Pest Detection in Agricultural Farms using SqueezeNet and Multi-Layer Perceptron Model

Intan Nurma Yulita[1*], Anton Satria Prabuwono[2], Firman Ardiansyah[3], Juli Rejito[4], Asep Sholahuddin[5], Rudi Rosadi[6]
Department of Computer Science, Universitas Padjadjaran, Sumedang, Indonesia[1, 4, 5, 6]
Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Rabigh, Saudi Arabia[2]
Magister of Management, Institut Teknologi dan Bisnis Ahmad Dahlan Lamongan, Indonesia[3]

***Abstract*—Pest detection is essential to protect agricultural systems from economic losses, lower food production, and environmental degradation. Detection of pests is a crucial aspect of agricultural sustainability because it helps to allocate resources, reduce production costs, and increase producers' profits. Artificial intelligence (AI) has revolutionized the detection of agronomic pests by employing deep learning models to accurately detect individual pests and differentiate between species and life stages. Combining SeueezeNet and Multi-Layer Perceptron, this study extracts feature vectors from image data to detect pests. There are four primary phases: preprocessing, image embedding with SqueezeNet, the final classifier with MLP, and 10-fold cross-validation. Data for this study is acquired in the form of plant pests. The total number of images acquired is 3150, with 350 from each class. Based on the research, the combination model demonstrates excellent performance. Each experiment's accuracy is greater than 99 %. It shows that Squeezenet can effectively extract the data's features, whereas Multi-Layer Perceptron can process these features for optimal classification performance. Even though there are still several classes, such as mites, sawflies, and stem borer, that have not been correctly classified. Since each image's background is unique, it cannot be classified correctly. These promising findings have broad implications for boosting agricultural output and decreasing pest-related losses. Optimal use of this approach in a variety of agricultural contexts requires more study and field testing.**

Keywords*—Pest detection; Squeezenet; multi-layer perceptron; deep learning

I. Introduction

Globally, agricultural systems are threatened by pests, which cause enormous economic losses, lower food production, and environmental degradation. To create efficient tactics that allow for early detection and focused control of pests, it is essential to research pest detection in agriculture [1-2]. Infestations of pests on crops and livestock can lead to significant financial damages. Crop pests, including insects, fungi, bacteria, and viruses, have the potential to cause harm to crops and agricultural goods, resulting in a decrease in anticipated yield. Consequently, farmers have less earnings, potentially leading to an increase in the pricing of agricultural goods, which can harm consumers. Globally, insect invasions can result in economic losses of up to billions of dollars annually. Effective pest identification is crucial for ensuring and upholding global food security. Annually, significant quantities of food are squandered or impaired as a result of unregulated insect infestations. This can result in food insecurity and famine for people reliant on agricultural yields for sustenance.

Agriculturalists encounter significant obstacles in managing pests. Frequently, they must confront assaults from a diverse range of pests, including those that have just appeared or have acquired immunity to the pesticides employed. These circumstances can lead to significant expenses, strenuous labor, and elevated levels of stress for agricultural farmers. Accurate identification of pests is crucial for successful pest management. Failure to promptly discover or diagnose a pest infestation may render preventative or treatment efforts ineffective, perhaps leading to exacerbated harm. Advanced detection technologies, such as sensors that rely on image analysis, data processing, and artificial intelligence, can assist in swiftly and precisely identifying objects or phenomena. The overutilization of pesticides for pest management can result in adverse effects on both the environment and human well-being. Improved pest identification enables farmers to employ pesticides with more precision and effectiveness while minimizing environmental repercussions and health hazards. Pest infestations can impede a nation's capacity to export agricultural commodities. Upon the detection of pest infestations on agricultural products, export destination nations have the authority to enforce trade restrictions, thereby causing harm to agricultural exports and the whole national economy. Within a worldwide framework, the identification of pests plays a crucial role in guaranteeing the safety of food, the advancement of the economy, and the preservation of the environment. Hence, the advancement of superior detection technologies and methodologies is crucial in endeavors to safeguard agricultural productivity and uphold worldwide food security.

It shows the need to investigate pest detection in agricultural settings. Insects, weeds, and diseases are all examples of pests that may significantly impair agricultural yields and quality [3-4]. Discoveries from the field of pest detection have helped farmers much in spotting pests at their earliest stages of infestation, at which point they may begin taking effective preventative measures. Greater agricultural output and food security can result from early detection strategies that limit losses, maintain crop health, and optimize yields.

To reduce the amount of harmful chemicals released into the environment while still effectively eradicating pests, Integrated Pest Management (IPM) was developed. IPM relies on accurate pest identification so that farmers can keep tabs on pest populations, set appropriate intervention levels, and take precise preventative measures. Studies on pest identification help farmers create IPM plans that work for their unique fields, climates, and pest populations. The overuse of chemical pesticides not only endangers human and environmental health but also encourages pests to develop resistance [5]. With the help of precision agricultural techniques, which rely on precise pest identification and population monitoring, the acceptance of research on pest detection has facilitated the widespread use of pesticides. By lowering chemical inputs thanks to better pest identification, farmers can protect beneficial creatures, maintain ecological balance, and protect the environment.

Crop yields and quality can be severely impacted by plant diseases [6-8], leading to significant economic losses. Effective disease control in agriculture relies on early identification and prevention. If farmers can discover diseases early on, they may take preventative actions like changing their irrigation methods, using resistant crop types, or using tailored treatments to lessen the impact the illness has on their crops. Pest infestations may have a devastating effect on a farm's bottom line and long-term viability. Overusing pesticides, losing crops, and having to hire extra help all add up, so it's important to be able to spot them quickly and accurately. Farmers can benefit from better pest detection and management decisions because of investments in research on pest detection that provide access to improved tools, technology, and information. Long-term agricultural sustainability depends on accurate pest identification, which improves resource allocation, lowers production costs, and boosts farmers' bottom lines.

The development of AI has completely transformed the detection of agronomic pests [9], [10], resulting in a paradigm change in the pest detection industry. AI-enabled systems have enabled improvements in precision, efficacy, and proactiveness. Image recognition and pattern recognition are two domains in which AI systems, particularly those based on deep learning, have demonstrated excellence. This innovation enables the automatic identification and categorization of parasites based on form, color, and texture. These systems can accurately detect individual pests, even distinguishing between species and life stages. This saves producers time and effort by eliminating the need for them to manually inspect crops for parasites. Intricate patterns and characteristics in images may be learned by deep learning models, allowing them to precisely detect pests in agricultural situations [4], [11], [12]. These models can extract image embeddings to characterize the visual features of pests compactly and understandably, allowing for more precise recognition. Li, Y., and Yang, J. present a few-shot cotton pest recognition method that requires only a small amount of raw training data, in contrast to traditional deep learning algorithms [13]. To prove the few-shot model works, they use data collected in real-world scenarios. A convolutional neural network (CNN) is used to extract feature vectors from images. The CNN feature extractor is trained using the triplet loss to ensure the system is flexible enough to deal with different types of pests.

In their study, Peng, Y., and Wang, Y. [14] offer a method for insect pest recognition that combines transformer architecture with convolution blocks. The representative features of an input image are extracted using a backbone convolutional neural network. The input images are processed through CNN structures made up of several CNN blocks to extract embeddings (visual features). Once the embeddings have been extracted from the backbone network, a simple global average pooling (GAP) layer is used to convert them into a one-dimensional vector. The next step is to feed this vector into a linear classifier, which typically consists of one or more fully connected layers, to generate prediction vectors. Both of these researchers embedded images using a convolutional neural network (CNN) that had not been pre-trained. In contrast, David et al. create embeddings from leaf images using a CNN image classification network [15]. The Inception V3 network was trained in the source domain to learn generic plant leaf properties. This data was sent to the desired domain to learn new types of leaves from a limited set of images. However, there are several drawbacks to Inception V3 as compared to SqueezeNet.

SqueezeNet is a relatively more straightforward architecture than Inception V3 [16]. The sequential, layered design makes it simple to learn and put into practice. The efficiency of SequeezeNet's computing resources is improved by its simplicity, which allows for faster training and inference times. SqueezeNet often requires less RAM than Inception v3. This is helpful when working with constrained resources, such as those found on mobile platforms or peripheral devices. With less RAM needed, SqueezeNet is easier to roll out and makes better use of available hardware. Additionally, SqueezeNet is ideal for transfer learning assignments. To fine-tune a model that was originally trained on a big dataset, transfer learning is used. Squeezenet's simplicity facilitates adaptation and fine-tuning for particular tasks and datasets [17]. It enables the efficient transmission of knowledge from large-scale image datasets to smaller insect recognition a situation with limited labeled training data, SqueezeNet can perform admirably. Due to its simplified architecture, it can obtain excellent performance with reduced training datasets [18]. Advantageous in agriculture, where obtaining large datasets of labeled pests can be difficult. The capacity of SqueezeNet to generalize effectively with limited data can help mitigate the problem of data scarcity [19]. This work aims to highlight the benefits of utilizing SqueezeNet for image embedding, with the final layer being implemented as a Multilayer Perceptron (MLP). This deviates from the common approach of applying the last layer straightforwardly. The provided technique aims to optimize system performance in modeling classification.

The objective of this research is to create a pest detection system by utilizing SqueezeNet to extract features from images of agricultural lands. To improve the classification accuracy of discovered pests, it is necessary to combine a Multi-Layer Perceptron model with SqueezeNet. The subsequent sections of this work are structured in the following manner. Section II: Material and Method - This section provides a detailed explanation of the proposed model, encompassing the structure of SqueezeNet and the MLP, along with the procedures involved in data preparation and the training procedure. Section III: Results and Discussion - This chapter provides the performance outcomes of the model on the pest image dataset. Section IV: Conclusions - This chapter provides a concise overview of the main discoveries, contributions, and potential future paths of the research.

II. Material and Method

The research process is depicted in Fig. 1. There are four main phases, including preprocessing, image embedding with SqueezeNet, the final classifier with MLP, and 10-fold cross-validation for image embedding, while the final classification layer is a multi-layer perceptron.

A. Data Collecting

As seen in Fig. 1, data for this study is acquired in the form of plant pests. To find data, the Kaggle data collection website (https://www.kaggle.com/simranvolunesia/pest-dataset) was used. Aphids, armyworm beetles, bollworms, grasshoppers, mites, mosquitoes, sawflies, and stem borers are among the plant parasites included in the dataset. It is vital to ensure that the amount of data obtained in each class is comparable so that the model may be weighted more easily throughout the learning phase. The total number of images acquired is 3150, with 350 from each class. Fig. 2 depicts the capturing of several pests.

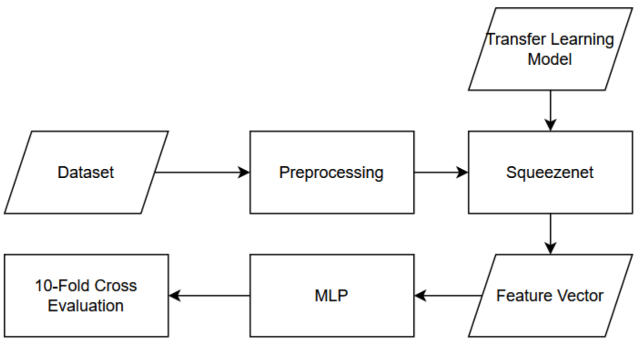

Fig. 1. Research methodology.

B. Image Processing

The process of preparing images was an extremely important step in guaranteeing the quality, uniformity, and compatibility of the data. Before continuing to this step, the system first performed any necessary data preparation, such as scaling the images to a consistent size, normalizing their pixel values, and converting them to grayscale.

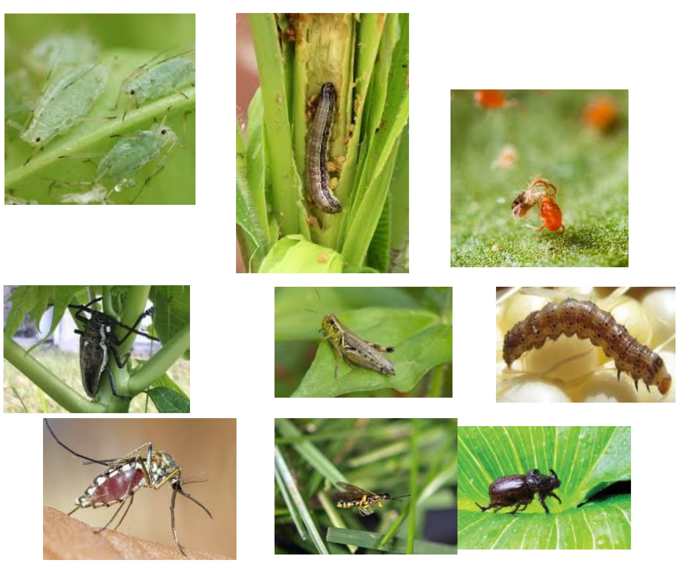

Fig. 2. Several pests in this study.

C. SqueezeNet

SqueezeNet is a lightweight deep-learning network that excels in image classification applications due to its small size and fast inference [19]. Although its primary purpose is categorization, the model can also be used to embed images. When an image is embedded, it is converted from its original format into a fixed-dimensional vector representation called an embedding vector. This vector summarizes the key aspects of the image, allowing for rapid comparison and study. It uses the intermediate layers to determine crucial image attributes. Numerous modules, including compress and expand layers, make up the network [16]. The number of input channels is constrained by the 1x1 filters used in the compress levels, while the local and global details are captured by the 1x1 and 3x3 filters in the expand layers.

Using SqueezeNet, this study eliminated classification layers at the end of the network and used activations from previous levels to create image embedding. These activations were capable of serving as image embeddings. As image embeddings, the activations from the last fully linked layer or the layer before it was frequently utilized. These activations produced a fixed-length vector representation of the discriminative properties of the input image. The generated image embeddings could then be used for image classification. By comparing the distance or similarity between embedding vectors, it performed tasks such as discovering similar images or detecting anomalies within a group of images.

D. Multi-Layer Perceptron

Artificial neural networks, such as the multi-layer perceptron (MLP) [20-21], are made up of layers upon layers of connected elements (neurons). Following image embedding, it is often employed as the final classifier in a variety of machine-learning applications, including image classification. In this research, SqueezeNet was used as the image embeddings, and the MLP functions were used as the final layer of the classification process. The final classifier's MLP architecture includes the following parts [22]:

- Input Layer: The MLP receives as input the embedded image vector computed by the image embedding model. The image's various features and dimensions are represented by individual elements of the vector input. The input layer of the network is the initial stop for the data. The input layer's neurons do not perform any transformations or calculations on the data. Their only job is to pass on the input data to the hidden-layer neurons that come later. All of the hidden layer's neurons are weighted and connected to the input layer [23-24]. It is common practice to normalize or scale input data before sending it to the input layer. The input values are processed so that they fall within a suitable range for the activation functions before the MLP is trained.

- Hidden Layer: One or more hidden layers, which sit between the input and output layers, are a standard component of the MLP architecture. The strength of the connections between neurons in a given layer is determined by weights. The neurons in the hidden layer act as the processors in between the input and output layers in a multi-layer neural network. To produce an output, hidden-layer neurons take information from the layer above them, be it the input layer or another hidden-layer neuron, in conjunction with their weights and biases. Because the neurons in the hidden layer perform a nonlinear change on the input data, the MLP can learn complex patterns and correlations [20]. Each neuron's activation function (such ReLU or tanh) is applied to the weighted sum of inputs and biases to introduce non-linearity. The number of hidden layers and the number of neurons in each hidden layer are two examples of design considerations in developing an MLP that is commonly established through testing and model refinement. While adding more hidden layers and neurons to a model may seem like a good way to train it to learn more sophisticated representations, this approach can backfire if the model isn't properly regularized. Each neuron in a hidden layer is connected to every other neuron in the layer below it via a weighted link. It could be a layer below the input layer or even deeper. These weights are used to assign relative relevance to the inputs from the layer below [25]. Each neuron in the hidden layer sends its output to the neuron in the next layer after being subjected to the activation function. The weights and biases of the hidden layer neurons, as well as the weights and biases of the input and output layers, are iteratively adjusted during training to minimize the error between the predicted and desired outputs for a given set of input data using optimization algorithms (like backpropagation).

- Activation Function: The output of each neuron in the hidden layers is then subjected to a non-linear activation function to introduce non-linearity into the network. Capturing complex interdependencies between input features is aided by the activation function. Common activation functions include the tanh and the Rectified Linear Unit (ReLU). In this analysis, we analyze and contrast the two. Networks may learn and approximate complex relationships between inputs and outputs when non-linearity is introduced with a basic mathematical function like ReLU [26]. Because of its flexibility, practicality, and low computational cost, ReLU is quickly gaining in favor [27]. However, dead neurons (units that output zero for all inputs) may arise in ReLU training if the learning rate is too high. In the context of a Multi-Layer Perceptron, the hyperbolic tangent function (tanh) is widely used as an activation function in neural networks (MLP). The tanh function, like the sigmoid function, provides the network with non-linearity, enabling it to learn intricate associations between its inputs and its outputs [28]. The tanh function might provide more noticeable results than the sigmoid function because of its steeper slope. However, it can experience the vanishing gradient problem for extremely large or very tiny inputs, just like the sigmoid function. In a typical multilayer perceptron (MLP), the output of each neuron in a hidden layer transforms a non-linear range using the tanh function, which is applied element-wise. Because of its non-linearity, the MLP network can more effectively learn and simulate intricate data patterns and correlations.

- Output Layer: The output layer is the MLP's final layer and is responsible for producing categorization outcomes. The number of classes in a problem of classification is proportional to the number of neurons in the output layer. The output or forecast is generated by the last layer, which is informed by the calculations performed in the previous levels. The number of neurons employed in the final output layer varies from task to task. With MLP being employed, the output layer was packed with neurons, each of which represented a different class and gave back a score or probability. By calculating the error or loss between the expected outputs and the actual labels, the output layer played a crucial part in the training process. By adjusting the weights and biases of neurons across the network in light of the error, optimization techniques were used to reduce the gap between the expected and desired results.

- Training and Backpropagation: Backpropagation is used to train the MLP, which entails tweaking the network's weights to reduce the discrepancy between the projected output and the actual labels [20]. Optimizing a loss function during training involves sending the error gradient from the output layer back through the hidden layers.

- Prediction: New embedded image vectors can be fed into the network after the MLP has been trained, allowing for predictive use [21]. The MLP will produce output probabilities for each class, allowing for the highest likelihood probability to be used in the categorization of fresh images. Prediction is the process of using a trained network to generate output values for new, unknown input data. It is necessary to train the MLP on a dataset before it can make predictions on untrained data. To make predictions, we first feed the input data into a trained neural network and then extract values from the network. The output numbers mean different things depending on the purpose of the MLP's training. The accuracy of the predictions is highly dependent on the training procedure's efficiency and the quality of the training data. A more diverse and representative training set boosts an MLP's likelihood of producing accurate predictions on novel and unknown data. The MLP is a flexible and expressive classifier that can learn complex patterns and make predictions based on the extracted features from the image embedding model. It is a popular choice for various classification tasks, including image classification after image embedding, due to its ability to capture nonlinear relationships and generalize from training data.

E. 10 Fold Cross Validation

A machine learning model's efficacy and generalizability can be measured with the help of a technique called ten-fold cross-validation [29]. The dataset is split into ten equal halves (called folds), and the model is trained and evaluated several times. Ten identically sized subsets (called folds) are randomly selected from the original dataset. There are about the same number of examples in each fold. The cross-validation process is repeated 10 times. A fold is selected at each iteration to act as the validation set, while the other nine folds are used as the training set. There are nine pleats in the training set used to educate the model. The model is trained with these examples to learn regularities and associations. The validation set is the remaining fold after training is complete, and it is used to test the learned model. How well the model predicts reality. Each time a model is run through an assessment cycle, the results of those evaluations on a variety of validation sets are recorded as evaluation metrics [30], [31]. These measures reveal the general applicability of the model to different types of data. Once the 10 iterations have been completed, the overall performance of the model is estimated by averaging the metrics collected during each fold. The ability of a model to generalize to new, unseen data can be gauged more accurately by looking at its average performance. By training and testing the model on ten separate groups of data, 10-fold cross-validation yields a more accurate picture of how well it performs overall. It helps reduce the magnitude of inconsistencies that can arise from having only one training-validation split. This research improved its model selection, hyperparameter tweaking, and generalizability by using 10-fold cross-validation to provide insight into how well the models would perform on unknown data. The evaluation parameter in this study is accuracy because no cases of imbalance class were found.

III. RESULTS AND DISCUSSION

Table I displays the findings of the study. The number of neurons and activation function employed were the two hyperparameters investigated. In most cases, the proposed technique performed well. Accuracy levels of 99 % were achieved in every experiment. The network's ability to interpret and make sense of complex incoming data is made possible by neurons. The amount and nature of the input data, the complexity of the task at hand, and the design of the network as a whole are just a few of the variables that must be considered when settling on the appropriate number of neurons. It is not possible to determine an ideal number of neurons. Typically, a deeper network will have a larger number of neurons, which will allow for the extraction of more nuanced and abstract properties. However, if you add too many neurons, the network may overfit, becoming excessively specialized in the training data and failing to generalize well to novel, unknown data. One approach to settling on the optimal number of neurons is starting with a small number and gradually increasing it while keeping an eye on the network's performance on a validation set. Furthermore, the quantity of neurons may be affected by the size of the input data. For instance, it may be wasteful to have a large number of neurons in the early layers if the input images are relatively small, as these neurons would already cover a sizable fraction of the input space. However, more neurons may be needed to pick up on the degree of detail needed for bigger images or more complicated tasks. Therefore, this research examined the effects of utilizing a range of values, including 25, 50, 100, 150, and 200 neurons. The optimal number of neurons was 50. The gap between them was barely perceptible. In this research, the classification of non-linear data was better represented by a network of 50 neurons.

TABLE I. EXPERIMENTAL RESULTS

Neurons	Activation	Accuracy (%)
25	Tanh	99.71
25	Relu	99.68
50	Tanh	99.78
50	Relu	99.74
100	Tanh	99.74
100	Relu	99.68
150	Tanh	99.74
150	Relu	99.71
200	Tanh	99.71
200	Relu	99.68

The second test compared Tanh and ReLu as activation functions. Both activation functions are commonly used in neural networks and have distinct characteristics that can affect the network's performance. The activation functions play a crucial role in neural networks by introducing non-linearity to the model's decision-making process. They determine the output of a neuron or a node in a neural network, based on the weighted sum of inputs. The choice of activation function can indeed have an impact on the accuracy of a neural network. Different activation functions have distinct properties that can affect the network's learning dynamics, convergence, and generalization abilities. In this study, Tanh outperformed Relu. Tanh added a smooth non-linearity to the system. It provided a smooth transition between values and had a continuous output, making it useful in situations where it is needed. On the other hand, the output experienced jumps, and others discontinued due to the piecewise linear non-linearity introduced by ReLU. Also, when using the tanh function, data was standardized and centered.

TABLE II. CONFUSION MATRIX OF THE BEST MODEL USING 50 NEURONS AND TANH FUNCTION WHERE APHIDS (A), ARMYWORM (B), BEETLE (C), BOLLWORM (D), GRASSHOPPER (E), MITES (F), MOSQUITO (G), SAWFLY (H), AND STEM BORER (I)

Predicted

Actual	A	B	C	D	E	F	G	H	I
A	350	0	0	0	0	0	0	0	0
B	0	344	0	0	0	0	0	0	0
C	0	0	350	0	0	0	0	0	0
D	0	0	0	342	0	0	0	0	0
E	0	0	0	0	350	0	0	0	0
F	0	0	0	0	0	348	0	2	0
G	0	0	0	0	0	0	350	0	0
H	1	1	0	1	0	0	0	346	1
I	0	0	0	1	0	0	0	0	349

TABLE III. CONFUSION MATRIX OF THE WORST MODEL USING 50 NEURONS AND TANH FUNCTION WHERE APHIDS (A), ARMYWORM (B), BEETLE (C), BOLLWORM (D), GRASSHOPPER (E), MITES (F), MOSQUITO (G), SAWFLY (H), AND STEM BORER (I)

Actual \ Predicted	A	B	C	D	E	F	G	H	I
A	350	0	0	0	0	0	0	0	0
B	0	344	0	0	0	0	0	0	0
C	0	0	350	0	0	0	0	0	0
D	0	0	0	342	0	0	0	0	0
E	0	0	0	0	350	0	0	0	0
F	1	0	1	0	0	346	0	2	0
G	0	0	0	0	0	0	350	0	0
H	1	0	0	2	0	0	0	346	1
I	0	0	0	2	0	0	0	0	348

Tables II and III show the confusion matrix produced by the best and worst models. Both have a small difference, so the confusion matrices of the two are also not much different. The most difficult species to classify are sawflies and mites. In the best model, the four sawfly species were classified as aphids, beetles, grasshoppers, and stem borers. Whereas in the worst model, this data was classified as aphids, grasshoppers, and stem borers. The performance of the best model had an advantage over the worst model in classifying mosquitoes. This model classified its two data sets as sawfly. Meanwhile, the worst model incorrectly classified the four objects because it predicted them as sawflies, aphids, and stem borers. This failure was caused by the different sizes of objects in each image as well as differences in the background of the image.

IV. CONCLUSION

Based on the research that has been done, the combination model of the SqueezeNet and MLP models obtained in each experiment was above 99% for the accuracy. It shows that SqueezeNet extracted the features of the data well, while the Multi-Layer Perceptron processed these features so that the classification ran optimally. There were several classes, for example, mites, sawflies, and stem borer that failed to be properly classified. It cannot be classified properly because the background of each image was different so it was difficult to find their patterns. Therefore, segmentation between objects and backgrounds is recommended for further research.

The conducted research highlights the substantial influence of deep learning on pest identification in agriculture, showcasing its immense potential to enhance agricultural output and sustainability. This approach enables expedited and more precise identification of pests. The systems provide exceptional precision in analyzing image data, enabling the early detection of pests. Utilizing this technology, the system can independently oversee agricultural fields and detect pest infestations without the need for human involvement, thereby conserving farmers' time and labor. Enhanced pest detection enables farmers to minimize the overuse of insecticides. Consequently, this results in a more sustainable kind of agriculture that has a reduced environmental footprint. Furthermore, by promptly and precisely identifying pests, agricultural productivity may be enhanced. Optimal plant health leads to enhanced crop yields, thereby boosting agricultural output on a broader scale. Pest identification in agriculture is being advanced via the development of deep learning technologies, leading to innovation in agricultural technology. This promotes sustainable agriculture that is both more efficient and ecologically benign. The field of deep learning has made significant advancements in detecting pests in agriculture, offering extremely efficient solutions that have the potential to greatly enhance agricultural output and sustainability. In the future, it may anticipate more enhancements in plant protection and the quality of agricultural output due to ongoing technical advancements.

ACKNOWLEDGMENT

This study received support from the Associate Professor Acceleration Research 2023 initiative at the University of Padjadjaran under contract number 1549/UN6.3.1/PT.00/2023. Furthermore, we would like to extend our appreciation to the Padjadjaran Academic Recharging 2024 program, administered by Universitas Padjadjaran.

REFERENCES

[1] C. Chen, Y. Liang, X. Tang, M. Dai, and K. Zhou, “The Research of Pest Detection in Granary Based on Yolov4,” SSRN Electronic Journal, 2022, doi: 10.2139/ssrn.4007163.

[2] J. Liu and X. Wang, “Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network,” Front Plant Sci, vol. 11, 2020, doi: 10.3389/fpls.2020.00898.

[3] P. Kalkura, P. R. B, S. K. N, Surya, and Ms. Ramyashree, “Pest control management system using organic pesticides,” Global Transitions Proceedings, vol. 2, no. 2, 2021, doi: 10.1016/j.gltp.2021.08.058.

[4] J. Liu and X. Wang, “Plant diseases and pests detection based on deep learning: a review,” Plant Methods, vol. 17, no. 1. 2021. doi: 10.1186/s13007-021-00722-9.

[5] J. Kaushal, M. Khatri, and S. K. Arya, “A treatise on Organophosphate pesticide pollution: Current strategies and advancements in their environmental degradation and elimination,” Ecotoxicology and Environmental Safety, vol. 207. 2021. doi: 10.1016/j.ecoenv.2020.111483.

[6] B. Tugrul, E. Elfatimi, and R. Eryigit, “Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review,” Agriculture (Switzerland), vol. 12, no. 8. 2022. doi: 10.3390/agriculture12081192.

[7] N. T. Sinshaw, B. G. Assefa, S. K. Mohapatra, and A. M. Beyene, “Applications of Computer Vision on Automatic Potato Plant Disease Detection: A Systematic Literature Review,” Computational intelligence and neuroscience, vol. 2022. 2022. doi: 10.1155/2022/7186687.

[8] A. Servin et al., “A review of the use of engineered nanomaterials to suppress plant disease and enhance crop yield,” Journal of Nanoparticle Research, vol. 17, no. 2. 2015. doi: 10.1007/s11051-015-2907-7.

[9] K. V. Ramesh, V. Rakesh, and E. V. S. Prakasa Rao, “Application of big data analytics and artificial intelligence in agronomic research,” Indian Journal of Agronomy, vol. 65, no. 4, 2020.

[10] A. Tzachor, M. Devare, B. King, S. Avin, and S. Ó hÉigeartaigh, “Responsible artificial intelligence in agriculture requires systemic understanding of risks and externalities,” Nature Machine Intelligence, vol. 4, no. 2. 2022. doi: 10.1038/s42256-022-00440-4.

[11] M. Türkoğlu and D. Hanbay, “Plant disease and pest detection using deep learning-based features,” Turkish Journal of Electrical Engineering and Computer Sciences, vol. 27, no. 3, 2019, doi: 10.3906/elk-1809-181.

[12] N. C. Kundur and P. B. Mallikarjuna, “Insect Pest Image Detection and Classification using Deep Learning,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 9, 2022, doi: 10.14569/IJACSA.2022.0130947.

[13] Y. Li and J. Yang, "Few-shot cotton pest recognition and terminal realization," Comput Electron Agric, vol. 169, 2020, doi: 10.1016/j.compag.2020.105240.

[14] Y. Peng and Y. Wang, "CNN and transformer framework for insect pest classification," Ecol Inform, vol. 72, 2022, doi: 10.1016/j.ecoinf.2022.101846.

[15] D. Argüeso et al., "Few-Shot Learning approach for plant disease classification using images taken in the field," Comput Electron Agric, vol. 175, 2020, doi: 10.1016/j.compag.2020.105542.

[16] X. Xu, H. Zheng, Z. Guo, X. Wu, and Z. Zheng, "SDD-CNN: Small data-driven convolution neural networks for subtle roller defect inspection," Applied Sciences (Switzerland), vol. 9, no. 7, 2019, doi: 10.3390/app9071364.

[17] S. Pargaien, D. Singh, R. Prakash, V. P. Dubey, H. Pant, and A. V. Pargaien, "Land Use Classification of Kathgodam Region using Transfer learning-based approach," in 2022 2nd International Conference on Artificial Intelligence and Signal Processing, AISP 2022, 2022. doi: 10.1109/AISP53593.2022.9760539.

[18] S. Verma, A. Chug, and A. P. Singh, "Application of convolutional neural networks for evaluation of disease severity in tomato plant," Journal of Discrete Mathematical Sciences and Cryptography, vol. 23, no. 1, 2020, doi: 10.1080/09720529.2020.1721890.

[19] Y. Yang et al., "A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions," Comput Biol Med, vol. 139, 2021, doi: 10.1016/j.compbiomed.2021.104887.

[20] I. N. Yulita, F. A. Hariz, I. Suryana, A. S. Prabuwono, "Educational innovation faced with COVID-19: deep learning for online exam cheating detection," Education Sciences, 13(2), 2023. doi: 10.3390/educsci13020194.

[21] I. N. Yulita, R. R. Julviar, A. Triwahyuni, and T. Widiastuti, "Multichannel electroencephalography-based emotion recognition using machine learning," In Journal of Physics: Conference Series, 2019. doi: 10.1088/1742-6596/1230/1/012008.

[22] Y. Zhou, Y. Niu, Q. Luo, and M. Jiang, "Teaching learning-based whale optimization algorithm for multi-layer perceptron neural network training," Mathematical Biosciences and Engineering, vol. 17, no. 5, 2020, doi: 10.3934/MBE.2020319.

[23] I. N. Yulita, N. A. Amri, and A. Hidayat, "Mobile application for tomato plant leaf disease detection using a dense convolutional network architecture," Computation, 11(2), 2023. doi: 10.3390/computation11020020

[24] Lorencin, N. Anđelić, J. Španjol, and Z. Car, "Using multi-layer perceptron with Laplacian edge detector for bladder cancer diagnosis," Artif Intell Med, vol. 102, 2020, doi: 10.1016/j.artmed.2019.101746.

[25] M. Zhu, G. Zhang, L. Zhang, W. Han, Z. Shi, and X. Lv, "Object Segmentation by Spraying Robot Based on Multi-Layer Perceptron," Energies (Basel), vol. 16, no. 1, 2023, doi: 10.3390/en16010232.

[26] D. Boob, S. S. Dey, and G. Lan, "Complexity of training ReLU neural network," Discrete Optimization, vol. 44, 2022, doi: 10.1016/j.disopt.2020.100620.

[27] B. Hanin, "Universal function approximation by deep neural nets with bounded width and ReLU activations," Mathematics, vol. 7, no. 10, 2019, doi: 10.3390/MATH7100992.

[28] T. De Ryck, S. Lanthaler, and S. Mishra, "On the approximation of functions by tanh neural networks," Neural Networks, vol. 143, 2021, doi: 10.1016/j.neunet.2021.08.015.

[29] K. Jung, D. H. Bae, M. J. Um, S. Kim, S. Jeon, and D. Park, "Evaluation of nitrate load estimations using neural networks and canonical correlation analysis with K-fold cross-validation," Sustainability (Switzerland), vol. 12, no. 1, 2020, doi: 10.3390/SU12010400.

[30] S. Arlot and M. Lerasle, "Choice of v for V-fold cross-validation in least-squares density estimation," Journal of Machine Learning Research, vol. 17, 2016.

[31] H. L. Vu, K. T. W. Ng, A. Richter, and C. An, "Analysis of input set characteristics and variances on k-fold cross validation for a Recurrent Neural Network model on waste disposal rate estimation," J Environ Manage, vol. 311, 2022, doi: 10.1016/j.jenvman.2022.114869.