Journal of Intelligent Learning Systems and Applications, 2022, 14, 71-88
https://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

Application of Artificial Intelligence Algorithm in Image Processing for Cattle Disease Diagnosis

Bezawit Lake[1]*, Fekade Getahun[2], Fitsum T. Teshome[3]

[1]Department of Computer Science, Dilla University, Dilla, Ethiopia
[2]Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia
[3]Department of Agricultural and Biological Engineering, Tropical Research and Education Center, IFAS, University of Florida, Homestead, USA
Email: *mahiderlake@gmail.com

How to cite this paper: Lake, B., Getahun, F. and Teshome, F.T. (2022) Application of Artificial Intelligence Algorithm in Image Processing for Cattle Disease Diagnosis. *Journal of Intelligent Learning Systems and Applications*, **14**, 71-88.
https://doi.org/10.4236/jilsa.2022.144006

Received: May 31, 2022
Accepted: November 1, 2022
Published: November 4, 2022

Abstract

Livestock is a critical socioeconomic asset in developing countries such as Ethiopia, where the economy is significantly based on agriculture and animal husbandry. However, there is an enormous loss of livestock population, which undermines efforts to achieve food security and poverty reduction in the country. The primary reason for this challenge is the lack of a reliable and prompt diagnosis system that identifies livestock diseases in a timely manner. To address some of these issues, the integration of an expert system with deep learning image processing was proposed in this study. Due to the economic significance of cattle in Ethiopia, this study was only focused on cattle disease diagnosis. The cattle disease symptoms that were visible to the naked eye were collected by a cell phone camera. Symptoms that were identified by palpation were collected by text dialogue. The identification of the symptoms category was performed by the image analysis component using a convolutional neural network (CNN) algorithm. The algorithm classified the input symptoms with 95% accuracy. The final diagnosis conclusion was drawn by the reasoner component of the expert system by integrating image classification results, location, and text information obtained from the users. We developed a prototype system that incorporates the image classification algorithms and the reasoner component. The evaluation result of the developed system showed that the new diagnosis system could provide a rapid and effective diagnosis of cattle diseases.

Keywords

Deep Learning, Expert System, Livestock, Machine Learning, Neural Network

1. Introduction

Livestock is an important component of socio-economic development in Ethiopia. In terms of livestock population, the country ranks first in Africa and tenth in the world. Livestock contributes approximately 40% of the country's total agricultural output and 15% of the total gross domestic product [1]. However, various factors limit the potential economic benefits that could be obtained from livestock farming. One of the factors is the scarcity of veterinary practitioners that provide timely and accurate diagnoses and treatment. In Ethiopia, the veterinarian-to-animal ratio is 1:500,000 [2]. Another reason is the presence of endemic and transboundary diseases, which reduce the productivity of the cattle population [3]. Some of the factors that contribute to the spread of cattle diseases in the country include a lack of sufficient medical infrastructures, a lack of endemic and transboundary disease controlling protocol, a lack of local and central government attention, and the remoteness of livestock owners. A lack of priorly established endemic and transboundary disease controlling protocol could expose countries to a severe spread of animal diseases [4]. Disease outbreaks reduce the quality and productivity of animal products such as milk, skin, and hides, and this in turn results in trade restrictions on these products [5]. Cattle skin diseases are conditions that cause inflamed, irritated, or scaly skin, hair loss, changes in skin pigmentation, and visible growth [6]. Cattles suffer from a variety of skin problems, some of which are simple to treat while others are more difficult. Lumpy skin disease (LSD), bovine papillomatosis (warts), and dermatophytosis (ringworm) are common skin diseases in Ethiopia. Approximately 65% of skin and hide products are rejected due to poor quality caused by these diseases [7]. The prevalence of these diseases should be taken seriously to prevent them from negatively impacting the quality of finished products. As a result, quick detection and diagnosis of livestock diseases are critical for preventing the spread of outbreaks of livestock diseases. Currently, most medical diagnostic systems use either a text or image-based approach for acquiring symptoms. [3] developed a text-based expert system with hybrid reasoning consisting of the case and rule-based reasoning for the diagnosis of cattle diseases. The system accepts text-based symptoms from a user and then searches for a solution from a similar case in the case base. However, the accuracy of a symptom description of the conditions that were encountered using a text-based approach depends on the understanding of the person who describes the symptoms. Having a system that does not require the assistance of an expert to describe the symptoms will allow people without medical backgrounds to easily diagnose and treat the diseases. [8] developed a web-based expert system to diagnose infectious and non-infectious cattle diseases using rule-based reasoning. [9] developed an android application expert system to simplify disease detection and show brief information about cattle's using a rule-based forward reasoning engine. [10] developed a mobile-based diagnosis of skin diseases using case-based reasoning with image processing to detect diseases. They reported the system showed an encouraging performance with an accuracy of 83%. [11] developed a mobile application for the diagnosis of skin diseases using case-based reasoning with image processing to detect diseases by applying similar past problems. Thus, we need a system that can easily describe the symptoms of a disease in a way that is satisfactory for the correctness of the diagnosis. A hybrid diagnosis system has the advantage of representing symptoms more conveniently and easily compared to image or text-based systems. However, a hybrid diagnosis system is not sufficiently studied for cattle disease diagnosis. Therefore, the main objective of this project was to integrate artificial intelligence models such as convolutional neural networks and expert systems to develop a system that can diagnose cattle diseases easily and efficiently.

2. Materials and Methods

In this study, the integration of an expert system with deep learning image processing was proposed to develop a system that could quickly detect and diagnose cattle disease. The design of the cattle disease diagnosis system includes two modules *i.e.*, the expert system and image analysis modules.

Artificial intelligence (AI) techniques are known for their success in human and animal health studies [10]. An expert system (ES) is a computer program that can act as an expert in a specific field by analyzing and making decisions based on the knowledge base [12]. Digital image processing (DIP) is a technique that uses a computer system to manipulate images [13].

2.1. Hybrid Cattle Disease Diagnosis System

The design and development of a hybrid cattle disease diagnosis (HCDD) system consider the basic procedure used by veterinaries as well as three other considerations. The symptoms that were identified by visual inspection were brought to the system as an image. The images that were brought to the system were classified by the image analysis module to identify the symptoms category. In addition, the symptoms that were noted by palpation were brought to the system as text dialogue. The text interface was in form of a checkbox, where the user can select the appropriate symptoms. The location (GPS), where the image was captured, was extracted from the image to guide the diagnosis procedure. Smartphones tag all kinds of metadata to images. This metadata is known as exchangeable image file format (EXIF). Depending on the camera, EXIF data will store the current state of the camera when the photo was taken including date and time, shutter speeds, focal lengths, lens type, and location. The reasoning component reaches a final diagnosis result by integrating image classification results, location, and text information obtained from the users. The system architecture for the system is illustrated in **Figure 1**. The components of each module are described along with relevant techniques, algorithms, and considerations as the following.

The user interface (UI) module was responsible for communication between

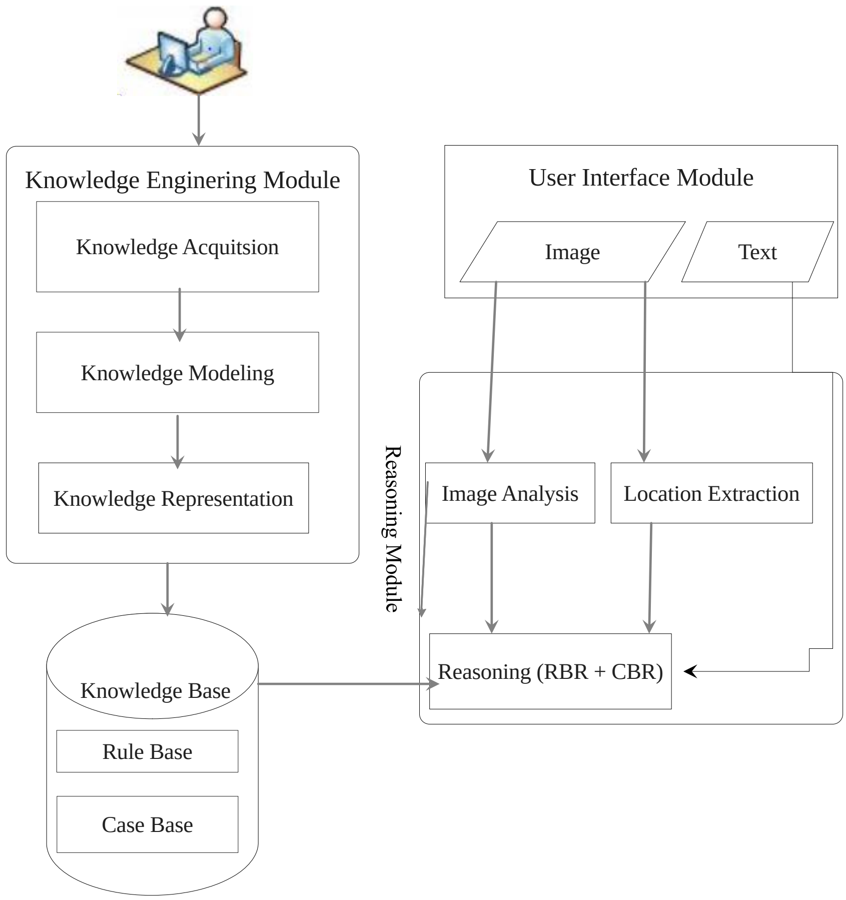

Figure 1. System architecture [RBR is rule-based reasoning and CBR is case-based reasoning].

the system and the user. It was used to collect important signs that occurred in symptomatic cattle. The UI has text and image-based information feeding components that allow users to feed critical information to the system.

The knowledge engineering module was responsible to build the required knowledge base for the proposed approach. It includes knowledge acquisition, knowledge modeling, and knowledge representation.

Knowledge acquisition was used to acquire knowledge and fact required for the diagnosis. The rules were extracted from the standard veterinary treatment guidelines such as Drug Administration and Control Authority (DACA) [14], black's veterinary dictionary [15] and Merck's veterinary manual [16]. The cases were collected from different case studies, and veterinary experts, and other literature were used as the main source of domain knowledge.

Knowledge modeling was responsible to model the acquired knowledge to understand the whole diagnosis and treatment techniques. It demonstrates the diagnosis procedures for skin diseases. The proposed disease diagnosis model considered all these features as shown in **Figure 2**.

Knowledge representation was a modeled knowledge that was represented in a way that was suitable for operation in a system. The case was represented using

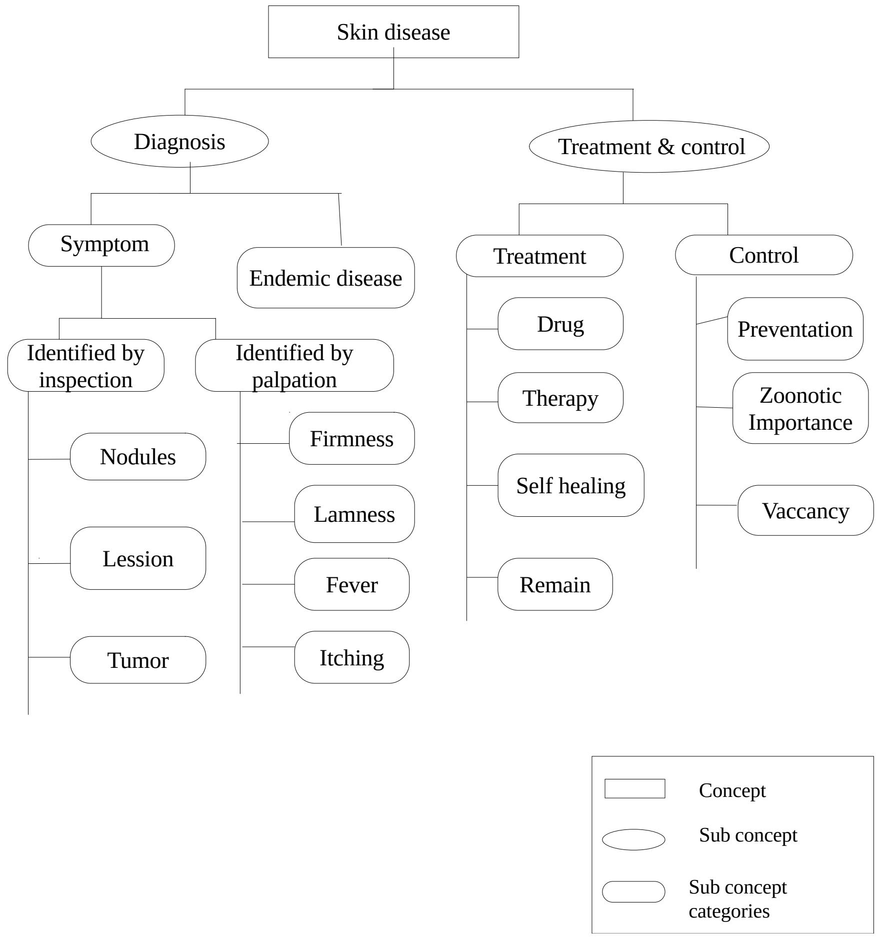

Figure 2. Disease diagnosis conceptual model.

cause-effect pairs and stored in the database. We used problem-solution case representation for location and their associated disease in that area. Data in the form of problem-solution pair was extracted for each case location and was stored in the database.

The design of the databases is shown in **Figure 3**. Table "CaseList" contained data about location information. The "LocationCaseDisease" table contained data about the diseases that occurred in each location. The "RuleMapping" table contained data about each rule and for what disease they were formulated. The rules acquired were represented using if … then representation, if a part contains the symptoms encountered and then part contains the consequence of the symptom. The treatment rules were represented using if … then representation if a part contains the disease and then part contains the treatment of the disease.

2.2. Image Analysis

The image analysis module was responsible for classifying the input image into different classes. In this study, a convolutional neural network classification algorithm was used to classify the images. A convolutional neural network (CNN) is a widely used image classification algorithm [17] [18] [19]. CNN automatically

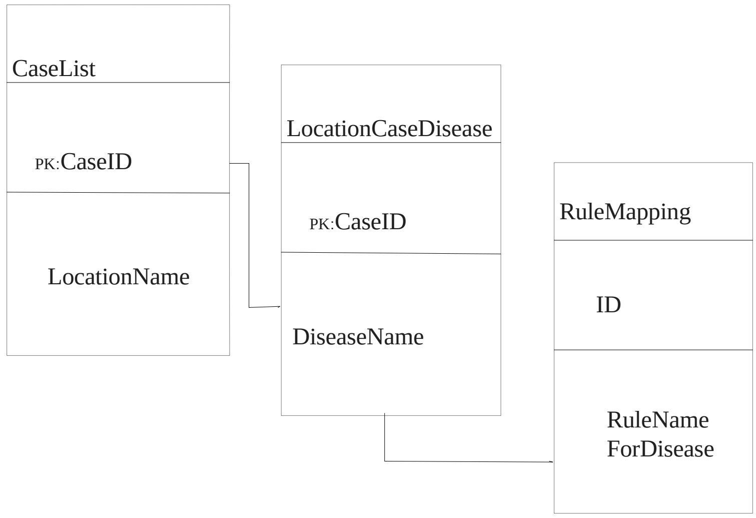

Figure 3. Database design [ID is identity document].

learns a hierarchy of features used for classification purposes. This is accomplished by successively convolving the input image with learned filters to build up a hierarchy of feature maps. Also, it is computationally efficient. The analysis starts with pre-processing the input images and stops when the classification result is found. **Figure 4** shows the architecture of the image analysis module.

2.2.1. Pre-Processing

The pre-processing component was responsible to make the images suitable for the overall image analysis activity. The main tasks were resizing and normalizing. Resizing is one of the pre-processing techniques which brings the whole image to the same size. Since the images were collected from different sources and had different sizes, we decided on a new image size that best reflects the contents of the image with less processing time. All images in our dataset were resized into 200 × 200 pixels. Normalizing was another pre-processing technique we used before further processing. We normalized our data values down to a decimal between 0 and 1 by dividing the pixel values by 255.

2.2.2. Classification Model

The architecture of the model composes layers that were responsible for feature extraction and classifying the input image into one of the categories. In our model 200 × 200 RGB (red, green, and blue) image was passed through a stack of convolutional (Conv) layers, where we used filters with 3 × 3 (which is the smallest size to skim pattern). The convolution stride was fixed to 1 pixel. The spatial resolution was preserved after convolution. Spatial pooling was carried out by three max-pooling layers, which follow the conv layers. Max pooling was performed over a 2 × 2-pixel window, with stride 2. **Figure 5** shows the proposed architecture of the CNN classification model.

In a convolutional network, the neurons are arranged in 3 dimensions width, height, and depth. The conv layers consist of a set of learnable filters. Every filter is arranged with width and height and extends through the full depth of the input

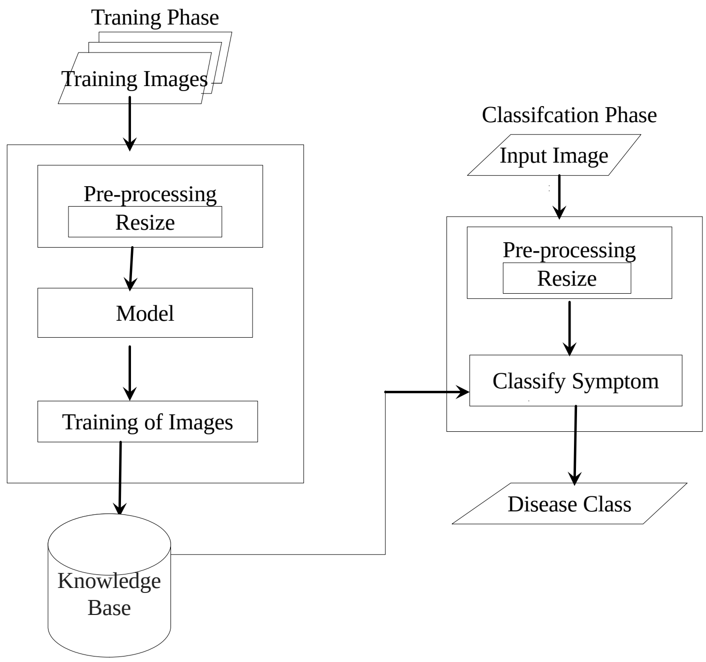

Figure 4. The architecture of the image analysis module.

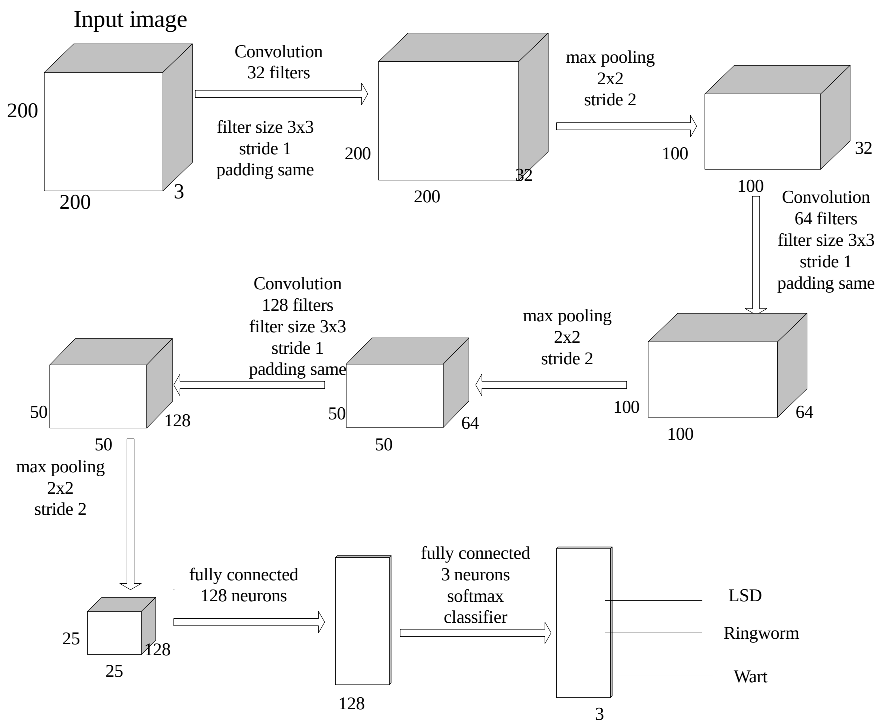

Figure 5. Classification model architecture [LSD is a Lumpy skin disease].

volume. The first conv layer accepted a 200 × 200 × 3 image and had a filter with 3 × 3 × 3 where the last 3 represented the depth of the filter which was like the input image. As we slide the filter over the width and height of the input volume, we produced a 2-dimensional activation map that gives the responses of that filter at every spatial position. Then we stacked these activation maps along the depth dimension and produced the output volume. In the first conv layer, 32 filters were applied which resulted in a 200 × 200 × 32 activation map. The max-pooling layer was performed a downsampling operation along the spatial dimensions (width, height), resulting in a volume of 100 × 100 × 32.

The second conv layer accepted the result of the first conv layer 100 × 100 × 32 and applied its 64 filters to the input which resulted in a 100 × 100 × 64 activation map. Max poling followed to perform downsampling and resulted in 50 × 50 × 64 volume. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and control overfitting. The third conv layer applied 128 filters. The final fully connected layer computed the class scores, resulting in a volume of size 1 × 1 × 3, where the 3 numbers correspond to a class score, among the 3 categories of our dataset. We observed that adding more layers did not improve the performance of our dataset. It leads to overfitting, increased memory consumption, and computation time. The removal of one layer from the model resulted in poor performance because the model had not generalized enough with a smaller number of layers in the model.

2.2.3. Training Classification Model

This component is where the model is trained for the task of image classification. The training starts by defining the optimizer, cost function, and metric. The workflow of the training is shown in **Figure 6**. The goal of a convolutional layer is feature extraction. When filter weights move over an image it checks for patterns in that section of the image. Filter weights change when the model is trained. In evaluation, these weights return high values if it thinks it is seeing a pattern it has seen before. The combinations of high weights from various filters let the network predict the content of an image.

The training was done using stochastic gradient descent with momentum (SGDM) optimizer, with a learning rate of 0.001, and a mini-batch size of 64 for 50 epochs. The training takes nine hours with HP Intel core i5-6200u CPU. The weight initialization was done using HE_UNIFORM initialization because it was compatible with the activation function, we use RELU for its less training time. The training (learning) in the model was presented as the function with the following form:

$$F(x) = f_5 f_4 f_3 f_2 f_1(x)$$

where, $f_1(x)$: Function learned on first conv layer, $f_2(x)$: Function learned on second conv layer, $f_3(x)$: Function learned on third conv layer, $f_4(x)$: Function learned on fourth hidden layer, and $f_5(x)$: Function learned on output layer.

The function value was computed by an iterative process of going and returning through layers of the model. The going was a forward propagation of the information, and the return was a backpropagation of the information. The first phase of forwarding propagation occurs when the network is exposed to the training data. Passing the input data through the network cause the neurons to

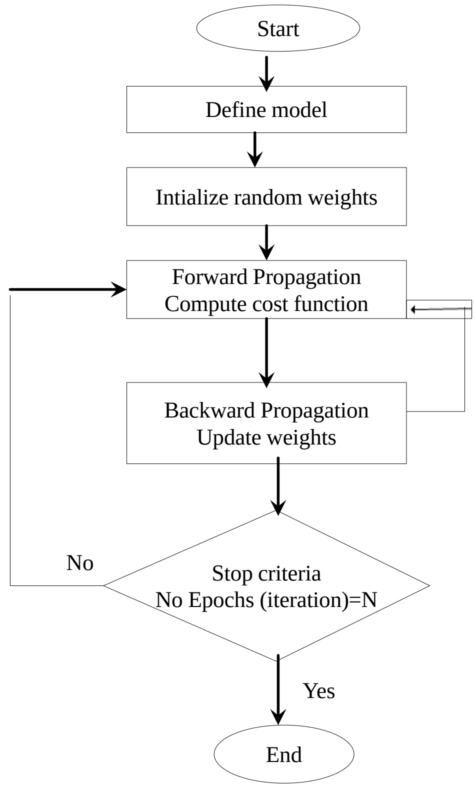

Figure 6. Workflow of the training algorithm [N is the number of iterations].

apply their transformation to the information they receive from the neurons of the previous layer and send it to the neurons of the next layer. When the data has crossed all the layers, the final layer was reaching a result of label prediction for those input data. After forward propagation loss function was calculated to estimate the error. The loss function was used to compare and measure how well or badly our prediction results compared to the correct result. Since our model task was the classification of the loss, we used Cross-Entropy Loss which can be defined by the following equation.

$$\text{CrossEntropyLoss} = -\left(T_y \log P_y\right) + \left(1 - T_y\right) \log\left(1 - P_y\right) \quad (1)$$

where T_y is the ground truth label for y and P_y is the predicted label for y.

The loss of information was calculated by propagating backward starting from the output layer to all neurons in the hidden layer. The neurons of the hidden layer only receive a fraction of the total loss, based on the contribution each neuron has contributed to the original output. This process was repeated, layer by layer, until all the neurons in the network have received a loss signal that describes their relative contribution to the total loss. This was done by calculating the partial derivate of the cost function relative to the weight of each neuron.

2.3. Reasoning Module

The reasoning module was responsible to seek information and relationship from the knowledge base to reach conclusion for the problem at hand. This component started with case-based reasoning (CBR) to acquire a disease that occurred in the input location case and rule-based reasoning (RBR) follows to reach a conclusion. The general high-level architecture of the reasoning module is shown in **Figure 7**.

The case which was like the input case was retrieved. Since the case base was a location case base, we needed an exact match when cases were retrieved. The algorithm used to retrieve cases was described in **Algorithm 1**.

Algorithm 1. Case retrieval from case base algorithm.

Input: input Case C_n, case stored in case base $C_m = [C_{m1}, C_{m2}, C_{mk}]$

Output: Similarity level

Begin:

For each case in the input case
Find the corresponding case in the stored case base
Compare the two values to each other
if $C_n = C_{mi}$
  $S_i = 1$
else $S_i = 0$
  END

Return Si

END

Adapt solution: the solution for the user query is decided if the retrieval stage identifies diseases, then the solution of the retrieved cases will be used. But if CBR retrieves no case, then the query is forwarded to the RBR component so that the final decision will be inferred from the rules. The workflow of the reasoning component was described in **Algorithm 2**.

Algorithm 2. Working algorithm of reasoning module.

Input: input Location case, classified image, text information

Output: Possible disease

Begin:

For input Location case
Location disease = result from CBR//list of diseases
If (Location disease = NULL)//Location case not found in KB
    Location = input Location case
    Possible disease = fire rules without mapping//no disease is stored on that location
    Store possible disease, Location in case base KB//as new case for the location
Else Fire rules with context//Location case found in KB
   Compute Mapping for Location disease with rule base
   Possible disease = fire rules with mapping
If (Possible disease = NULL)//no solution means it is new or epidemic disease for that Location
   Possible disease = fire rules without mapping
   Store possible disease, location in case base KB//as new case for the location

END

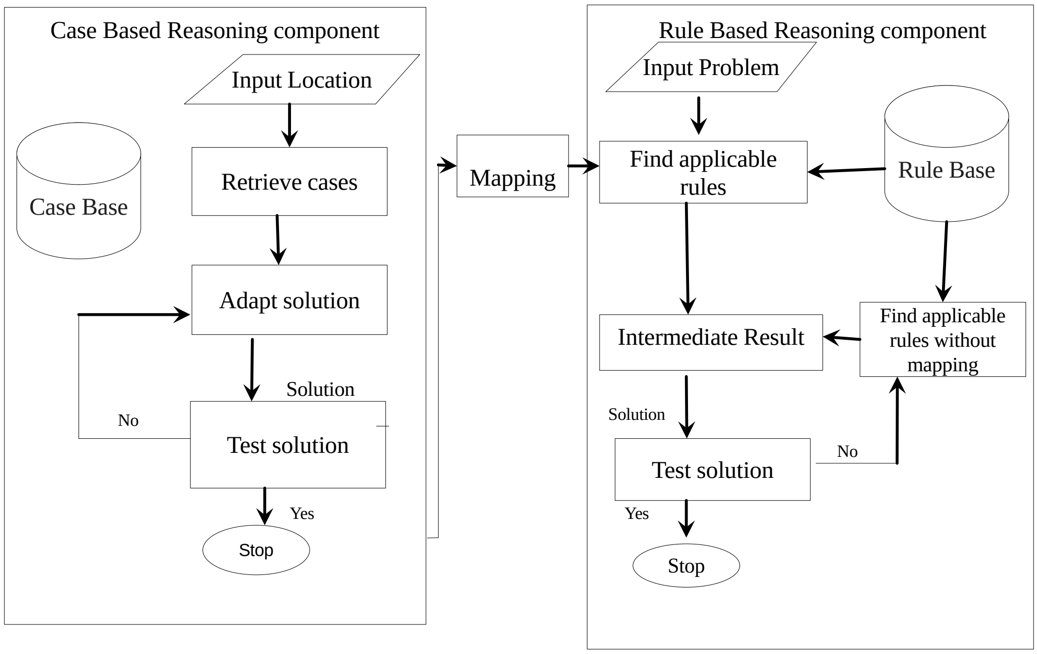

Figure 7. The architecture of the reasoner components.

2.4. Dataset Preparation

The process used for getting data ready for the classification model can be summarized in three steps: collect data, pre-process data and transform data. We followed this process iterative with many loops to prepare the dataset required.

Step 1. Collect data

This step involves collecting the available data needed to solve the problem. Image data were collected from Debre Markos University, Addis Ababa University-School of Veterinary Medicine, the Internet, and other secondary sources.

Step 2. Pre-process data

The pre-processing step was about getting the collected data into a form that can be easy to work. This step includes formatting and cleaning. The collected images were converted into JPEG format. JPEG format was selected because most of the collected images were in JPEG format and extraction of location information was possible with this format. We removed image data with unidentified labels so that we cannot get the proper label for them.

Step 3. Transform data

In the diagnosis system, we only needed the part of the cattle which was symptomatic. Therefore, we transformed the collected image by cropping the area where symptoms were presented. The original images are cropped into images containing symptoms regions. The cropping follows the following rules, including healthy and symptom parts, isolated symptoms taken individually, and widespread symptoms taken both as a whole and divided into regions. After transformation based on the specified rules, we transformed the input images into several images. The data we collected before and after transformation is shown in Table 1.

Table 1. Collected data before and after transformation.

No	Disease	Number of images collected	
		Before transformation	After transformation
1	Lumpy skin disease (LSD)	84	146
2	Ringworm	57	100
3	Wart	64	124

Our model required a large amount of labeled data. But getting enough data was a major problem in our cases which leads to the use of other techniques to expand our dataset. Data augmentation provides a means for increasing the quantity of training data available for machine learning and is particularly relevant when training deep learning systems from scratch [20]. Image augmentation is the process of taking images that are in a training dataset and manipulating them to create many altered versions of the image. It provides more images to train and expose our classifier to a wider variety of transformed images to make the classifier more robust. It has been widely used on small datasets for combatting over-fitting [19] [21]. Techniques of augmentation used in our dataset include horizontal and vertical flipping, zoom, shear, and rotation. **Figure 8** shows lumpy skin disease (LSD) infected cattle images after applying augmentation. After applying augmentation our dataset expands to 3990. Then we split into 90% training and 10% testing.

2.5. System Evaluation

The accuracy of the system is affected by the performance of the classification model, so evaluation was done on the model first. We evaluated the performance of the classification model using deep learning evaluation techniques accuracy and confusion matrix. Then, the accuracy of the entire system was evaluated with user evaluation. The performance of the model could be poor either due to overfitting or underfitting the data. The training of the model was plotted to see the possibility of overfitting and underfitting in the model.

The entire proposed system was evaluated by four people we selected randomly from different professions. Two of them were veterinarians, and the rest were individuals who have cattle. The selection assumed that those with a veterinary background can see and evaluate technical details while others may evaluate the applicability, accuracy, and importance of the system. Before starting the evaluation process, the system was explained in detail to the evaluators. The questionnaire we use for system evaluation is shown in **Table 2** where the user puts the weight for each evaluation. Weight values show the value of the evaluation, where 3 indicates the highest, 2 indicates the medium, and 1 indicates the lowest.

In addition, we constructed an evaluation matrix to evaluate the system. The evaluation matrix was a table with one row for each evaluator and columns that address evaluation questions shown in **Table 2**.

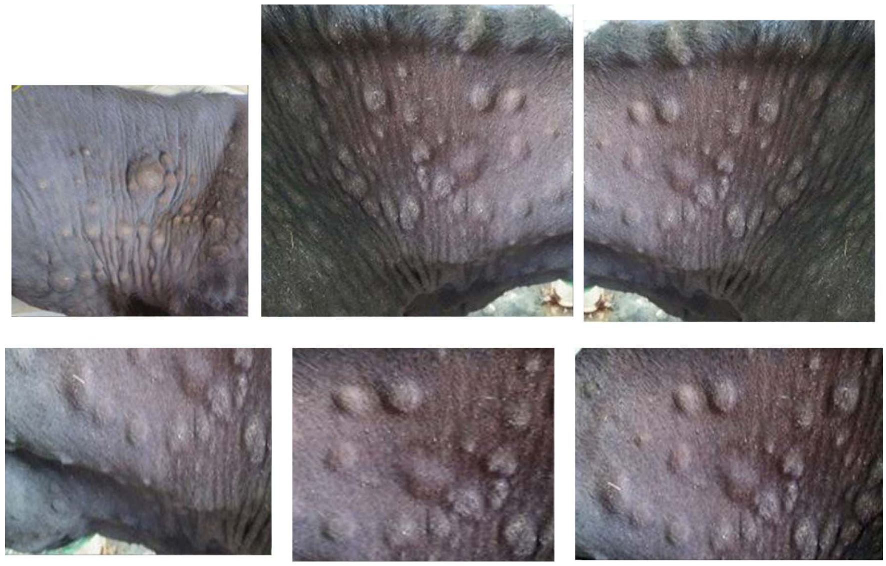

Figure 8. Images after augmentation techniques.

Table 2. System evaluation form.

No	Questions	1	2	3
1	Is the prototype system a user-friendly interface?			
2	How is the response time of the prototype system?			
3	Is the description of the symptom valid?			
4	Is the diagnosis result, correct?			
5	Is the treatment and recommendation, correct?			
6	How is the applicability of the prototype system?			
7	How is the performance of the prototype system?			

3. Results and Discussion

3.1. System Prototype

To demonstrate the validity of the proposed system, we developed a porotype system. The prototype shows the user interface and output of the system (**Figure 9**). The user interface allows the user to upload a photo and fill in the text information to the system. After the inputs are fed into the system, location extraction was done. Then, the image was passed to the processing components and the diagnosis result was given to the user.

3.2. System Evaluation

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. When the training accuracy is above the test accuracy it means the model is overfitting. Our model was not overfitting as shown in **Figure 10**. There was no significant difference between the value of training and test accuracy.

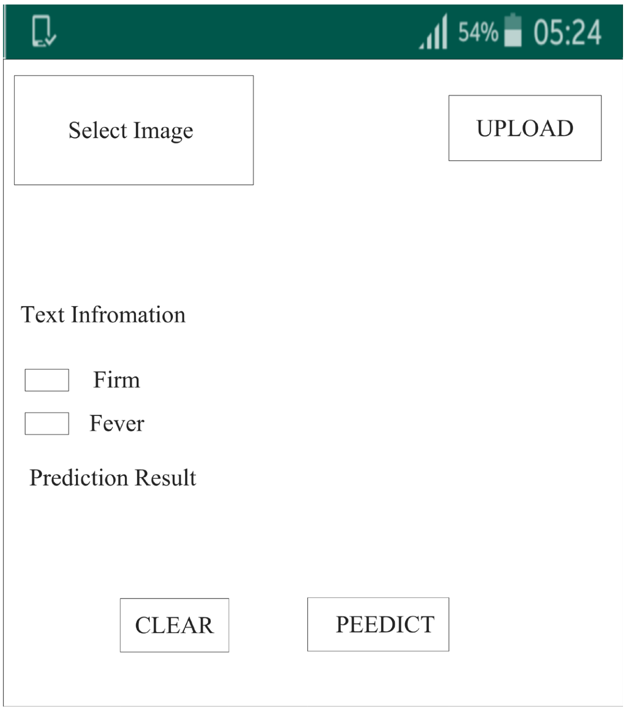

Figure 9. Prototype user interface.

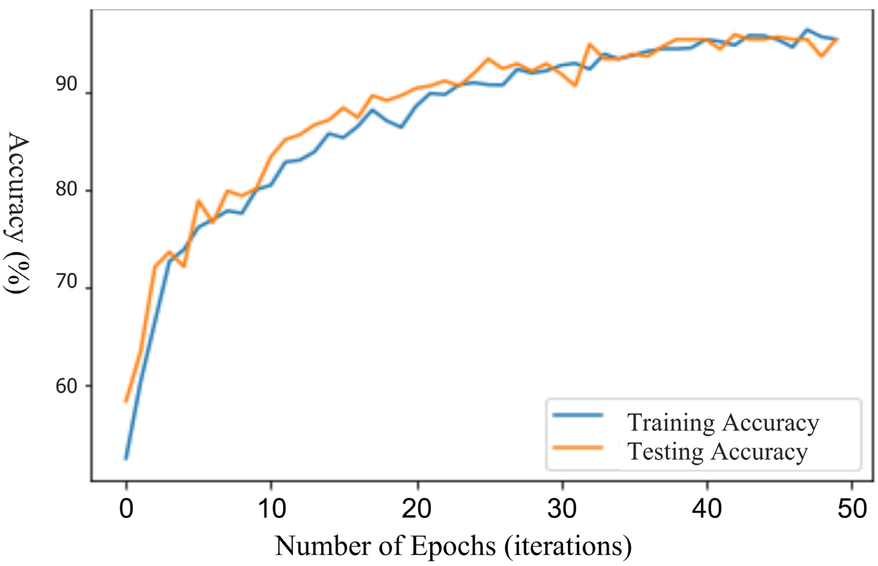

Figure 10. The plot of training and testing accuracy [X-axis is accuracy and Y-axis is number of epochs].

Underfitting refers to a model that can neither model the training data nor generalizes to new data. When validation loss is below the training loss the model is underfitting. As shown in Figure 11, our model was not underfitting. The model achieves 95% accuracy in 50 epochs.

To summarize the performance of the model we used a confusion matrix. In the confusion matrix, the number of correct and incorrect predictions was summarized with count values and broken down by each class. Figure 12 shows

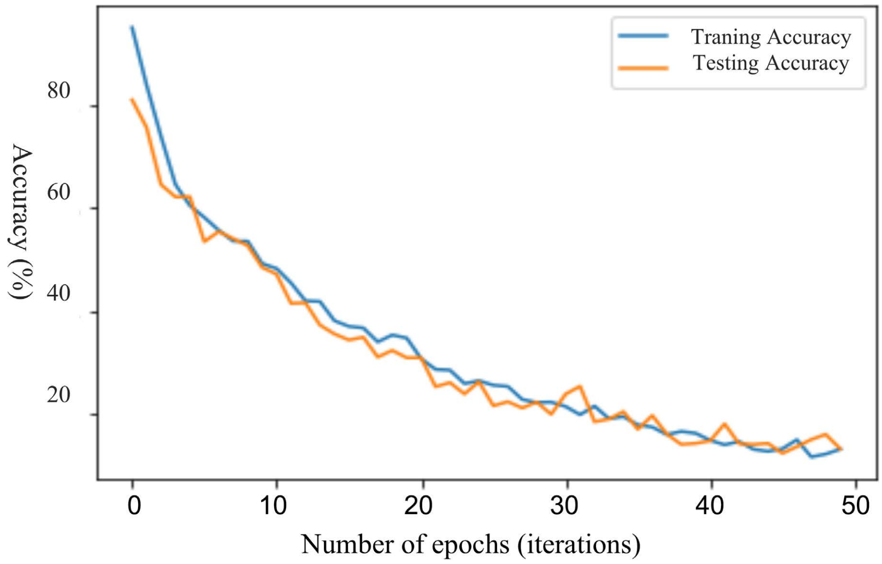

Figure 11. A plot of training and testing loss (X-axis is loss and Y-axis is number of epochs).

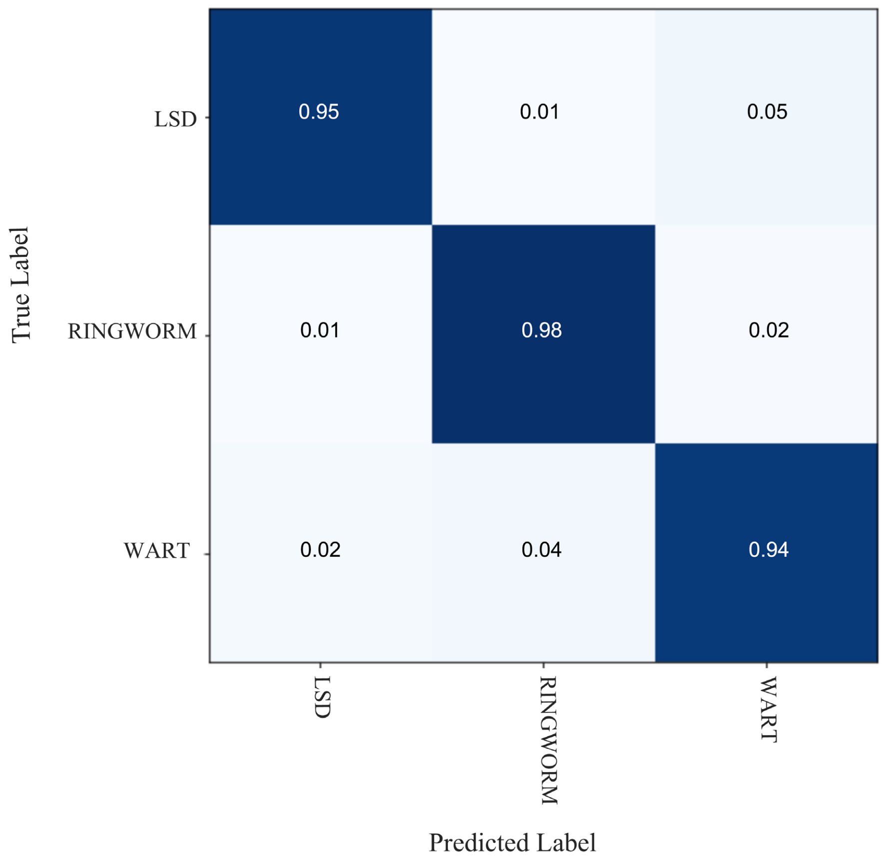

Figure 12. Confusion matrix of the model (LSD is a Lumpy skin disease).

how many of the images were misclassified and classified correctly. The ringworm classification was high compared to others because it has an easily distinctive feature from the other two classes. The misclassification of LSD as Wart and

Table 3. System evaluator's response.

No	Q1	Q2	Q3	Q4	Q5	Q6	Q7
1	3	3	3	3	3	2	3
2	3	3	3	3	3	2	2
3	3	2	3	3	2	3	3
4	3	3	3	3	2	3	3

Wart as LSD was noticed. Because LSD and Wart have more similar symptoms it is sometimes difficult to distinguish distinctive features among them.

The evaluator's response is recorded in **Table 3**. In Q1, the evaluation result shows that the prototype was user-friendly. It is easy for anyone to use because the interaction does not need any background knowledge. The evaluation result of Q2 shows that the response time for user queries was good. The response time is constrained by the storage and processing capability of the user's phone. For all users, the response time will be different. In Q3, the evaluation result shows a high value for the validity of symptom descriptions. So, representing symptoms using the image was a valid diagnosis method for cattle disease. In Q4, the diagnosis result shows a high value, which validates our diagnosis approach. In Q5, the correctness of treatment shows a lower result. The veterinarian recommends considering body mass index when a drug is prescribed, which our system did not include. In Q6 and Q7, the diagnosis result shows the performance and applicability of the proposed system were high. We can conclude that the integration of expert systems and image processing using deep learning gives an efficient and timely diagnosis of cattle diseases.

4. Conclusion

The potential economic benefit of livestock farming is impacted by numerous factors. One of these factors is the prevalence of livestock disease. This condition has been a significant factor that has affected the economic benefit of livestock farming. Quick detection and diagnosis of livestock disease are critical to prevent any outbreak of livestock disease from further spreading, as well as to improve the economic benefits that could be obtained from livestock production. In this study, the cattle disease diagnosis approach was developed by integrating an expert system and image processing using a deep learning algorithm. Diagnosis starts by acquiring information on the occurred disease through imageries and texts. Symptoms identified by inspection were acquired by capturing the image through mobile phones. Symptoms identified by palpation were presented to the system using text. To know the epidemic capability, location information was presented to the diagnosis system. Then, the image was pre-processed, and the class of the images was identified by the trained CNN model. The final diagnosis conclusion is drawn by the reasoner component of the expert system. The evaluation result shows that the developed approach effectively diagnoses cattle disease.

Acknowledgements

We would like to express our greatest gratitude to Dilla University for providing the funding for this research project.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Birhanu, T. (2014) Prevalence of the Major Infectious Animal Diseases Affecting Livestock Trade Industry in Ethiopia. *Journal of Biology, Agriculture and Healthcare*, **4**, 76-82.

[2] Tefera, M. (2010) Global Crisis and the Challenge of Veterinary Teaching in Ethiopia. *Global Veterinaria*, **5**, 294-301.

[3] Gebre-Amanuel, E.K., Taddesse, F.G. and Assalif, A.T. (2018) Web-Based Expert System for Diagnosis of Cattle Disease. In: *Proceedings of the* 10*th International Conference on Management of Digital EcoSystems*, *MEDES'*18, Association for Computing Machinery, New York, 66-73. https://doi.org/10.1145/3281375.3281400

[4] Domenech, J., Lubroth, J., Eddi, C., Martin, V. and Roger, F. (2006) Regional and International Approaches on Prevention and Control of Animal Transboundary and Emerging Diseases. *Annals of the New York Academy of Sciences*, **1081**, 90-107. https://doi.org/10.1196/annals.1373.010

[5] Donald, B., John, D. and David, H. (2006) USDA ERS—Economic Effects of Animal Diseases Linked to Trade Dependency [WWW Document]. https://www.ers.usda.gov/amber-waves/2006/april/economic-effects-of-animal-diseases-linked-to-trade-dependency

[6] Amin, D.M., Shehab, G., Emran, R., Hassanien, R.T., Alagmy, G.N., Hagag, N.M., Abd-El-Moniem, M.I.I., Habashi, A.R., Ibraheem, E.M. and Shahein, M.A. (2021) Diagnosis of Naturally Occurring Lumpy Skin Disease Virus Infection in Cattle Using Virological, Molecular, and Immunohistopathological Assays. *Veterinary World*, **14**, 2230-2237. https://doi.org/10.14202/vetworld.2021.2230-2237

[7] Belete, S. and Derso, S. (2015) Prevalence of Major Gastrointestinal Parasites of Horses in and around Mekelle (Quiha and Wukro). *World Journal of Animal Science Research*, **3**, 1-10.

[8] Lake, D. (2013) Web-Based Expert System for Cattle Diseases Diagnose. Masters, Addis Ababa University, Addis Ababa.

[9] Zamsuri, A., Syafitri, W. and Sadar, M. (2017) Web Based Cattle Disease Expert System Diagnosis with Forward Chaining Method. *IOP Conference Series*: *Earth and Environmental Science*, **97**, Article ID: 012046. https://doi.org/10.1088/1755-1315/97/1/012046

[10] Sallam, A. and Ba Alawi, A.E. (2019) Mobile-Based Intelligent Skin Diseases Diagnosis System. *The* 2019 *First International Conference of Intelligent Computing and Engineering* (*ICOICE*), Hadhramout, 15-16 December 2019, 1-6. https://doi.org/10.1109/ICOICE48418.2019.9035129

[11] Mieras, L.F., Taal, A.T., Post, E.B., Ndeve, A.G.Z. and van Hees, C.L.M. (2018) The Development of a Mobile Application to Support Peripheral Health Workers to Diagnose and Treat People with Skin Diseases in Resource-Poor Settings. *Tropical Medicine and Infectious Disease*, **3**, Article No. 102.

https://doi.org/10.3390/tropicalmed3030102

[12] Tripathi, K.P. (2011) A Review on Knowledge-based Expert System: Concept and Architecture. *IJCA Special Issue on Artificial Intelligence Techniques—Novel Approaches & Practical Applications*, No. 4, 21-25.

[13] Ravikumar, R. and Arulmozhi, D.V. (2019) Digital Image Processing—A Quick Review. *International Journal of Intelligent Computing and Technology* (*IJICT*), **2**, 11-19.

[14] DACA (2006) Standard Veterinary Treatment Guidelines (Drug Administration and Control Authority of Ethiopia). Veterinary Medicine. Infection. https://www.scribd.com/doc/48645004/STANDARD-VETERINARY-TREATMENT-GUIDELINES-Drug-Administration-and-Control-Authority-of-Ethiopia

[15] Boden, E. (2005) Black Veterinary Dictionary.

[16] Branion, H.D. (1963) The Merck Veterinary Manual. *Poultry Science*, **42**, 552-553. https://doi.org/10.3382/ps.0420552

[17] Alimboyong, C.R., Hernandez, A.A. and Medina, R.P. (2018) Classification of Plant Seedling Images Using Deep Learning. *The TENCON* 2018-2018 *IEEE Region* 10 *Conference*, Jeju, 28-31 October 2018, 1839-1844. https://doi.org/10.1109/TENCON.2018.8650178

[18] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, **60**, 84-90. https://doi.org/10.1145/3065386

[19] Sajjad, M., Khan, S., Muhammad, K., Wu, W., Ullah, A. and Baik, S.W. (2019) Multi-Grade Brain Tumor Classification Using Deep CNN with Extensive Data Augmentation. *Journal of Computational Science*, **30**, 174-182. https://doi.org/10.1016/j.jocs.2018.12.003

[20] Spyridonos, P., Papageorgiou, E.I., Groumpos, P.P. and Nikiforidis, G.N. (2006) Integration of Expert Knowledge and Image Analysis Techniques for Medical Diagnosis. In: Campilho, A. and Kamel, M., Eds., *Image Analysis and Recognition*, Springer, Berlin, 110-121. https://doi.org/10.1007/11867661_11

[21] Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J. and Johannes, A. (2019) Deep Convolutional Neural Networks for Mobile Capture Device-Based Crop Disease Classification in the Wild. *Computers and Electronics in Agriculture*, **161**, 280-290. https://doi.org/10.1016/j.compag.2018.04.002