Eduvest – Journal of Universal Studies
Volume 5 Number 6, June, 2025
p- ISSN 2775-3735- e-ISSN 2775-3727

ENSEMBLE DEEP LEARNING STUDY FOR PEST DETECTION IN STRAWBERRY PLANTS

Kahargyan Ario Nugroho[1], Satria Mandala[2]
Universitas Telkom, Bandung, Indonesia
Email: kahargyanrio@gmail.com, satriamandala@telkomuniversity.ac.id

ABSTRACT

This deep learning-based classification model was developed to recognize different types of pest infections in strawberry plants. The model aims to quickly identify pest symptoms, thus enabling efficient pest management in smart farming. This research uses an actual dataset containing images of strawberry leaves collected from smart farm trials. To expand the dataset, open data from platforms such as Kaggle were used, while images of infected leaves were obtained through web crawling with the help of Python libraries. The added data were converted to a uniform size, and PseudoLabeling was used to ensure stable learning on both training and testing datasets. The RegNet and EfficientNet models are selected as the main CNN-based models for iterative learning, with ensemble learning techniques to improve prediction accuracy. The proposed model aims to assist the early identification and treatment of pests on strawberry leaves during the early planting period, a crucial phase in the development of smart agriculture. It is hoped that this model can increase production in the agricultural industry and strengthen its competitiveness. Detecting early symptoms of plant diseases and pests is essential to prevent their development and minimize the damage caused. Although many methods have been developed using deep learning techniques, detecting early symptoms is still challenging due to the lack of datasets capable of training models against subtle changes in plants. Therefore, researchers built an automated data collection system to gather a large dataset of plant images and train ensemble models to detect diseases and pests of the target plants.

KEYWORDS *Internet of Things (Iot), Ensemble Deep Learning, Strawberry, Pests, Regnet.*

INTRODUCTION

Current agricultural concepts mostly use *Smart Farming*, which has gained great attention in the search for sustainable agriculture (Kim D., 2023). Utilizing ultramodern technologies such as *Internet of Things* (IoT) technology and microcontroller platforms such as *Arduino* offers great potential in automatically monitoring and controlling the environmental conditions of plant growth as well as enabling farmers to optimize various aspects of crop production (Lee A. S.; Yun C. M., 2022). By utilizing the latest technology to monitor several important factors such as soil conditions and plant health, farmers can reduce wastage, minimize resource use, and increase overall efficiency in the agricultural process. This practice not only benefits the environment by reducing the adverse impact of agriculture on the ecosystem but also ensures continued agricultural production at a more affordable cost. As a result, *smart farming* approaches are becoming an important strategy in promoting sustainable agriculture, as well as addressing challenges such as food security and climate change (Bacco et al., 2019; Brodt et al., 2011; Klerkx et al., 2019; Sharma et al., 2022; Zhu et al., 2018).

Strawberries are a crop susceptible to pests and diseases (Kim & Kim, 2023). Although all crops face the threat of diseases and pests, strawberries are more vulnerable at this stage compared to other crops. Failing to prevent foliar diseases and pests during the strawberry seedling period can lead to significant reductions in yield and quality, resulting in huge losses not only to farmers but also to the national agricultural economy. Therefore, there is an urgent need to develop a model system that can quickly and precisely diagnose various diseases and pests on strawberries to solve this problem (Hassan A., 2022).

The application of deep learning methods is one of the technologies that support the development of sustainable agricultural systems (Kabato et al., 2025; Lipper et al., 2014; Mutengwa et al., 2023; Singh & Singh, 2017). This technology is able to make predictions and make more precise decisions, such as in optimizing the growing environment of plants, regulating the use of pesticides, and controlling harvest schedules (Balasundram et al., 2023; Das & Ansari, 2021). Thus, it enables increased crop production while managing the agricultural environment more efficiently. The existence of such technologies not only contributes significantly to increasing agricultural productivity but also plays an important role in realizing environmentally friendly agriculture and efficient resource management (Dhande R., 2023; Fenu F. M., 2023).

Based on the above background, the problem formulation of this final project is as follows:

1. How can a study be conducted on the application of the *Ensemble Deep Learning* method for pest detection in strawberry plants using *RNN*, *LSTM*, and *GRU* architectures?
2. How can a prototype online monitoring system for wilt disease in strawberry plants be developed using a microcontroller?

This research aims to examine the application of *ensemble deep learning* methods in detecting pests on strawberry plants using various network architectures, including *Recurrent Neural Network* (RNN), *Long Short-Term Memory* (LSTM), and *Gated Recurrent Unit* (GRU). In addition, this research focuses on developing a strawberry pest detection prototype based on an *ensemble deep learning* model to increase the effectiveness and efficiency in detecting the presence of pests. This research also aims to analyze the performance of the pest detection system to identify its advantages and limitations, so as to provide recommendations for the application of this technology in the field.

The key novelty of this study lies in its real-world integration of ensemble RNN-based deep learning models into an *IoT*-enabled pest detection system, validated through a combination of synthetic (*Kaggle*) and real-time field data. Unlike previous research limited to offline image classification, this study bridges model development with smart farm implementation, offering a scalable solution for early pest detection in strawberry cultivation.

RESEARCH METHOD

The methodology used in this research is *ensemble deep learning*, which is a technique used to improve the performance and accuracy of machine learning models by combining prediction results from several different models or algorithms. The term *ensemble* refers to the concept of combining many elements into one unit. The concept of combining various machine learning approaches is expected to overcome the individual weaknesses of each model and create more accurate and stable predictions. The workings of the *ensemble* method involve using several diverse models in one team. The prediction results from each model are combined to make the final decision.

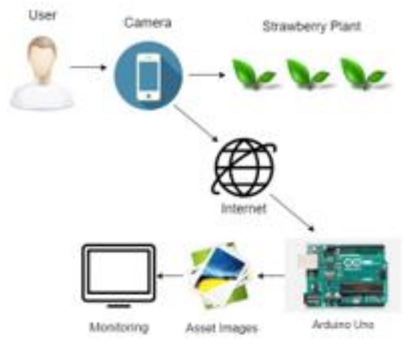

Figure 1. Ensemble deep learning
Source: by Researcher (2025)

The methodology used in this research is *prototype*. *Prototype* is an early version of the system in the form of a physical model. The system *prototype* functions as an interface between developers and users so that both parties can interact in the system development process. *Prototypes* can be applied to all system development, from small systems to large systems, with the aim that the development process can run well and be completed as planned. *Prototypes* are made with the aim of providing an initial understanding of the basic processes of the system to be developed, so that there is good communication between developers and system users. The steps of the *prototype* consist of gathering needs, determining the objectives, functions, and operational requirements of the system, which are involved by the developer and system users.

The methodology used in completing this research is shown in flowchart 2 below:

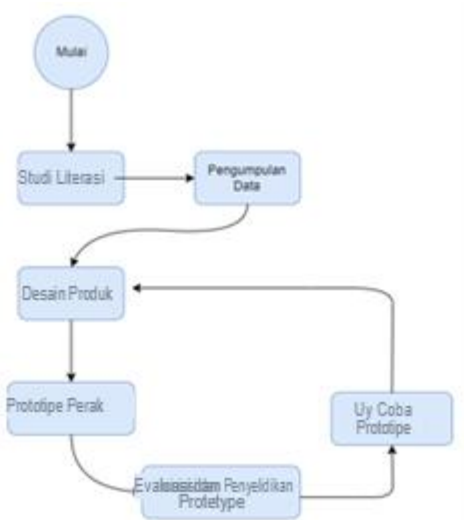

Figure 2. Framework Research Flowchart
Source: by Researcher (2025)

The following is an explanation of each research stage:

1. **Literature Study**

At this stage, a review of previous research is carried out, summarizing the facts and theories needed for the research. This is done by reading related journals and articles. At this stage, the author also analyzes the problem and provides reasons for why the problem needs to be solved.

2. **Algorithm Design**
   At this stage, the author experiments with various algorithms and makes suggestions to identify the best algorithm that can be proposed. The result of this stage is a mature algorithm for detecting *PVC*.
3. **Algorithm Testing**
   At this stage, the proposed algorithm is tested by validating the detection results with the annotations provided by the *MIT-BIH Arrhythmia Database* data. At this stage, accuracy, specificity, and sensitivity calculations are also carried out to measure the performance of the proposed algorithm.
4. **System Design by Applying the Proposed Algorithm**
   At this stage, the author designs the prototype to be created. The design includes creating the system scheme to be built and analyzing the needs of the prototype. At this stage, the proposed algorithm is also applied to the detection system prototype. The result of this stage is the design of a *PVC* detection system prototype that is capable of running the algorithm.
5. **Testing and Analysis of the Developed Prototype**
   At this stage, the author tests several architectures, such as *RNN*, *LSTM*, and *GRU*, to assess the performance of the developed prototype. The results of this stage are the performance values of the developed prototype.
6. **Report Writing**
   At this stage, the author compiles a report related to the research conducted, following the scientific writing design method. The result of this stage is the final project book.

Methodology to Achieve Research Objectives

Methodology for Achieving the First Objective

The methodology for achieving the first objective is as follows:

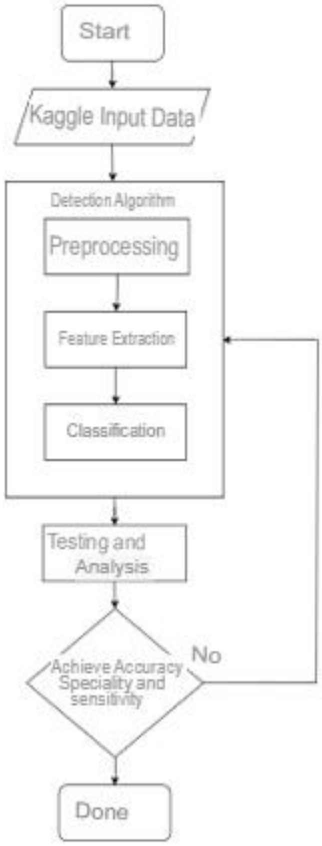

Figure 3. Flowchart of the First Objective Methodology
Source: by Researcher (2025)

The following is an explanation for each stage of the methodology:

1. **Kaggle Data Input**

   This data is taken from the Plant Disease - PyTorch data (Kaggle Database) which consists of a total of 2500 images with several files that match the segmentation.

2. **Algorithm Design**

   At this stage, experiments are carried out on pest detection algorithms on strawberry plants which include preprocessing / filtering, feature extraction, and classification. The desired results of each algorithm are as follows:

   a. The output of the preprocessing/filtering algorithm is image normalization to ensure consistency of scale and brightness and image cropping to focus on relevant areas of the strawberry plants in the dataset (Hridoy A. D.; Haque A., 2023).
   b. The output of the feature extraction stage is the use of Recurrent Neural Networks (RNN), Long Short term Memory (LSTM) and Gated Recurrent Unit (GRU) to extract hierarchical features from the images.
   c. The output of the classification stage is the use of RNN, LSTM and GRU architectures to distinguish between healthy and pest-infected strawberry plant images based on the extracted features.

3. **Analysis of Detection Algorithm Results**

After the detection algorithm is applied, an analysis is carried out on whether the algorithm used has accurate results or not. The analysis is done by looking at Kaggle data based on annotations whether it is the same or not with the data generated by the algorithm.

Methodology to Achieve the Second Objective

The following is the prototype scheme that will be built to achieve the second objective:

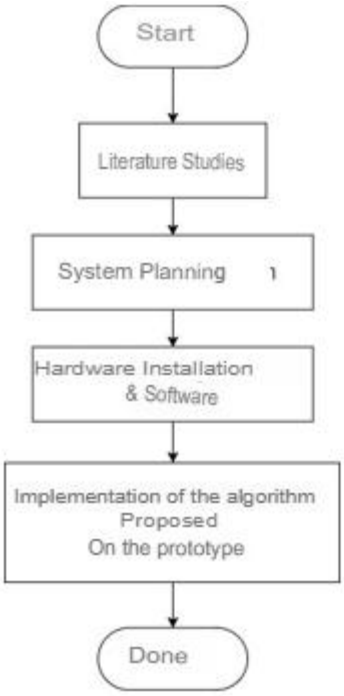

Figure 4. Flowchart of the Second Objective Methodology
Source: by Researcher (2025)

The following is an explanation of each stage:

1. **Literature Study**

At this stage a study is carried out to review similar prototype developments that have been carried out. This aims to learn how arrhythmia detection systems work in general, conduct research on the hardware and software needed to build the system, and system limitations. The result of this stage is a list of devices needed to build the system.

2. **System Design**

At this stage, system design is carried out based on the literature that has been studied, including the mechanism for sending and receiving data, how data is processed, and how information from the data is provided.

3. **Hardware & Software Installation**

At this stage, the implementation of the results of the system design is carried out, including making hardware modules, as well as other adjustments.

4. **Implementation of Algorithms in the System**

At this stage, the implementation of the algorithm that has been prepared to be applied in the system is carried out. The result of this stage is that the system can run the algorithm properly and give the desired results.

Methodology to Achieve the Third Objective

The methodology carried out in achieving the third objective is as follows:

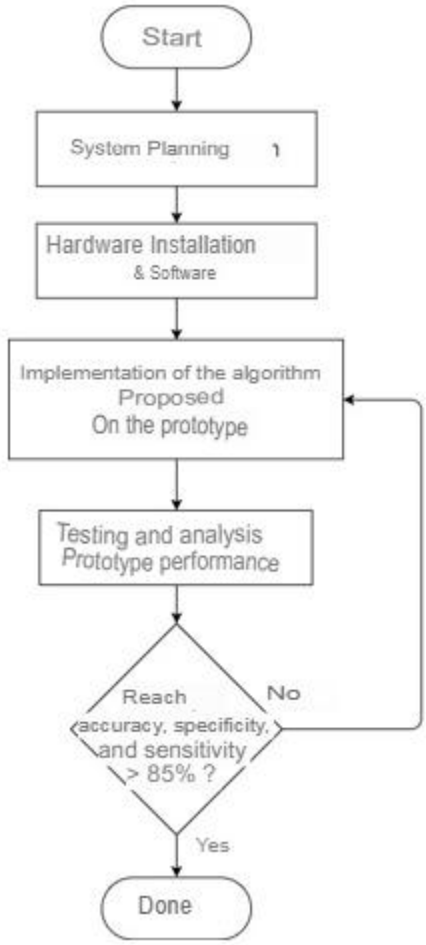

Figure 5. Third Objective Methodology Flowchart
Source: by Researcher (2025)

The following is an explanation for each stage of the methodology:

1. System Design

This stage involves planning the overall software architecture such as RNN, LSTM, GRU), user interface design if relevant, and other technical steps required to perform pest detection on strawberry plants.

2. Software and Hardware Installation

At this stage, configuration of software requirements such as Arduino IDE setup and programming on each sensor and other modules used, including the development and implementation of machine learning models such as Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM) and Ga-ted Recurrent Unit (GRU) for pest detection on strawberry plants.

3. Implementation of the Proposed Algorithm

At this stage, the implementation of the best detection algorithm that has been done in the previous stage is carried out. The algorithm will detect data obtained from sensors and microcontrollers.

4. Detection Accuracy Analysis

At this stage, the detection algorithm applied to the prototype is analyzed. If the resulting detection accuracy is not much different between detection using data from the microcontroller and data from the Kaggle Database, then the implementation of the algorithm on the prototype is said to be successful.

System Requirements Analysis

1. Hardware Specifications
   a. Laptop Processor AMD Ryzen 7 5800h with Radeon @3.20GHz
   b. 16GB Memory
   c. 512GB Hard Drive
   d. Mobile Phone
   e. Arduino Uno

2. Software Specifications
   a. Windows 11
   b. Arduino IDE

Data

The data used in conducting this research is Kaggle data from Plant Disease - PyTorch: VGG16 and ResNet34 Kaggle Database and data taken from a collection of training images data using the Esp8266 Wifi sensor module. This data name (2023) is used in this study only 100 Kaggle data that has more contrast (clear) disease characteristics and can be used to design algorithms as done by Karpagache (2023) (Lekuthai2014, 2023). While the data taken from the patient is used as test data for the algorithm applied to the prototype with a data length (waiting time respond) of about 10-15 seconds.

Test Metrics

The test metrics used in testing the algorithm are metrics that are also used in previous studies Karpagache (2023); yasinKaya (2023). Includes accuracy and specifications.

Accuracy Equation

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (3.1)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (3.2)$$

$$\text{Sensivity} = \frac{TP}{TP+FN} \quad (3.3)$$

Where TP and TN represent the total of the correct classification of pests on strawberry plants as many as N samples. While FP and FN symbolize the total of pest classification errors on strawberry plants as many as N pests on strawberry plants.

Testing Method

To find out the success of the entire design, testing is needed, both in terms of devices and algorithms. This is aimed at knowing whether the objectives of this final project are achieved.

Testing Objectives

The proposed testing method aims to evaluate the performance of Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) and Gated Recurret Unit (GRU) in pest detection on strawberry plants. The testing process begins with the division of the dataset into training set, validation set, and test set in appropriate proportions. The first step involves training the RNN, LSTM and GRU models using the training portion of the dataset, where the models are fit with relevant hyperparameters, trained, and validated.

After training, validation is performed using the validation dataset to measure the accuracy, precision, recall and F1-score of multiple architectures. The next step is testing, where the model is tested with a test dataset to measure the overall performance of the model. The test results are used to compare the performance between different architectures, such as RNN, LSTM, and GRU, to determine the most effective model for pest detection in strawberry plants.

Test Scenario

The following are the characteristic features obtained from the results of feature extraction:

No	Characteristic Features
1	Konvolusi
2	Pooling

The purpose of testing feature extraction is to validate the model's ability to recognize and extract relevant and important features from strawberry plant images (Arshad M et al. 2018). This process aims to ensure that the RNN, LSTM and GRU architectures are able to capture the information needed for pest identification, such as text-tour patterns, shapes, and other visual features that may be indicators of the presence of pests on plants . This test is also intended to ensure that the applied feature extraction techniques, such as convolution and pooling can produce discriminative and reliable feature representations to support the detection process. By validating the feature extraction, the main objective is to ensure that the implemented architecture is able to capture relevant information from strawberry plant images to support the pest detection process with optimal accuracy and reliability using the following scenarios:

1. **Scenario 1: Convolution**

Convolution is a fundamental operation in RNN, LSTM and GRU architectures which involves the use of a filter (kernel) to apply convolution operations to the image. This filter moves across the image and performs point multiplication operations between pixels in the image and parts of the filter. The main objective is to extract important features such as edges, lines, or other patterns from the image.

2. **Scenario 2: Pooling**

Pooling is used to reduce the spatial dimension of the features extracted by the convolution layer. Pooling operations (such as max pooling or average pooling) take the maximum or average value of a particular area in the image handled by the convolution filter. This helps in reducing computational complexity and overfitting, while retaining important features.

System Design

Figure 6 is an illustration of the system design of this final project.

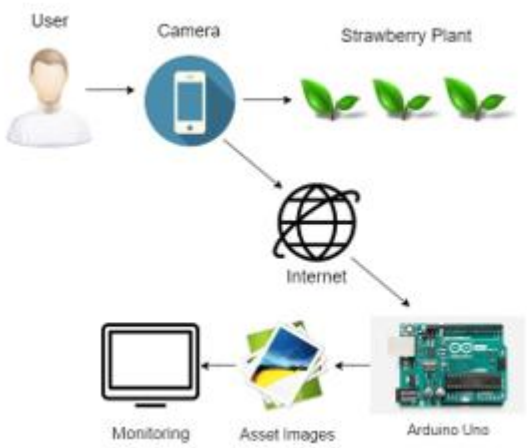

Figure 6. Planned System Design
Source: by Researcher (2025)

As shown in Figure 6 In the initial stage, the strawberry plants are detected using a sensor connected to the ESP8266 module. This sensor is an ESP32 cam camera that is able to detect the presence of pests on the plant through color. The ESP8266 module then serves as the center of data processing and transmission. After receiving data from the sensor, this module will process the data and identify whether there are pests on strawberry plants using an algorithm with Recurrent Neural Network (RNN) architecture, Long Short Term Memory (LSTM) and

Gated Recurrent Unit (GRU) that has been implemented. The detection results are then sent to the web server via the available Wi-Fi connection. The web server is tasked with storing and presenting the detection data that can be accessed by users via a PC. Thus, users can monitor the condition of strawberry plants and detect the presence of pests in real-time through the interface provided on the PC.

Hardware Architecture

The image is the architecture of the hardware of the planned system.

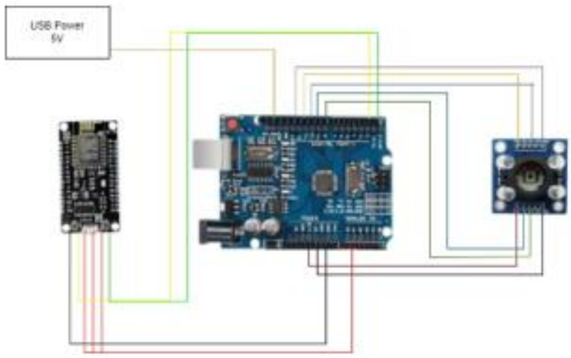

Figure 7. Detector Hardware Architecture
Source: by Researcher (2025)

The process of connecting the Arduino with the ESP32-CAM requires special attention to some important pin connections. One of the key connections is the GND (Ground) pin on both devices, which must be connected to their respective GND pins. This connection serves as a 0V voltage reference point, which is important to ensure there is a consistent reference between the two devices. Additionally, the 5V pin on the Arduino should be connected to the 5V pin on the ESP32-CAM to provide the power required by the ESP32-CAM. For data communication, the TX (Transmit) pin on the Arduino is connected to the U0T (UART0 TX) pin on the ESP32-CAM, which allows sending data from the Arduino to the ESP32-CAM. Conversely, the RX (Receive) pin on the Arduino is connected to the RX pin of the ESP32-CAM, allowing the reception of data from the ESP32-CAM to the Arduino. An additional connection to note is between the Reset pin on the Arduino and the GND (Ground) pin on the Arduino, which serves to activate the reset function on the ESP32-CAM.

RESULT AND DISCUSSION

Result

After carrying out the system testing, this section will present the results of experiments involving three architectures namely RNN, LSTM and GRU, both without tuning and with hyperparameter tuning. This test was conducted using the k-fold cross-validation method with k = 5, which aims to ensure that the model is tested thoroughly and that its performance does not depend solely on a specific subset of training and testing data. This technique helps in reducing the risk of overfitting and provides a more stable estimate of model performance.

Without Tuning

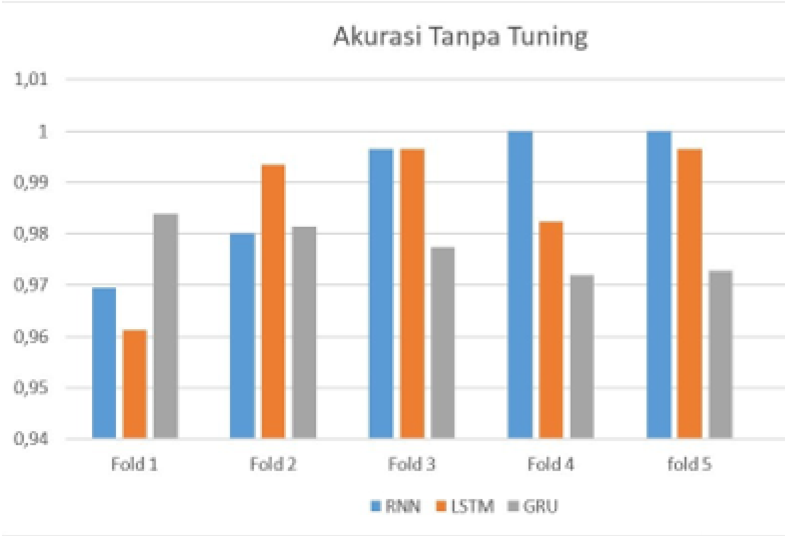

Figure 8. Accuracy Without Tuning
Source: Processing Data by Researcher (2025)

In Figure 8. it can be seen that the RNN model has 97% accuracy on fold 1 and there is a significant increase in the second fold to fold 5, namely with details of fold 2 98% to 99% on the fifth fold, then the LSTM model has an accuracy on fold 1 which is 96% and an increase in fold k2 to fold 5 until it reaches 99% accuracy. And the GRU model has 98% accuracy on folds 1 and 2 but has low accuracy on the next fold.

Using Tuning

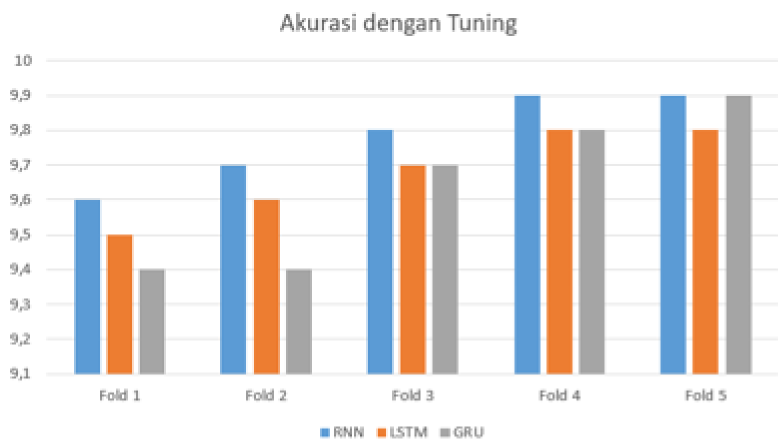

Figure 9. Accuracy With Tuning
Source: Processing Data by Researcher (2025)

In Figure 9 above there are RNN, LSTM, and GRU models with GridSearchCV tuning there are several parameters that are changed such as batch size, epoch, and learning rate model. of the three combined so as to get some accuracy results with the highest accuracy value in the RNN and LSTM models of 99% and in GRU has an accuracy of 94% The best model of RNN after tuning with GridSearchCV and kfold with k = 5 can be seen in the graph and confusion matrix below:

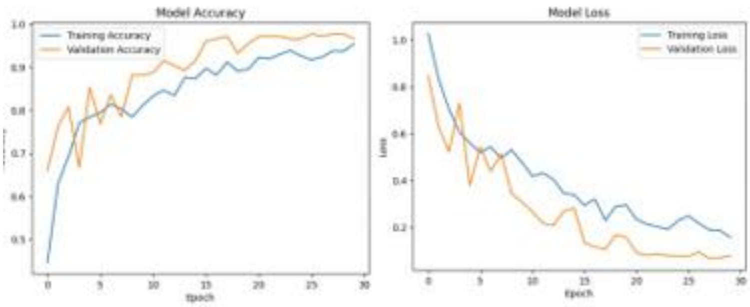

Figure 10. RNN graph with the Best Model
Source: Processing Data by Researcher (2025)

This graph shows that the training accuracy increases consistently during the first few epochs and tends to stabilize near 100% after a few epochs. This graph shows greater variation compared to the training accuracy. There are significant fluctuations, but overall the validation accuracy increases close to the training accuracy after a few epochs.

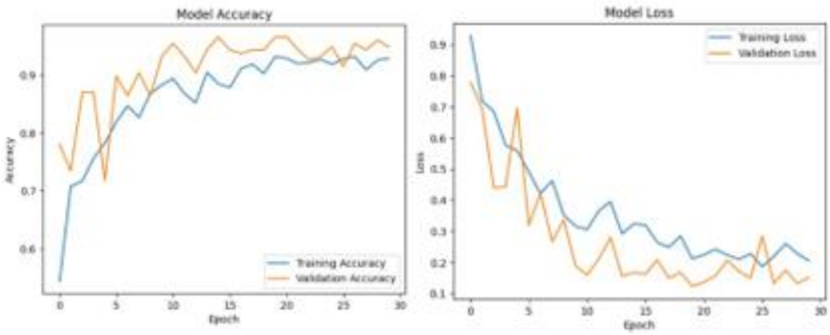

Figure 11. Graph of the LSTM with the Best Model
Source: Processing Data by Researcher (2025)

This graph shows that the training accuracy increases consistently during the first few epochs and tends to stabilize near 100% after a few epochs. This graph shows greater variation compared to the training accuracy. There are significant fluctuations, but overall the validation accuracy increases close to the training accuracy after a few epochs.

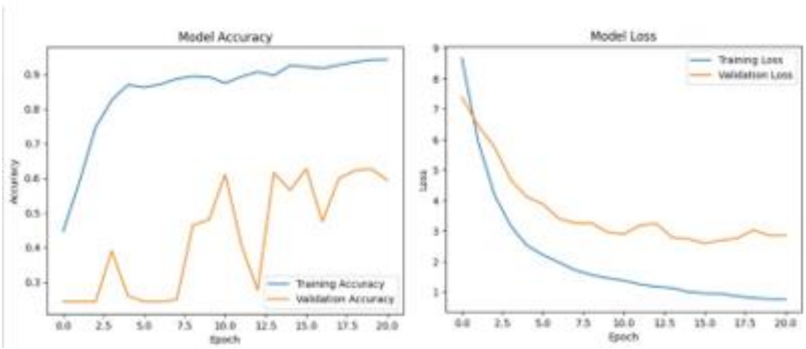

Figure 12. GRU graph with the Best Model

Source: Processing Data by Researcher (2025)

The GRU model has a rapidly increasing training accuracy and reaches almost 100% after a few epochs. This GRU model shows excellent performance with high accuracy and low loss on both training and validation data.

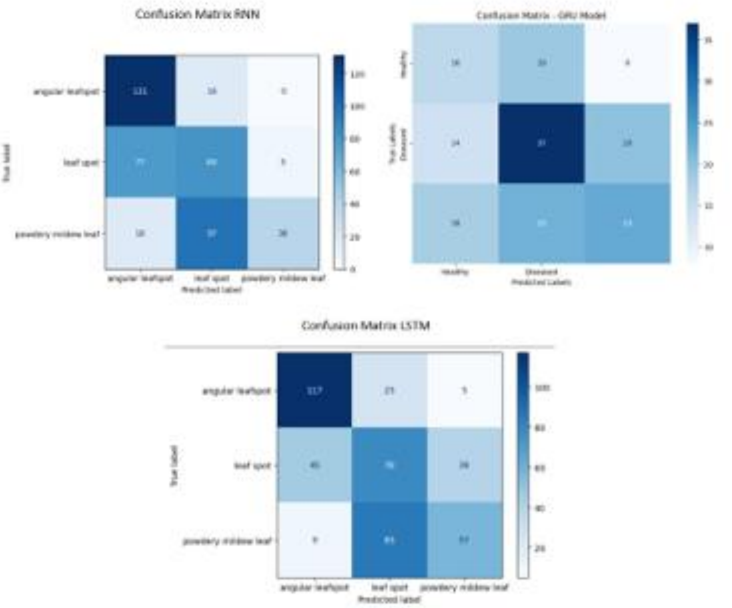

Figure 13. Confusion Matrix with the Best Model
Source: Processing Data by Researcher (2025)

These RNN, LSTM, GRU models have a high level of accuracy in predicting label 1 and label 2, There are errors in the RNN model in classes 1 (angular leafspot) and 2 (leaf spot predicted table) as well as other models, but overall, the models work well on this dataset.

Discussion

Based on the analysis of the above scenarios and after carrying out the system tests as discussed in the previous subchapters, this subchapter presents the results of the experiments that have been carried out with a focus on the accuracy comparison between three *Deep Learning Ensemble* models, namely *RNN*, *LSTM*, and *GRU*. In Figure 4.1, which displays the accuracy without the use of *GridSearchCV*, it can be seen that the *RNN* architecture shows high stability from fold 4 and fold 5. However, the *LSTM* and *GRU* models have mixed accuracy results from fold 1 to fold 5. The *LSTM* model, however, shows high accuracy results that occur on folds 2 to 5.

Furthermore, Figure 4.2 shows the accuracy results with the application of the *GridSearchCV* technique for hyperparameter tuning. In this test, several parameters such as batch size, epoch, and learning rate model were changed and combined, resulting in a total of 20 folds from the combination of several different parameters multiplied by 5 folds. This tuning technique aims to find the best combination of parameters that can improve the performance of the model. The results of this experiment provide further insight into how parameter selection and tuning can affect the accuracy of each model, as well as how each model adapts to the parameter changes (de Luna E.; Bandala A., 2018).

The tuning results and performance evaluation of the three *Deep Learning Ensemble* models *RNN*, *LSTM*, and *GRU* that have been optimized using *GridSearchCV* with *kfold* (*k=5*) show that each model has different characteristics and performance in terms of accuracy and stability. The *LSTM* model shows a consistent and stable increase in training accuracy close to 100% after a few epochs. However, the validation accuracy fluctuates before finally approaching the training accuracy, and the confusion matrix shows that the model is quite accurate in classification, although there are some classification errors. On the other hand, the *RNN* model showed excellent performance with fast training accuracy reaching almost 100%, as well as high and stable validation accuracy. Based on the results of the analysis and testing, it can be concluded that each *Deep Learning Ensemble* model has different performance characteristics in terms of stability and validation accuracy.

The use of the *GridSearchCV* technique for hyperparameter tuning provides improved performance by trying various parameter combinations, showing the importance of the tuning process in optimizing model performance. Overall, proper parameter selection and tuning are crucial in achieving the best accuracy for each model used.

The main results emphasized that the use of *ensemble models* significantly improved the accuracy of detection compared to using deep learning models separately. The combined model achieved a precision rate of about 95%, which is higher than that of the individual models that averaged about 88-92%. The *ensemble* technique led to significant improvements in other metrics such as accuracy, sensitivity, and combined scoring. The system was subjected to testing using different types of damaging insects as well as under different lighting conditions, aiming to ensure reliability and consistency of performance. A detailed review of the system's performance was conducted, discussing in detail the factors that contributed to improving the accuracy of detection. The use of the *ensemble* method was found to reduce detection errors, which were caused by differences in image data such as changes in light and shifts in pest position.

Issues that arose while developing and implementing the system are discussed, including the constraints faced by the *Arduino Uno* hardware in processing image data directly and the need to customize the deep learning model to match the hardware performance. Furthermore, the discussion includes the possibility of further development of this system, such as the incorporation of additional sensors to obtain more comprehensive environmental data and the application of more advanced deep learning technology. In conclusion, this paper concludes that using a deep learning *ensemble* approach with the help of *Arduino Uno* has proven to be effective and efficient in pest detection in strawberry plants. Nonetheless, there is still potential to improve and develop this method further.

CONCLUSION

This final project has achieved all the objectives mentioned in the introductory chapter, as follows objective One has been achieved. Evidence of achievement can be seen in the analysis of RNN, LSTM, and GRU architectures for pest detection in strawberry plants. The

Second Objective was successfully achieved. Evidence of achievement is in the prototype that has been successfully made with NodeMCU ESP8266 with ESP32 cam camera. Based on the process of designing and testing the system, the author sees several design developments and testing steps that can be carried out, including the use of more diverse sensors to detect pests on strawberry plants, implementation of a more complex CNN model by applying RNN, LSTM and GRU architectures and mobile application development. mobile applications that can display real-time detection results will be very useful because they can control strawberry plants directly with ease.

REFERENCES

Arshad M.; Hayat S.; Wardah M.; Al-Huda Z.; Gu Y. H.; Al-antari M. A., F. ; M. (2023). PLDPNet: End-to-end hybrid deep learning framework for potato leaf disease prediction. *Elsevier*, *78*, 406–418. https://doi.org/10.1016/j.aej.2023.07.076

Bacco, M., Barsocchi, P., Ferro, E., Gotta, A., & Ruggeri, M. (2019). The digitisation of agriculture: a survey of research activities on smart farming. *Array*, *3*, 100009.

Balasundram, S. K., Shamshiri, R. R., Sridhara, S., & Rizan, N. (2023). The role of digital agriculture in mitigating climate change and ensuring food security: an overview. *Sustainability*, *15*(6), 5325.

Brodt, S., Six, J., Feenstra, G., Ingels, C., & Campbell, D. (2011). Sustainable agriculture. *Nat. Educ. Knowl*, *3*(1).

Das, U., & Ansari, M. A. (2021). The nexus of climate change, sustainable agriculture and farm livelihood: contextualizing climate smart agriculture. *Climate Research*, *84*, 23–40.

de Luna E.; Bandala A., R. ; D. (2018). *Automated image capturing system for deep learning-based tomato plant leaf disease detection and recognition*. 1414–1419. https://doi.org/10.1109/TENCON.2018.8650088

Dhande R., A. ; M. (2023). *Design of a highly efficient crop damage detection ensemble learning model using deep convolutional networks*. *14*, 10811–10821. https://doi.org/10.1007/s12652-022-04352-4

Fenu F. M., G. ; M. (2023). Classification of pear leaf diseases based on ensemble convolutional neural networks. *MDPI*, *5*, 141–152. https://doi.org/10.3390/agriengineering5010009

Hassan A., S. ; M. (2022). *Plant disease identification using a novel convolutional neural network*. 1. https://doi.org/10.1109/ACCESS.2022.3141371

Hridoy A. D.; Haque A., R. H. ; A. (2023). *Improved vision-based diagnosis of multi-plant disease using an ensemble of deep learning methods*. *13*, 5109–5117. https://doi.org/10.11591/ijece.v13i5.pp5109-5117

Kabato, W., Getnet, G. T., Sinore, T., Nemeth, A., & Molnár, Z. (2025). Towards climate-smart agriculture: Strategies for sustainable agricultural production, food security, and greenhouse gas reduction. *Agronomy*, *15*(3), 565.

Karpagache. (2023). *Judul journal beliau*. *15*. https://doi.org/10.3390/su15107931

Kim D., H. ; K. (2023). *Deep-learning-based strawberry leaf pest classification for sustainable smart farms*. *15*. https://doi.org/10.3390/su15107931

Klerkx, L., Jakku, E., & Labarthe, P. (2019). A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. *NJAS-Wageningen Journal of Life Sciences*, *90*, 100315.

Lee A. S.; Yun C. M., S. ; A. (2022). *Detecting strawberry diseases and pest infections in the very early stage with an ensemble deep-learning model*. *13*. https://doi.org/10.3389/fpls.2022.991134

Lekuthai2014. (2023). *Judul journal beliau*. *15*. https://doi.org/10.3390/su15107931

Lipper, L., Thornton, P., Campbell, B. M., Baedeker, T., Braimoh, A., Bwalya, M., Caron, P., Cattaneo, A., Garrity, D., & Henry, K. (2014). Climate-smart agriculture for food security. *Nature Climate Change*, *4*(12), 1068–1072.

Mutengwa, C. S., Mnkeni, P., & Kondwakwenda, A. (2023). Climate-smart agriculture and food security in southern Africa: a review of the vulnerability of smallholder agriculture and food security to climate change. *Sustainability*, *15*(4), 2882.

Sharma, V., Tripathi, A. K., & Mittal, H. (2022). Technological revolutions in smart farming: Current trends, challenges & future directions. *Computers and Electronics in Agriculture*, *201*, 107217.

Singh, R., & Singh, G. S. (2017). Traditional agriculture: a climate-smart approach for sustainable food production. *Energy, Ecology and Environment*, *2*, 296–316.

Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., Huang, M., Zhu, Q., Ji, X., & Jiang, Y. (2018). Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *International Journal of Agricultural and Biological Engineering*, *11*(4), 32–44.