detection of diseases with image processing and deep learning methods is a widely used method. According to the report published by the United Nations Agriculture and Food Organization in 2010, the Eimeria parasite infects animals and causes the death of poultry in a short time. Detecting the Eimeria protozoan parasite in poultry is difficult and causes an epidemic disease called coccidiosis [1][2]. In Eimeria, each chicken breed has more than one species. Because each chicken breed has more than one species, it can spread rapidly and cause diseases that negatively affect animal productivity and increase animal deaths [1]. Therefore, it is critical to identify the disease and stop its spread as soon as possible. This study aims to improve classification performance through deep learning algorithms by analyzing microscopic images for seven different Eimeria species. With this study, we will contribute to the literature with a different CNN model. Literature studies show that few studies have been done in recent years using the CNN method to increase classification accuracy. Some of these studies are as follows: Büyükyılmaz et al., with the preprocessing software they developed in Python software, detected the cells along the borders of the 4402 Eimeria image, deleted the background of the image, and left the image of the grey cell on a black background; They determined the size of the largest cell and made all images the same size. Their second application classified seven different Eimeria species using the multilayer perceptron artificial neural network algorithm. With this study, they found the accuracy rate to be approximately 83.75% [1]. Castañón et al., using three group features of 3891 images, classified seven Eimeria species with approximately 85.75% accuracy with the help of Gaussian function and Bayesian classifier [3]. Monge et al. developed a 6-convolution model using the 4485 images CNN algorithm. This developed model classified seven different Eimeria species with an average accuracy of 90.42% [4]. Abdalla et al. Eimeria parasite images; Since they differ in shape, size, and texture, they classified the differences in these features by measuring them. For this, they presented an approach developed by analyzing pixel-based features rather than image morphological features. After 3891 images were grey-scaled, the images were made binary with the determined threshold value. In the continuation of the process, they extracted the features of the image by detecting the cell edges with the Moore-Neighbor Tracing algorithm. They used the k-nearest neighbour algorithm in the classification step and performed a 5-fold cross-validation for validation. Their proposed method correctly classified seven different Eimeria species with 82% per cent accuracy [5].

* Corresponding author. E-mail address: *zekikucukkara@gmail.com*

In addition to image processing-based studies in this field, few studies have used the CNN method to increase classification accuracy in recent years. This study makes the following contributions.

- A deep learning model was created for the early detection and classification of seven different parasite types in chickens.
- A different CNN model from the literature has been developed to solve the problem.
- Microscopic images were classified with the CNN model, and a detailed analysis was performed.

The rest of the study is structured as follows: The materials and methods used in the study are described in the second section. Dataset preparation, CNN architecture and performance metrics are explained in this section. The experimental results are presented in the third section. The fourth section includes conclusions and recommendations.

2. Material and Method

2.1. Dataset

The dataset used in this study was prepared and presented to researchers by the Laboratory of Molecular Biology of Coccidia at the Department of Parasitology of the Institute of Biomedical Sciences and the Cybernetic Vision Research Group at The Institute of Physics. The types and numbers of images that make up the data set are shown in Table 1 [6]. The dataset contains digital microscopic images for different Eimeria species. The dataset can be downloaded as a complete microscopic image and cell fragmented images. In this study, fragmented images were used as cells. The chicken dataset was created using local chickens' samples for seven Eimeria species. Figure 1 shows photomicrographs of seven Eimeria species of domestic poultry [3].

Table 1. Number of images taken from chickens by parasite class

Species Name	Number of Training Images
Eimeria Acervulina	743
Eimeria Brunetti	443
Eimeria Maxima	361
Eimeria Mitis	826
Eimeria Necatrix	502
Eimeria Praecox	898
Eimeria Tenella	696
TOTAL	4469

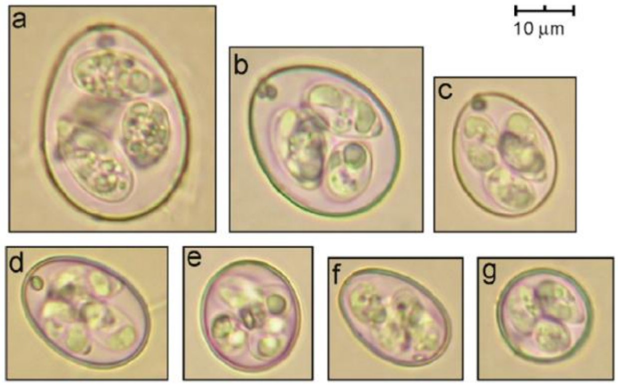

Figure 1. Photomicrographs of seven Eimeria species of domestic poultry. Examples: (a) Eimeria maxima, (b) Eimeria brunetti, (c) Eimeria tenella, (d) Eimeria necatrix, (e) Eimeria praecox, (f) Eimeria acervulina, and (g) Eimeria mitis.

2.2. Deep Learning Architecture

This study used the CNN model to train and classify the Eimeria dataset. CNN architecture is a mathematical structure made up of three types of layers. There are three of them: the convolutional layer, the pooling layer, and the fully connected layers. The first two layers, convolution and pooling, extract features, while the third layer, fully connected, provides the final output, such as classifying the extracted features. The convolution layer, consisting of a series of mathematical operations, a linear operation type, plays an essential role in CNN. An optimizable feature extractor, a small parameter grid called the kernel, is applied at each image position. Because a feature can appear anywhere in the image, CNNs are very efficient for image processing. As one layer feeds its output to the next, extracted features can become hierarchical and progressively more complex. Forward propagation is the process by which input data is converted to output data via these layers. Backpropagation is an optimization algorithm used to optimize kernel parameters. Outputs through these optimized parameters cut the difference between precise accuracy labels. Training is the process of optimizing these parameters. This section describes the convolution and pooling operations for 2D CNN. Similar operations can be carried out for 3D CNN [7][8].

2.2.1. Convolution layer

A convolution layer is made up of both linear and nonlinear operations. This is a critical feature extraction component of the CNN architecture [9]. The convolution layer has two functions: convolution operation and activation.

2.2.1.1. Convolution

Convolution is used to extract features. A multidimensional array of numbers known as tensors runs along with the input. This array is treated with a small array of numbers known as cores. This is an example of a linear operation. The kernel is processed with each element along with the input tensor array. It is summed to obtain the output tensor sequence, characterized as a feature map. This procedure generates a random number representing different properties of the input tensors. It is iterated by applying many cores to create the feature map. As a result, different kernels can be considered feature extractors. Two key hyperparameters define the convolution operation. These are the size and the number of cores [7]. The training process in the convolutional layer works on a specific training dataset. It identifies the best working kernels for a given task during the training process. Kernels are convolution layer parameters that are learned during the training process. Before the training process can begin, certain hyperparameters must be set. These are the size of the cores, the number of cores, the padding and the number of steps. The list of layers, parameters and hyperparameters is given in Table 2.

Table 2. CNN parameters and hyperparameters

Layer	Parameters	Hyperparameters
Convolution layer	Kernels	The number of kernels, the size of the kernels, the padding, the stride, and the activation function
Pooling layer	None	Method of pooling, filter size, stride, and padding
Fully connected layer	Weights	The number of weights and the activation function
Others		Model architecture, learning rate, optimizer, batch size, loss function, regularization, epochs, dataset splitting and weight initialization

2.2.2. Pooling Layer

The pooling layer is employed to reduce the number of learnable parameters and focus on more important features while ignoring redundant features. It is intended to reduce the size of feature maps, thereby reducing the processing power required. It should be noted that none of the pooling layers has a learnable parameter. Filter size is hyperparameters similar to convolution operations in step and pad pooling operations. Generally, the method used in the pooling layer is the operation of selecting the maximum value in the specified frame. In addition, pooling layers perform functions such as taking the mean value and selecting the minimum value in the specified framework [7]. In addition, pooling layers perform operations such as averaging and selecting the minimum value within the specified framework [7].

2.2.3. Fully connected layer

The fully connected layer flattens the output of the convolution and pool layer. That is, it is converted to a one-dimensional array of numbers or vectors. The one-dimensional array is attached to one or more fully connected layers. In this layer, each input is linked to each output with a learnable weight. The feature map is created in the convolution layer and sub-sampled in the repository layer. The generated subsample feature map is converted from matrix form to flat vector for classification. Each element of the transformed vector is mapped by a subset of layers completely dependent on the final outputs. The number of outputs on the fully connected layer is the same as the number of classes [7].

$$\partial(y)_j = \frac{e_j^y}{\sum_{k=1}^{K} e_k^y}, j = 1,2,\ldots,K \qquad (1)$$

2.3.3.1 Dropout and Last layer activation function

Sometimes CNN memorizes during the training process. This layer is used to prevent the network from memorizing. The basic logic applied in this layer is removing some network nodes. Unlike the others, the activation function applied to the last fully connected layer must be selected according to each task. Classification is done in this layer. The softmax classifier is mostly preferred because of its success. In classification, the function produces an output with a certain value in the range of 0 and 1. The output that produces a result close to 1 is understood to be the class predicted by the network [7][17][18]. Especially in solving the multi-class classification problem, the probability of finding the output value in each class is determined. As seen in Figure 2 and equation (1), K indicates the total number of classes, j any class number, and y is the value of the neuron before the softmax operation. The probabilistic output values of which class the test data can take place are calculated with softmax. The probability values of the classes in which test data can be found are given as the output of the function [7].

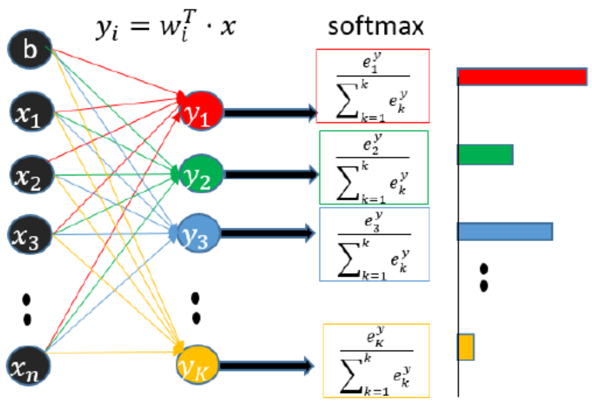

Figure 2. Multi-class classification with softmax function

2.3. Performance Metrics

Classification metrics evaluate the performance of a model and tell how good or bad the classification is. The success of the models created in the classification problems is based on the number of correct predictions obtained from entire predictions. This information does not just give the accuracy of the classification. It is to evaluate whether the prediction results of a model are good using a confusion matrix. The performance of the classification model of the confusion matrix test data is measured using four parameters. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the four types [10],[11]. TP: It denotes the number of predictions in which the classifier correctly predicts the positive class to be positive. FP: It denotes the number of predictions in which the classifier predicts the negative class as positive. TN: It denotes the number of predictions in which the classifier correctly predicts the negative class to be negative. FN: It denotes the number of predictions in which the classifier predicts the positive class as negative [16]. As shown in Table 3, these four values form the classification complexity matrix [12]. The total number of data is given by n=TP + TN + FP + FN [13]. Criteria such as accuracy, error rate, sensitivity (recall), specificity, precision, and f-score were computed to assess the model’s success, as shown in Table 4 [14].

Table 3. Confusion matrix and binary class representation

		Predicted Class	
		Positive	Negative
True Class	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Table 4. Success criteria and formulas

Accuracy	$\frac{TP + TN}{TP + FP + FN + TN}$
Sensitivity (recall)	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{FP + TN}$
Precision	$\frac{TP}{TP + FP}$
f-score	$\frac{2 * \text{precision} * \text{sensitivity}}{\text{precision} + \text{sensitivity}}$

Accuracy: Expresses the total number of records classified correctly by the classifier [15].

Sensitivity: Expresses the proportion of true positive data predicted as positive [15].

Specificity: Expresses the ratio of true negatives to total negatives [15].

Precision: Expresses the number of correct positive predictions [15].

F-score: Mathematically, it is the harmonic mean of precision and recall [15].

3. Results

The experiments were carried out on a computer workstation with an Intel® Xeon® Gold 6226R CPU @2.90 GHz, an NVIDIA RTX8000-24Q GPU, and 64 GB of RAM. The created model is tested with different parameters to train images using deep learning architecture. The model and run application have been modified numerous times to find the best model. Furthermore, the activation and optimization algorithms significantly impacted the results. These algorithms were chosen by trial and error in this study. The layers of the CNN are created, as shown in Figure 3.

In the input layer, images with 299x299 pixels are used. The first layer is 3x3 kernel size and consists of 16, 32, 64, 128, 256, 512, and 1024 convolution masks in layers, respectively. Relu and 2x2 kernel size MaxPooling were applied as activation functions in each layer. It is converted into a one-dimensional array in the eighth layer. It has been reduced to 1024 in the eighth layer, 512 in the ninth layer, 32 in the tenth layer, and 7 in the eleventh layer. The avoidance value between the eighth and ninth layers was 0.4, the avoidance value between the ninth and tenth layers was 0.5, and the avoidance value between the tenth and eleventh layers was 0.4. The output layer has seven neurons, which corresponds to the number of classes. In this layer, Softmax is used as the activation function.

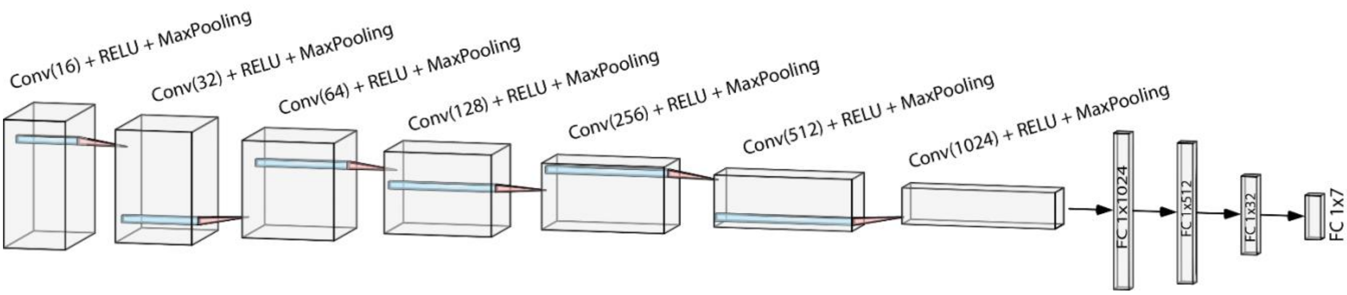

Figure 3. Layers of CNN used in the application

In the study, the images in the data set were resized as 299 x 299 in the preprocessing step. The dataset was split 20:80 for validation. For training, 3575 images were used, and 894 images were used for validation. The adam function was chosen as the learning algorithm. Table 5 displays the hyperparameters with the highest value (93.85%) obtained in the tested results.

Table 5. Hyperparameters and values used in the model

Hyperparameter	Value
InitialLearnRate	1e-04
MaxEpochs	100
MiniBatchSize	128
ValidationFrequency	30

Figure 4 gives the Accuracy and Loss values of the Training data in each epoch, the Confusion Matrix in Figure 5, and Table 6 shows the average accuracy, sensitivity (recall), specificity, precision, and f-score values.

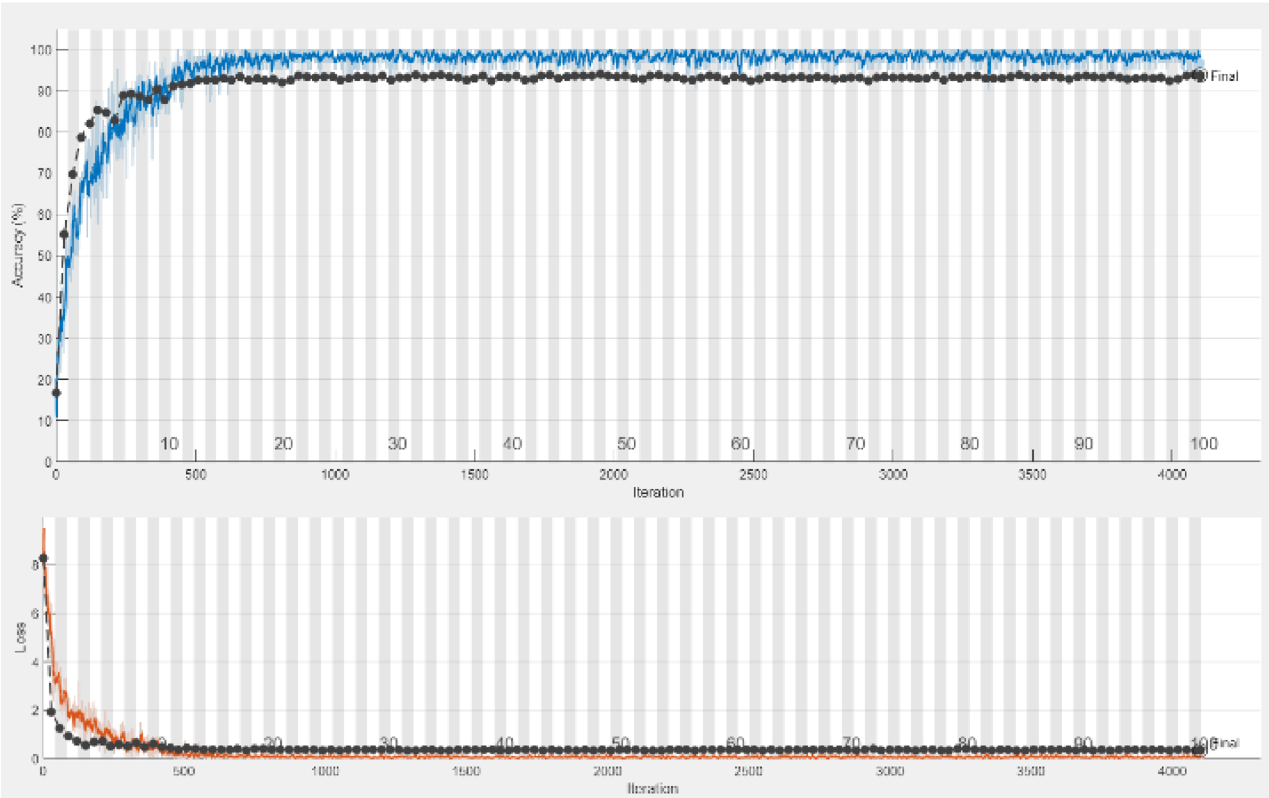

Figure 4. Accuracy and loss values of training data at each epoch

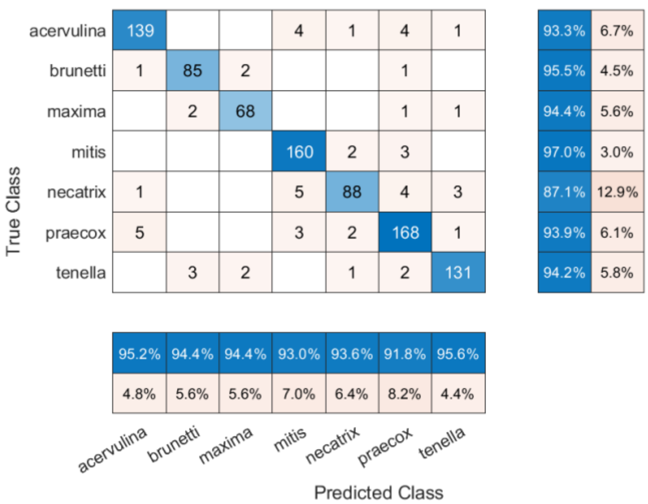

Figure 5. Confusion Matrix

Table 6. Accuracy, Sensitivity, Specificity(recall), Precision, and f-score value

Parameter	Value
Accuracy	0.98242
Sensitivity (recall)	0.93634
Specificity	0.98951
Precision	0.94023
f-score	0.93802

4. Conclusions

Eimeria is a parasite that lives in the intestinal, bile duct, and liver tissues of various domestic animals such as rabbits, chickens, geese, ducks, cattle, pigs, cats, and dogs. These conditions can spread rapidly, negatively affect animal productivity, and produce deadly results. For this reason, it is vital to determine the disease early and prevent its spread at an early age. Because of these parasites' diversity, complexity, and similarity, a system that automatically analyzes them using their microscopic images is needed. Because there is no specific model for the data set in the classification method, we developed a model to classify seven different Eimeria species using the deep learning-based CNN algorithm. We achieved the highest classification accuracy in the literature for automatically identifying and classifying Eimeria species using this model, which we developed using a CNN algorithm. In studies and models developed using previous CNN architectures, Abdalla et al. accuracy rate was 82%, Cesar et al. accuracy rate was 85.75%, Büyükyılmaz's accuracy rate was 87.44%, Diego et al. the accuracy rate was 90.42%. The model we developed has an accuracy rate of 93.85%. This model, which was developed to detect seven different types of parasites, has shown that it can be used successfully. It can be used in future studies to improve accuracy rates in identifying chicken Eimeria species by using a pre-trained network or extracting image features with a pre-trained network.

Acknowledgement

We would like to thank the Scientific Research Coordinatorship of Selcuk University for their support with the project titled "Data-Intensive and Computer Vision Research Laboratory Infrastructure Project", numbered 20301027.

Author contributions

Conceptualization: Zeki Kucukkara, Ilker Ali Ozkan; Methodology: Zeki Kucukkara, Ilker Ali Ozkan; Formal analysis and investigation: Zeki Kucukkara; Writing - original draft preparation: Zeki Kucukkara; Writing - review and editing: Ilker Ali Ozkan Özkan, Sakir Tasdemir; Supervision: Sakir Tasdemir

Conflicts of interest/Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Buyukyilmaz, A. Cibikdiken, M. Abdalla, and H. Seker, "Identification of chicken Eimeria species from microscopic images by using MLP Deep learning algorithm,", In Proceedings of the International Conference on Video and Image Processing (ICVIP 2017), 2017, Association for Computing Machinery, New York, NY, USA, 84–88. doi: 10.1145/3177404.3177445

[2] Mohamed A. E. Abdalla, Huseyin Seker, "Recognition of protozoan parasites from microscopic images: Eimeria species in chickens and rabbits as a case study", Engineering in Medicine and Biology Society (EMBC) 2017 39th Annual International Conference of the IEEE, pp. 1517-1520, 2017. doi: 10.1109/EMBC.2017.8037124

[3] C. A. B. Castã˜n´on, J. S. Fraga, S. Fernandez, A. Gruber, and L. da F. Costa, "Biological shape characterization for automatic image recognition and diagnosis of protozoan parasites of the genus Eimeria," Pattern Recognition, vol. 40, no. 7, pp. 1899–1910, 2007. doi: 10.1016/j.patcog.2006.12.006

[4] D. F. Monge and C. A. Beltrán, "Classification of Eimeria species from digital micrographies using CNNs," 10th International Conference on Pattern Recognition Systems (ICPRS-2019), 2019, pp. 88-91, doi: 10.1049/cp.2019.0254.

[5] Mohamed A. E. Abdalla, Huseyin Seker, Richard Jiang, "Identification of rabbit coccidia by using microscopic images", Engineering & MIS (ICEMIS) International Conference on, pp. 1-4, 2016. doi: 10.1109/ICEMIS.2016.7745328

[6] C. A. B. Castañón, J. S. Fraga, S. Fernandez, M. Pakandl, L. d. F. Costa, and A. Gruber. (2007). The Eimeria Image Database [Online]. Available: http://www.coccidia.icb.usp.br/imagedb/

[7] Yasaka, Koichiro & Akai, Hiroyuki & Kunimatsu, Akira & Kiryu, Shigeru & Abe, Osamu. (2018). Deep learning with convolutional neural network in radiology. Japanese Journal of Radiology. 36. doi: 10.1007/s11604-018-0726-3

[8] Gu, J. X., Wang, Z. H., Kuen, J., Ma, L. Y., Shahroudy, A., Shuai, B., . . . Chen, T. (2018). Recent advances in convolutional neural networks. Pattern Recognition, 77, 354-377. doi: 10.1016/j.patcog.2017.10.013

[9] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol 1, 655-665. doi: 10.3115/v1/p14-1062

[10] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874. doi: 10.1016/j.patrec.2005.10.010

[11] Koklu, M., & Ozkan, I. A. (2020). Multi-class classification of dry beans using computer vision and machine learning techniques. Comput. Electron. Agric., 174, 105507. doi: 10.1016/j.compag.2020.105507

[12] Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process, 5(2), 1. doi: 10.5121/ijdkp.2015.5201

[13] Taspinar, Y.S, Çinar, İ., and Koklu, M., "Prediction of Computer Type Using Benchmark Scores of Hardware Units" Selcuk University Journal of Engineering Sciences, vol. 20, no. 1, pp. 11–17, April. 2021.

[14] Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45, 427–437. 10.1016/j.ipm.2009.03.002.

[15] V. N. Vapnik and V. Vapnik, Statistical learning theory vol. 1: Wiley New York, 1998.

[16] W. Dai and W. Ji, "A mapreduce implementation of C4.5 decision tree algorithm," International journal of database theory and application, vol. 7, pp. 49-60, 2014.

[17] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research, 15 (1), 1929-1958. doi:10.5555/2627435.2670313

[18] İnik, Ö. and Ülker, E. (2017). "Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri", Gaziosmanpaşa Bilimsel Araştırma Dergisi, 6 (3), 85-104, Retrieved from https://dergipark.org.tr/tr/pub/gbad/issue/31228/330663