INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Deep Metric Learning with Augmented Latent Fusion and Response-Based Knowledge Distillation on Edge Device for Paddy Pests and Disease Identification

Hendri Darmawan [a], Mike Yuliana [a,*], Moch. Zen Samsono Hadi [a], Arun Kumar Sangaiah [b]

[a] *Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Sukolilo, Surabaya, Indonesia*
[b] *International Graduate School of AI, National Yunlin University of Science and Technology, Yunlin, Taiwan*
Corresponding author: [*]mieke@pens.ac.id

***Abstract*—The health of paddy fields significantly impacts rice yields and the economic stability of farmers. Limited number of experts available to watch these issues poses a challenge. Consequently, a reliable diagnostic system is necessary to find pests and diseases in rice crops. In this study, we propose deep metric learning with augmented latent fusion (FADMAKA) combined with a response-based knowledge distillation (KD) approach. The student model, which processes single RGB input images, is trained using soft latent labels derived from four augmented input from the teacher model. Our method delivers a high validation accuracy of 0.973, keeps an accuracy of 0.782 on the unseen data, and with rapid inference time of 38.911 milliseconds. This approach's accuracy outperforms SoftMax deep learning classification with fine-tuning, which only has a maximum accuracy of 0.739 on the unseen data with computation time of 36.224 ms, and the DML with augmented latent fusion with k-NN classifier on the same base model, which achieves an accuracy of 0.78 with computation time of 124.977 ms. Our proposed model has 0.12 giga floating point operations per second (GFLOPs) that is suitable for edge devices with low computational resources. Following the modeling phase, we deployed the highest-accuracy student model to a Raspberry Pi 4B device equipped with a camera. This system can provide biological agent-based recommendations for identified pest and disease threats in rice fields. Our approach not only improved accuracy but also proved efficiency, enabling farmers to identify pests and disease without relying on internet connectivity.**

***Keywords*— Knowledge distillation; deep metric learning; model compression; paddy plant pests; disease.**

Manuscript received 5 Aug. 2024; revised 10 Sep. 2024; accepted 21 Oct. 2024. Date of publication 30 Nov. 2024.

I. INTRODUCTION

The health of paddy fields is crucial for farmers, as it directly affects crop yields and their economic stability. According to the International Rice Research Institute, pests and diseases can significantly reduce rice yield by up to 37%, with variations ranging from 24% to 41% depending on environmental and farming conditions [1]. One particularly destructive pest is the brown planthopper (*Nilaparvata lugens*), which causes considerable damage to rice crops [2]. Additionally, paddy fields are vulnerable to diseases such as leaf blight, brown spot, and blast, which are prevalent in the South Asian region [3]. In Indonesia, the situation is exacerbated by the limited number of agricultural experts available to monitor these problems. Typically, detecting plant diseases manually involves expert observation with the naked eye, which is time-consuming, costly, and prone to errors [4]. By machine learning, it is possible to show diseases in plants at an early phase, which could enhance the longevity of crops [5]. Therefore, a fast and exact system is needed to prevent outbreaks of pests and diseases in rice crops. A recommendation system is of paramount importance in guiding farmers to take necessary actions against potential threats before they escalate into severe outbreaks. While several existing products can find rice pests or diseases, they often focus solely on recognition and do not provide eco-friendly recommendations. Furthermore, these models are not deployed in a device that allows farmers to use them for identification in real-world environments.

Our earlier research proposed a system for paddy pests and disease identification using the deep metric learning (DML) with augmented latent fusion (FADMAKA) [6]. This model was deployed on cloud computing systems, providing real-time monitoring through the concept of a 5D world map. In this research, we address the challenge of internet

connectivity for cloud computing platforms. Specifically, we deploy the model on an edge device, cutting the need for an internet connection during inference, as internet access is often unavailable in remote fields. We also aim to improve the model complexity and computational time from our earlier work [6]. This is particularly crucial because deploying neural network models on edge devices presents a challenge due to the limited computational and memory capabilities.

Numerous methods to find pests and diseases in rice crops have been explored in past research. For instance, Ni et al. [7] proposed a novel model RepVGG_ECA, which integrates efficient channel attention blocks in RepVGG model to improve the feature extractor. This model, which is specifically designed for classifying rice pests and diseases, utilizes convolutional neural network (CNN) with SoftMax deep learning classification techniques. However, its effectiveness is constrained by its substantial dependence on supervised learning, which requires labeled samples. Rahman et al. [8] explored the use of CNN architectures for recognizing pests and disease in crops. Their research yielded the best performance with a fine-tuned VGG16 model. Malathi et al. [9] employed SoftMax classification techniques to classify pests in rice plants. They achieved the highest accuracy with the ResNet-50 model, which was further fine-tuned for their specific task. Our previous studies have shown that these methods, when applied to our dataset, yield a reasonably high level of accuracy [6]. However, these methods predominantly rely on highly complex models, leading to high computational costs. As a result, they are not suitable for deployment on devices with low specifications.

Several research studies have proposed the use of a simple CNN architecture for classifying images of pests and diseases in rice plants using a SoftMax classifier. One such study was conducted also by Rahman et al. [8], who proposed a simple CNN for classifying pests and diseases in paddies. Petchiammal et al. [10] proposed optimized deep CNN model architectures, such as PaddyNet, for classifying paddy diseases. They used dropout to reduce overfitting problem. Other studies have proposed an optimized approach that does not use CNNs as feature extractors for finding diseases in rice plants. Instead, these studies use a classical image processing approach, and the extracted features are fed into neural networks. For instance, Ramesh et al. [11] proposed the JAYA algorithm. They removed the image background using a fusion thresholded image saturation part of HSV and RGB images and then extracted color features and texture features using GLCM from the segmented image from k-MEANS. A simple neural model is then used to classify rice leaf diseases based on these features. Lu et al. [12] employed a method that combined histogram equalization, median filtering, and edge segmentation to process images of rice sheath disease. They combined color and texture features as the input for backpropagation (BP) neural network by concatenating them. Despite its innovative approach without CNN, these methods may not fully capture complex image patterns due to its limited feature representation. Additionally, high-dimensional input data can lead to the curse of dimensionality and overfitting. Moreover, AI algorithms used for image classification are based on deep learning techniques, such as SoftMax classification. SoftMax classification that employ cross-entropy loss are not well-suited where there is significant variation within classes and limited variation between classes in the input data distribution [13].

In this paper, we propose the DML technique, which is trained to recognize similarities in images by mapping data onto latent representations that can handle high intraclass and low interclass variances. DML also can find new classes without retraining the model, making it suitable for identifying the numerous pests and diseases found in nature. Additionally, unlike traditional classification methods, DML can be used for image retrieval applications because it can query databases for similar images. We propose the FADMAKA-KD algorithm, which compress the complex DML teacher model using augmented latent fusion from our previous research [6] to the lightweight MobileNetV3s student model through knowledge distillation (KD) technique. Our experiments focus on improving the accuracy of lightweight student model by using the robust large teacher model so that enables deployment on low-specific hardware and achieving effective results. At the end of this research, we also deployed the model to a smart portable device for pests and disease identification in rice crops using an edge device integrated with a high-resolution camera and touchscreen display. In Table 1, we present comparative study between existing research conducted by earlier researchers in the context of pest or disease identification and the features of our proposed research. The paper is organized as follows: Section 2 provides an in-depth explanation of the proposed approach, Section 3 discusses the experimental setup, results, and analysis, and Section 4 concludes with a summary of the main insights.

TABLE I
RESEARCH COMPARATION ON PADDY PEST AND DISEASE CLASSIFICATION APPROACH USING COMPUTER VISION TECHNIQUE

Authors	Approach	Lightweight model	Model feature		Model deployment feature		
			Pest	Disease	Implementation method	Offline Usage	Follow-up recommendation
Darmawan et al. [6]	DML with augmented latent fusion: FADMAKA	–	✓	✓	Web-based application with cloud deployment model	–	✓
Ni et al. [7]	SoftMax classification: RepVGG_ECA	–	✓	✓	–	–	–
Rahman et al. [8]	SoftMax classification: VGG16 (fine-tuned)	–	✓	✓	–	–	–
Malathi et al. [9]	SoftMax classification: ResNet-50	–	✓	–	–	–	–
Rahman et al. [8]	SoftMax classification: SimpleCNN	✓	✓	✓	–	–	–

Authors	Approach	Lightweight model	Model feature		Model deployment feature		
			Pest	Disease	Implementation method	Offline Usage	Follow-up recommendation
Petchiammal et al. [10]	SoftMax classification: PaddyNet	✓	–	✓	–	–	–
Ramesh et al. [11]	Classical algorithm for feature extraction and fed it into neural network model: JAYA algorithm	✓	–	✓	–	–	–
Lu et al. [12]	Classical algorithm for feature extraction and fed it into neural network model	✓	–	✓	–	–	–
Ours	**Model compression for DML with augmented latent fusion: FADMAKA-KD**	✓	✓	✓	**Portable device with edge inference system**	✓	✓

II. Materials and Methods

The entire research framework is shown in Fig. 1, with six main components, is explained in that follows: (1) acquiring images, enhancing using augmentation techniques, and preprocessing; (2) training the teacher model; (3) obtaining the enhanced latent fusion image as soft latent labels from the teacher model and training the student model; (4) k-NN retrieval modeling; (5) accuracy testing and comparing to baseline and state-of-the-art (SOTA) techniques; (6) edge device deployment.

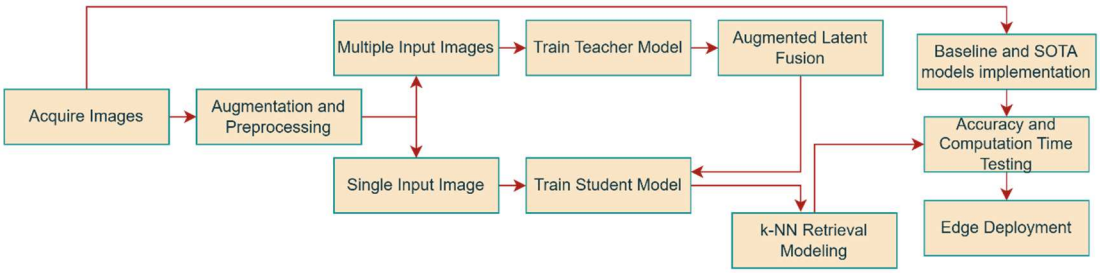

Fig. 1 Research framework workflow diagram

A. Dataset Preparation and Preprocessing

In our study, we used the same dataset previously employed in our previous research that used to be teacher models in this research [6]. This dataset comprises six distinct classes: three related to rice diseases (rice blast, brown spot, yellow rice borer) and three associated with pests (bacterial leaf blight, brown planthopper, rice leafhopper). The dataset is partitioned into 8,859 training samples, 2,224 validation samples, and 600 test samples [6]. To collect this data, we sourced information from various channels, including web scraping from Google Images, the IP102 pest dataset from field environments, rice plant disease images from field environments, and a rice leaf disease dataset with a white background. Additionally, we performed manual analysis to increase the dataset. This involved cropping images into square shapes and separating those having multiple diseases, thereby increasing the overall sample size. We also corrected mislabeled images and showed and removed duplicates. Importantly, we ensured that the test dataset remained distinct from the validation data to prevent any data leakage. Our dataset served as the foundation for training and evaluating our proposed method, as well as for reimplementing existing approaches for direct comparison. Fig. 2 visually depicts the distribution of the dataset.

Fig. 2 Sample distribution with visual example of each class

During the training process, we also applied image augmentation techniques, including blurring, random rotation, and horizontal flipping. These techniques enhanced the diversity of the training data, mitigating overfitting and

improving generalizability. Furthermore, we preprocessed the images by enhancing contrast and normalizing the image using the standard deviation and mean from the ImageNet dataset. Our base network for student models leveraged pretrained weights from IMAGENET1K_V1.

B. Distill and Relation-based Knowledge Transfer

In our earlier work, we introduced a novel method known as "latent fusion augmented images", which utilizes DML as the teacher model [6]. This method is characterized by the augmentation of input images during the inference process, followed by the computation of the average of the resulting latent representations. The final latent representation serves as the soft latent labels, i.e., $z^{f\prime}$, produced by the teacher model, which the student model learns to mimic. We used an offline distillation training approach, where the teacher model is trained first and then used to train the student model, as shown in Fig. 3.

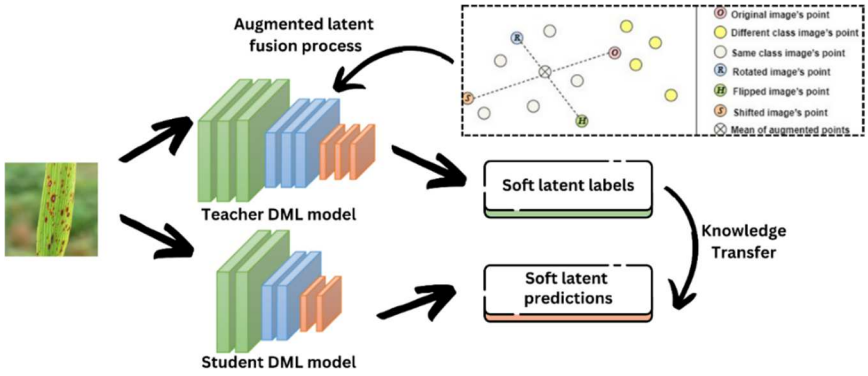

Fig. 3 Knowledge distillation on DML with augmented latent fusion

Algorithm 1: FADMAKA-KD

1 **Input:**
Training set $\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$,
Validation set $\mathcal{D}_v = \{(x_i^v, y_i^v)\}_{i=1}^{n_v}$,
Holdout set $\mathcal{D}_e = \{(x_i^e, y_i^e)\}_{i=1}^{n_e}$,

2 Data augmentation and preprocessing
- Augment $\mathcal{D}_t$ using blur, random rotation, and horizontal flip
- Preprocess x_i^t, x_i^v, x_i^e using contrast stretching and image normalization

3 Train student model to mimic teacher model
- Load W', b' the pretrained teacher model $t(\cdot)$
- Define the student model $s(\cdot)$ using MobileNetv3s and load $\widehat{W}, \hat{b}$ from ImageNet and produce $z \in R^h$
- Collect soft latent labels using $t(x_i^t)$

$$z_i^{t\prime} = \{t(x_i^t; W', b') \mid x_i^t \in \mathcal{D}_t\}$$
$$z_i^{r\prime} = \{t(Rotate(x_i^t, \gamma);\ W', b') \mid x_i^t \in \mathcal{D}_t\}$$
$$z_i^{h\prime} = \{t(HFlip(x_i^t);\ W', b') \mid x_i^t \in \mathcal{D}_t\}$$
$$z_i^{s\prime} = \{t\big(Shift\big(x_i^t, (\Delta p, \Delta q)\big);\ W', b'\big) \mid x_i^t \in \mathcal{D}_t\}$$
$$z_i^{f\prime} = \frac{1}{4}(z_i^{t\prime} + z_i^{r\prime} + z_i^{h\prime} + z_i^{s\prime})$$

- Collect soft latent predictions using $s(x_i^t)$

$$z_i^t = \{s(x_i^t; \widehat{W}, \hat{b}) \mid x_i^t \in \mathcal{D}_t\}$$

- Define $\mathcal{L}_D$ from Eq. (4) using distance metric from Eq. (1) to Eq. (3)
- Optimize $s(\cdot)$ to achieve final W_t', b_t' using SGD or AdamW to solve Eq. (1)
- Validate and checkpoint $s(\cdot)$ using x_i^v

4 - Compute latent for training and validation set

$$z^{\mathrm{t}} = \{s(x_i^t; \mathrm{W}, \mathrm{b}) \mid x_i^t \in \mathcal{D}_t\}$$
$$z^{\mathrm{v}} = \{s(x_i^v; \widehat{W}, \hat{b}) \mid x_i^v \in \mathcal{D}_v\}$$

- Find the best k value for maximizing z^v accuracy in z^{t}
- Compute latent for holdout set

$$z^e = \{s(x_i^e; W, b) \mid x_i^e \in \mathcal{D}_e\}$$

- Classify the z^e using k-NN in z^t and measure accuracy with y_i^e as the labels

Output: predicted class of x_i^e

We assigned the teacher model as $t(\cdot)$, which the parameters were previously trained with FADMAKA algorithm, denoted as W', b'. We obtained the soft latent labels, i.e., z^t, using the average value of the latent representation from different input images using the teacher model. After that, we optimize the parameters of the student model, i.e., W, b, using the distillation loss defined by Eq. (1). The objective of the distillation loss is to align the student model's soft latent predictions with the teacher model's soft latent labels. d is a specific similarity metric defined by Eq. (2)-Eq. (4).

$$\widehat{W}, \hat{b} = \underset{W_t, b_t}{arg\,min}\, \mathcal{L}_D$$
$$\mathcal{L}_D = \sum_{i=1}^{N} min\left(d\left(z_i^{f\prime}, z_i^t\right)\right) \tag{1}$$

C. Training a Student Model Using a Response-based Knowledge Scheme

The loss function measures the error between the prediction and the label [14]. To mimic the teacher model's soft latent labels, the student model was trained with a distillation loss function, optimizing weights and biases during backpropagation [15]. The whole method for our FADMAKA-KD is outlined in Algorithm 1. The similarity metrics used in this study are the Euclidean distance (Eq. 2), cosine distance (Eq. 3), and Pearson correlation (Eq. 4).

The distillation loss metric is determined in accordance with what is used to optimize the teacher model [6]. We also employed the same specific optimizers, AdamW and SGD,

which were used to train the teacher model. These were implemented using 0.001 initial learning rate and 16 batch size.

$$d(z^{f\prime}, z^t) = \sqrt{\sum_{d=1}^{h} \left(z_d^{f\prime} - z_d^t\right)^2} \tag{2}$$

$$d(z^{f\prime}, z^t) = 1 - \frac{\sum_{d=1}^{h} z_d^{f\prime} z_d^t}{\sqrt{\sum_{d=1}^{h} (z_d^{f\prime})^2 \sum_{d=1}^{h} (z_d^t)^2}} \tag{3}$$

$$d(z^{f\prime}, z^t) = 1 - \frac{\sum_{d=1}^{h} (z_d^{f\prime} - \overline{z^{f\prime}})(z_d^t - \overline{z^t})}{\sqrt{\sum_{d=1}^{h} (z_d^{f\prime} - \overline{z^{f\prime}})^2 \sum_{d=1}^{h} (z_d^t - \overline{z^t})^2}} \tag{4}$$

The number of training epochs was 100 for 16-dimensional and 200 for 256-dimensional latent representations. We did not use the 1024-dimensional representation due to its low performance in previous research and to minimize processing time on edge devices. Moreover, k-NN classifier uses high-dimensional data can increase computation also suffer from the curse of dimensionality [16].

D. Teacher–student Network Architecture

We employed ResNet-50 and ResNet-152 as the teacher networks due to their extensive number of parameters, which also contribute to their high accuracy. The base model ResNet-50, employed as a teacher in this study, boasts as many as 25,557,032 parameters with 8.26 giga floating point operations per second (GFLOPs). Moreover, the ResNet-152 model possesses an even larger parameter count, totaling 60,192,808 with 23.21 GFLOPs. GFLOPs is a measure of the computational complexity of a neural network model. Generally, the inference time of the neural network model tends to increase linearly with the number GFLOPs. Thus, higher GFLOPs will take longer inference times [17]. In addition, the teacher model that used ResNet base model performed augmented latent fusion during inference using four different image characteristics, resulting in a computational cost that scales linearly with the number of images [6]. On the other hand, we selected MobileNetV3s as the student model due to its ability to reduce the number of parameters and computations needed to process the image. The base model MobileNetV3s has a parameter count of 927,008 with 0.12 GFLOPs. This makes it an ideal choice for mobile and embedded systems with limited computational resources [18]. Remarkably, we trained the student model only using one input image, which can significantly reduce the computational time during inference since we don't need to augment the image to do latent fusion.

E. Classifier and retrieval task

The trained student model creates augmented latent from a single image and uses k-NN to classify them, calculating distances to all training data latent and ranking nearest neighbors. Within the scope of finding pests and diseases, the k-NN algorithm enables users to verify predictions by comparing input images with corresponding images from the database. We conducted dimensionality reduction on the latent representation generated by the student model using the t-SNE algorithm, with an automatically optimized learning rate and a perplexity of 15. Based on Fig. 4, there is an overlap between the groups representing rice blast and brown spot. This suggests that some images may contain more than one type of disease, which aligns with previous research findings that images of brown spot can also show indications of rice blast on the same leaf [6]. Furthermore, the brown planthopper cluster also contained overlapping points from the rice leafhopper sample. This could be due to similarities in wing features between the two classes, as they share anatomical similarities.

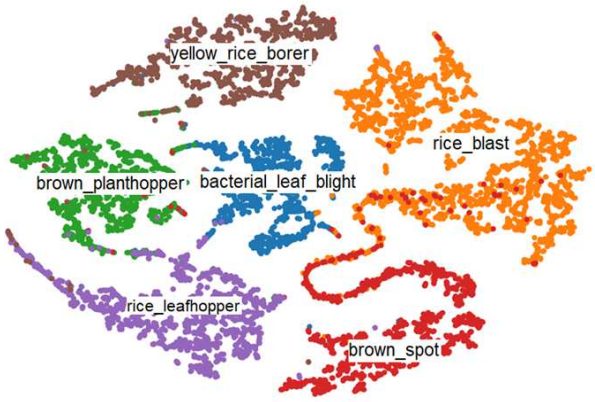

Fig. 4 Latent space from student model with t-SNE

We also conducted a quantitative analysis of silhouette scores across three distinct scenarios. The first scenario involved the best model from previous research without latent fusion [6], which was a ResNet50 optimized with Pearson correlation using AdamW and a 16-dimensional latent. This model achieved a silhouette score of 0.918. The second scenario was the best teacher model from earlier research [6], with latent fusion, which achieved a silhouette score of 0.900. The third scenario was FADMAKA-KD using the MobileNetV3s base model, which achieved a silhouette score of 0.878. A decrease in the silhouette score with augmented latent fusion can occur because this technique involves calculating the mean of the data points, which are altered through augmentation. The silhouette score evaluates how well a data point is grouped within its own cluster relative to other clusters, tends to decrease when this means significantly deviates from the center of the original cluster due to augmentation. Meanwhile, the student model that attempts to mimic the augmented latent fusion only uses a single input, causing the model to struggle to align closely with the teacher cluster that utilizes multiple inputs. This results in a diminished ability of the model to maintain cohesion within its own cluster, reflected in the lowered silhouette score.

F. Experimental Modeling and Performance Comparison

We tested various knowledge distillation modeling schemes on the small MobileNetV3s student model, which transferred knowledge from the ResNet-50 and ResNet-152 teacher models and was optimized with Euclidean distance, cosine distance, and Pearson correlation metrics. We also

evaluated the performance of MobileNetV3s, used as the base model of the FADMAKA algorithm, in the two best scenarios from our previous research [6] to examine the effectiveness of the FADMAKA-KD algorithm. These were Pearson correlation with the AdamW optimizer and 16-dimensional latent space and Euclidean distance with the AdamW optimizer and 256-dimensional latent space, with the objective function in Eq. (5) [19].

$$\widehat{W'}, \hat{b}' = \underset{W_t', b_t'}{arg\,min}\, \mathcal{L}_T$$
$$\mathcal{L}_T = \sum_{i=1}^{N} \max\left(d\begin{pmatrix} f(X_i; w_t', b_t'), \\ f(X^p;\, w_t', b_t') \end{pmatrix} - d\begin{pmatrix} f(X_i;\, w_t', b_t'), \\ f(X^n;\, w_t', b_t') \end{pmatrix} + m, 0\right) \quad (5)$$

We also compared the performance of the SoftMax classifier with supervised learning objective loss in Eq. 6 to determine the effectiveness of the FADMAKA-KD against SoftMax classification approach.

$$\mathcal{L}_X = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} \log\left(p_{ij}\right) \quad (6)$$

Furthermore, we compared the results of FADMAKA-KD with SOTA models previously studied in the context of paddy plant pests and disease classification, all of which were reimplemented using our dataset. We also assessed the inference computation time on a Raspberry Pi 4B edge device, which is the device intended for deployment. Given that the image classification task involves balanced class categories in the holdout set, we used accuracy as the primary metric to compare the best performance across different approaches. The accuracy is calculated by Eq. 7 [20].

$$Accuracy = \frac{TruePos+TrueNeg}{TruePos+FalsePos+TrueNeg+FalseNeg} \quad (7)$$

G. Model Implementation on Edge Devices

Few studies have implemented models for end-user applications. In this study, we deployed the best-performing student model on a Raspberry Pi 4B edge device as shown in Fig. 5, enabling farmers to use it in the field without relying on internet connectivity [21]. The Raspberry Pi 4B features a Broadcom BCM2711 SoC, which integrates four Cortex-A72 cores running at 1.5 GHz, along with 4 GB of LPDDR4-3200 RAM [22]. In our application, we integrated biological agent recommendations to reduce the reliance on chemical pesticides.

Fig. 5 Smart portable device alpha version

This aligns with the Indonesian Ministry of Agriculture's goal to preserve the ecosystem by minimizing pesticide use, protecting natural predators, and reducing harmful residues. This system is invaluable for promptly identifying and managing the health of paddy fields, helping farmers protect their crops effectively. The device employs an Arducam 64 MP camera complemented by compact LCD touchscreen. This combination enhances the device's portability.

III. Results and Discussion

This section presents the evaluation of the model's training results. Furthermore, we conducted comparative analysis to show the effectiveness of FADMAKA-KD by comparing it against other methods used in the paddy pests and disease classification.

A. Model Checkpointing on the Validation Set and Classifier Selection

Our first experiment aimed to determine the best k-value from the validation set, evaluating k-values within the range $k \in \{1,6,11,16\ldots,201\}$ to achieve the highest accuracy on the student model. Fig. 6 shows the k-NN accuracy on the latent representation validation set, varying k-values and different similarity metrics for each scenario of the student model.

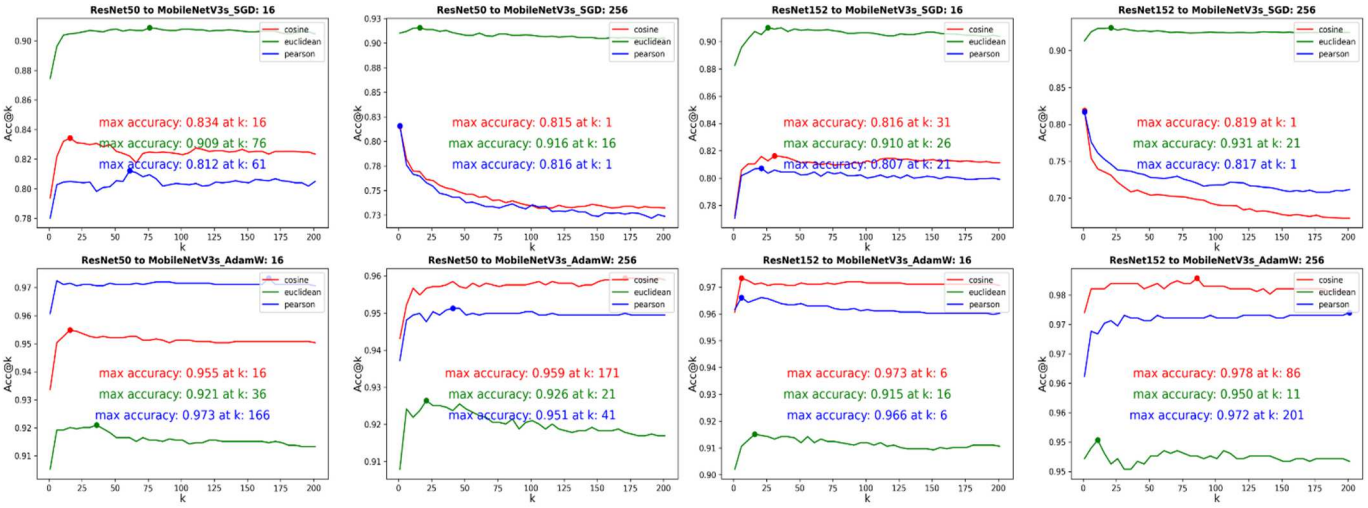

Fig. 6 Validation set accuracy of FADMAKA-KD with different k-NN configurations across all schemes

Fig. 6 illustrates that the selection of k profoundly influences the accuracy [23]. An optimal k-value is required to balance accuracy and computational efficiency, as a lower k-value increases noise sensitivity, while a higher k-value increases computational complexity. The trends in Fig. 6 resemble those of the teacher model in previous research, suggesting successful knowledge transfer from the teacher to the student. The Euclidean distance metric performs well when the student model is optimized using SGD, achieving an average accuracy of 0.916. However, this is still lower than the accuracy achieved with AdamW. Specifically, the SGD optimizer results in lower accuracies with the cosine metric at 0.821 and the Pearson metric at 0.813.

In general, AdamW yields superior validation accuracy across all metrics, achieving 0.928 for Euclidean, and 0.966 for both cosine and Pearson. The choice of optimizer and similarity metric significantly changes the performance of a DML model [24]. Adaptive learning rate and weight decay in AdamW is probably effective with distance metrics sensitive to the size of the weights, such as the cosine distance and Pearson correlation. In contrast, SGD stays effective with the Euclidean distance, which are less sensitive to weight magnitudes

B. Analysis on the Unseen Data/holdout Data

In this subsection, we evaluate the classification performance using the optimal k value obtained during validation testing. Subsequently, we apply this model to previously unseen data and present the results in Tables 2 and 3. Table 2 summarizes the accuracy results for the FADMAKA-KD student model with the teacher model for ResNet-50.

TABLE II
FADMAKA-KD HOLDOUT SET ACCURACY WITH RESNET-50-TEACHER MOBILENETV3S-STUDENT ON THE BEST K-NN SETTINGS

Optimizer	Latent dims	Max epochs	Triplet cosine				Triplet Euclidean				Triplet Pearson			
			Teacher		Student		Teacher		Student		Teacher		Student	
			k	Acc	k	Acc	k	Acc	k	Acc	k	Acc	k	Acc
SGD	16	100	46	0.679	16	0.627	31	0.730	76	0.635	31	0.690	61	0.613
	256	200	1	0.733	1	0.587	11	0.752	16	0.67	1	0.723	1	0.593
AdamW	16	100	106	0.745	16	0.74	16	0.745	36	0.738	**36**	**0.772**	**166**	**0.782**
	256	200	31	0.740	171	0.745	21	0.742	21	0.695	16	0.712	41	0.742

TABLE III
FADMAKA-KD HOLDOUT SET ACCURACY WITH RESNET-152-TEACHER MOBILENETV3S-STUDENT THE BEST K-NN SETTINGS

Optimizer	Latent dims	Max epochs	Triplet cosine				Triplet Euclidean				Triplet Pearson			
			Teacher		Student		Teacher		Student		Teacher		Student	
			k	Acc	k	Acc	k	Acc	k	Acc	k	Acc	k	Acc
SGD	16	100	96	0.723	31	0.62	26	0.733	26	0.673	86	0.710	21	0.628
	256	200	1	0.667	1	0.583	11	0.750	21	0.687	1	0.715	1	0.612
AdamW	16	100	21	0.783	6	0.708	6	0.713	16	0.73	6	0.755	6	0.741
	256	200	6	0.733	**86**	**0.753**	**31**	**0.785**	11	0.691	6	0.760	201	0.733

Meanwhile, Table 3 provides the outcomes for the teacher model ResNet-152. During these evaluations, we explored various hyperparameters, including optimizers (SGD and AdamW), latent dimensions (16 and 256), and epochs (100 and 200). Additionally, we incorporated checkpoint mechanisms and early stopping. The loss function employed similarity metrics such as Cosine, Euclidean, and Pearson.

On average, the AdamW optimizer outperformed SGD across all the metrics, achieving mean accuracies of 0.737, 0.714, and 0.750 for the cosine distance, Euclidean distance, and Pearson correlation, respectively. In contrast, SGD yielded mean accuracies of 0.604, 0.666, and 0.612 for the same metrics. Thus, the AdamW optimizer with the Pearson correlation metric generally provides higher accuracy in unseen data. Our best model, derived from a ResNet50-teacher and MobileNetV3s-student setup, achieved the highest accuracy of 0.782 using AdamW and the Pearson correlation metric, with a latent dimension of 16 and k=166. This shows a 1.3% accuracy improvement over the teacher model, compared to the highest single-input accuracy of 0.772 on ResNet50 base model [6]. The optimal model from the ResNet152-teacher and MobileNetV3s-student setup achieved a peak accuracy of 0.753 with AdamW and the cosine distance metric at a latent dimension of 256 and k=86. Meanwhile, the teacher model has the highest accuracy for ResNet152 with latent dimensions of 256, using Euclidean distance and AdamW, achieving 0.785. However, this didn't translate to high accuracy in FADMAKA-KD scheme due to training differences, as the student model mimics the latent fusion from augmented inputs, not just a single pure teacher's output which produces accuracy of 0.691. Therefore, the teacher model's highest accuracy does not necessarily translate to the highest student model accuracy, particularly when using augmented latent fusion during model training. FADMAKA-KD can introduce added diversity and robustness into the learning process, as the student model is exposed to a wider variety of data representations.

In this paper, we also re-implemented vanilla DML (without augmented during inference) and FADMAKA techniques using the MobileNetV3s base model. Additionally, we compared the performance of these methods with the best accuracy achieved by teacher models based on ResNet-50 and ResNet-152. We proved a new k-value from the validation set, and the results are shown in Table 4.

TABLE IV
UNSEEN DATA ACCURACY OF FADMAKA ALGORITHM VS FADMAKA-KD ALGORITHM WITH MOBILENETV3S INFERENCE ON RASPBERRY PI 4B

Teacher model	Base model	KD	Distance Triplet loss	Total parameters	GFLOPs	Augmented inference	Latent dims	k	Inf. Time (ms)	Acc
–	ResNet-50	–	Pearson	25,573,050	8.26	–	16	36	576.368	0.772
–	ResNet-50	–	Pearson	25,573,050	33.04	Rotation, shift, flip	16	36	2312.472	0.920
–	ResNet-152	–	Euclidean	60,241,986	23.21	–	1024	6	2342.7	0.789
–	ResNet-152	–	Euclidean	60,241,986	92.84	Rotation, shift, flip	1024	6	9375.829	0.878
–	MobileNetV3s	–	Pearson	936,242	0.12	–	16	26	39.387	0.767
–	MobileNetV3s	–	Pearson	936,242	0.36	Rotation, shift, flip	16	26	124.977	0.78
–	MobileNetV3s	–	Cosine	1,074,722	0.12	–	256	6	70.463	0.763
–	MobileNetV3s	–	Cosine	1,074,722	0.48	Rotation, shift, flip	256	6	155.607	0.765
ResNet-50	**MobileNetV3s**	✓	**Pearson**	**936,242**	**0.12**	**–**	**16**	**166**	**38.911**	**0.782**
ResNet-152	MobileNetV3s	✓	Cosine	1,074,722	0.12	–	256	86	65.027	0.753

The FADMAKA algorithm on the MobileNetV3s base model achieved the best accuracy of 0.78 using the Pearson metric. Meanwhile, the accuracy without augmented latent fusion was 0.767, the difference is not very significant. Meanwhile, on the teacher model with ResNet-50 base model, the highest accuracy without augmented latent fusion was 0.772 and 0.92 with augmented latent fusion [6]. For the ResNet-152, it was 0.789 without augmented latent fusion and 0.878 with it [6]. The ResNet-based model significantly improved the accuracy when implemented with augmented latent fusion, while the accuracy of the MobileNetV3-based model did not significantly increase. This could be due to the ResNet50 and ResNet152, which have deeper layer, capture more complex and abstract features [25]. In contrast, MobileNetV3s architecture, which prioritizes efficient inference on mobile devices over accuracy due to its simplicity and fewer layers. Among the various scenarios tested, the highest accuracy using the base model MobileNetV3 Small was achieved in the FADMAKA-KD scenario, which utilized a ResNet-50 teacher model and a latent dimension of 16. This configuration resulted in an accuracy of 0.782 and a significantly faster inference time of just 38.911 milliseconds, outperforming other approaches.

C. Comparison with SoftMax Baseline and SOTA Models

We also reimplemented MobileNetV3s base model trained using softmax classifier approach as shown in Table 5. We explored four scenarios: fine-tuning and non-fine-tuning conditions, evaluating both the AdamW and SGD optimizers.

TABLE V
UNSEEN DATA ACCURACY OF BASELINE AND SOTA MODELS VS FADMAKA-KD INFERENCE ON RASPBERRY PI 4B

Method	Image augmentation during training	Total parameters	GFLOPs	Optimizer	Inf. time (ms)	Acc
MobileNetV3s	Rotation, flip, blur	930,470	0.12	AdamW	30.655	0.643
MobileNetV3s	Rotation, flip, blur	930,470	0.12	SGD	31.785	0.422
MobileNetV3s with fine tuning	Rotation, flip, blur	930,470	0.12	AdamW	36.224	0.739
MobileNetV3s with fine tuning	Rotation, flip, blur	930,470	0.12	SGD	37.695	0.602
FADMAKA ResNet-50 (teacher model) [6]	Rotation, flip, blur	25,573,050	33.04	AdamW	2312.472	0.920
FADMAKA ResNet-152 [6]	Rotation, flip, blur	60,241,986	92.84	AdamW	9375.829	0.878
RepVGG_ECA [7]	Contrast, saturation, blur, flip	86,476,990	41.04	Adam	3895.03	0.716
VGG16 [8]	Shear, contrast, skew, flip, rotation	14,717,766	30.80	Adam	2553.405	0.804
ResNet-50 [9]	Shear, zoom, shift, flip, rotation	27,796,358	8.27	Adam	1127.673	0.811
Simple CNN [8]	Shear, contrast, skew, flip, rotation	275,406	0.22	Adam	155.676	0.722
PaddyNet [10]	Rotation, shift, flip, shear	255,686	0.90	Adam	297.328	0.334
Jaya [11]	-	806	1.752e-06	Adam	3.455	0.51
ANN [12]	-	675,906	0.0134	SGD	8.642	0.313
Ours (student model)	Rotation, flip, blur	936,242	**0.12**	AdamW	**38.911**	**0.782**

The highest accuracy for these scenarios is 0.739 with AdamW fine-tuning, with a computation time of approximately 36.224 ms. In contrast, FADMAKA-KD derived from ResNet-50 teacher and MobileNetV3s student scenario, achieved a higher accuracy of 0.782 with a computation time of approximately 38.911 ms. Furthermore, as shown in Table 4, the classification accuracy of the DML and k-NN models, both utilizing the MobileNetV3 small base model, exceeds the results achieved by the standard SoftMax classifier shown in Table 5. Specifically, the DML and k-NN models achieved a peak accuracy of 0.767, which further improved to 0.78 when used augmented latent fusion. However, it's worth noting that the computational cost of k-NN-based classification is higher due to the added time required for nearest neighbor computations across all training data points [26]. This differs from the SoftMax classification approach, where predictions are directly obtained from the neural network's output. In DML, the output takes the form of latent representations [27]. Despite this computational trade-off, the DML and k-NN classifiers consistently outperform the standard SoftMax classifier. Our findings align with earlier research where the DML model with k-NN

classification outperformed the SoftMax classifiers, and the accuracy further improved with augmented inference [6]. Additionally, DML and k-NN models based on augmented latent fusion (FADMAKA) require longer computation times compared to vanilla DML and k-NN models. This is due to the need to extract latent features from each augmented image and then calculate their averages before classification with k-NN. The FADMAKA-KD method improves computational efficiency by learning the latent features from the fusion of several augmented image processes using only a single input image. This approach reduces computation time significantly, by up to 68.9%, while maintaining similar accuracy. Meanwhile, the process of knowledge distillation from the teacher model to the student model can reduce computation time by up to 98.3%. However, this comes with a decrease in accuracy of 15%. Despite this reduction in accuracy, this approach is more effective and efficient compared to the SoftMax classifier method with the same model complexity.

We also compared FADMAKA-KD's performance with SOTA methods, which were reimplemented using our dataset as shown in Table 5. These SOTA models, which focus on pests or disease classification in paddy research, were also retested for inference times on a Raspberry Pi to assess competitiveness. Most complex base models, such as VGG16, ResNet, and RepVGG, trained with SoftMax classification achieved high accuracy but also high inference times. Simpler models such as Simple CNN, PaddyNet, neural network without CNN layer such as Jaya and BP ANN, despite following the hyperparameters mentioned in their respective papers, yielded faster inference time but lower accuracy than our model.

During our computational testing, we explored an approach that combines classical image processing techniques, such as Jaya [11] and ANN [12]. Our focus was on GFLOPs and inference time. We did not specifically test preprocessing time, if data had already undergone preprocessing, similar to the other methods. One notable insight from Table 5 is that the number of GFLOPs doesn't always have a direct correlation with the number of parameters. In other words, having a high number of parameters does not necessarily result in higher GFLOPs. This discrepancy arises because they are not strictly related since GFLOPs measure computational cost meanwhile parameters are the size of the model [28]. For instance, SimpleCNN [8] and PaddyNet [10] have fewer parameters than our model (FADMAKA-KD), yet they yield higher GFLOPs. This difference can be attributed to variations in hyperparameter configurations within the CNN layers, such as kernel size and network design efficiency [29]. Our base network is built upon MobileNet, which incorporates depthwise separable convolutions. This innovation reduces the number of GFLOPs while supporting or slightly increasing the number of parameters [30]. In this research, we address a critical gap in the field of machine learning by focusing on the deployment of our best model on edge devices. This approach is particularly significant given the current trend where many studies conclude with theoretical papers without practical implementation. Our work bridges the divide between research and real-world application, demonstrating that advanced AI technologies can be made accessible and practical in resource-limited environments [31]. Our proposed research leverages computational efficiency and effective base model design to achieve remarkable improvements. We improved the image classification method, achieving accuracy surpassing that of fine-tuned SoftMax classifiers in deep learning, all while supporting similar GFLOPs and computation time with remarkable 5.82% improvement. Importantly, we address the internet connectivity issues met in previous studies that relied on online cloud computing for deployment. By enabling offline functionality, our approach helps on-site processing, which is particularly helpful for applications such as the early detection of pests and diseases in agricultural settings.

IV. Conclusion

We introduce FADMAKA-KD to find paddy pests and diseases, which leverages DML with augmented latent fusion and response-based knowledge distillation. Our approach involves training the student model with soft latent labels that are derived from four distinct augmented images from the teacher model. The student model demonstrated superior accuracy, achieving 0.973 on the validation set and 0.782 on unseen data, outperforming both SoftMax classification at 0.739 and DML with k-NN classification at 0.78 using the same base model. Unlike SoftMax classification, metric learning focuses on developing meaningful embeddings, which can be more effective for fine-grained visual tasks. Additionally, the augmented latent fusion technique exposes the model to diverse data representations during training, enhancing its ability to generalize to unseen data. Inference times were also significantly reduced by up to 98.3%, from 2312.472 ms to 38.911 ms, compared to the teacher model. Our method enhances computational efficiency by learning latent features from a fusion of several augmented image processes using just a single input image. This makes it highly suitable for deployment on edge devices with limited computational resources and memory. We successfully deployed the best-performing model on a Raspberry Pi 4B edge device, bridging the gap between theoretical research and real-world application. This addresses a critical need in the field, where many studies conclude without practical implementation. Future studies could experiment with lightweight architectures for the student model and apply quantization and pruning techniques to reduce model size. Additionally, exploring multi-teacher distillation and continuous learning could enhance performance. Developing methods for on-device fine-tuning would also enable better adaptation to local conditions.

Nomenclature

$d(\cdot)$: distance metric function
$s(\cdot)$: student model
$t(\cdot)$: teacher model
W, b: student's weights and biases
W', b': teacher's weights and biases
z: latent representation
x: image input
y: image's label
$\mathcal{L}_D$: distillation loss
$\mathcal{L}_X$: cross-entropy loss

ACKNOWLEDGMENT

This research received funding from the Directorate General of Vocational Education under the 2024 master's thesis research scheme.

REFERENCES

[1] L.-D. Quach, Q. K. Nguyen, Q. A. Nguyen, and L. T. T. Lan, "Rice pest dataset supports the construction of smart farming systems," *Data in Brief*, vol. 52. p. 110046, 2024.

[2] L. Listihani, P. E. P. Ariati, I G. A. D. Yuniti, and Dewa G. W. Selangga, "The brown planthopper (*Nilaparvata lugens*) attack and its genetic diversity on rice in Bali, Indonesia," *Biodiversitas Journal of Biological Diversity*. 2022.

[3] S. Jesie and Dr. M. S. G. Premi, "A Review on Machine Learning to Detect and Classify Paddy Leaf Disease," *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*. pp. 1822–1828, 2023.

[4] M. Shoaib, B. Shah, S. EI-Sappagh, A. Ali, A. Ullah, F. Alenezi, T. Gechev, T. Hussain, and F. Ali, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers in Plant Science*, vol. 14. Frontiers, p. 1158933, 2023.

[5] S. Kurzadkar, A. Meshram, A. Barve, K. Dhargave, M. Alone, and V. Bhongale, "Plant Leaves Disease Detection System Using Machine Learning," *International Journal of Computer Science and Mobile Computing*. 2022.

[6] H. Darmawan, M. Yuliana, and Moch. Z. S. Hadi, "Cloud-based Paddy Plant Pest and Disease Identification using Enhanced Deep Metric Learning and k-NN Classification with Augmented Latent Fusion," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 6. pp. 158–170, 2023.

[7] H. Ni, Z. Shi, S. Karungaru, S. Lv, X. Li, X. Wang, and J. Zhang, "Classification of Typical Pests and Diseases of Rice Based on the ECA Attention Mechanism," *Agriculture*, vol. 13, no. 5. 2023.

[8] C. R. Rahman, P. S. Arko, M. E. Ali, M. A. Iqbal Khan, S. H. Apon, F. Nowrin, and A. Wasif, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosystems Engineering*, vol. 194. pp. 112–120, 2020.

[9] V. Malathi and M. P. Gopinath, "Classification of pest detection in paddy crop based on transfer learning approach," *Acta Agriculturae Scandinavica, Section B — Soil & Plant Science*, vol. 71, no. 7. Taylor & Francis, pp. 552–559, 2021.

[10] P. A, M. D, and B. K. S, "PaddyNet: An Improved Deep Convolutional Neural Network for Automated Disease Identification on Visual Paddy Leaf Images," *International Journal of Advanced Computer Science and Applications*. 2023.

[11] S. Ramesh and D. Vydeki, "Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm," *Information Processing in Agriculture*, vol. 7. pp. 249–260, 2020.

[12] Y. Lu, Z. Li, X. Zhao, S. Lv, X. Wang, K. Wang, and H. Ni, "Recognition of Rice Sheath Blight Based on a Backpropagation Neural Network," *Electronics*, vol. 10, no. 23. 2021.

[13] B. Ciapas and P. Treigys, "Self-Checkout Product Class Verification using Center Loss approach," *Computer Science Research Notes*. 2023.

[14] H. Darmawan, M. Yuliana, and Moch. Z. Samsono Hadi, "Realtime Weather Prediction System Using GRU with Daily Surface Observation Data from IoT Sensors," *2022 International Electronics Symposium (IES)*, pp. 221–226, 2022,

[15] A. Alkhulaifi, F. Alsahli, and I. Ahmad, "Knowledge Distillation in Deep Learning and Its Applications," *PeerJ Comput Sci*, vol. 7. p. e474, 2021.

[16] Y. Ma, Q. Hua, Z. Wen, R. Zhang, Y. Zhang, and H. Li, "k Nearest Neighbor Similarity Join Algorithm on High-Dimensional Data Using Novel Partitioning Strategy," *Security and Communication Networks*, vol. 2022, no. 1. p. 1249393, 2022.

[17] F. Dang, D. Chen, Y. Lu, and Z. Li, "YOLOWeeds: A novel benchmark of YOLO object detectors for multiclass weed detection in cotton production systems," *Computers and Electronics in Agriculture*, vol. 205. p. 107655, 2023.

[18] L. Zhao and L. Wang, "A new lightweight network based on MobileNetV3," *KSII Trans. Internet Inf. Syst.*, vol. 16. pp. 1–15, 2022.

[19] D. Shi, M. Orouskhani, and Y. Orouskhani, "A conditional Triplet loss for few-shot learning and its application to image cosegmentation," *Neural networks: the official journal of the International Neural Network Society*, vol. 137. pp. 54–62, 2021.

[20] H. Darmawan, M. Yuliana, and Moch. Z. S. Hadi, "GRU and XGBoost Performance with Hyperparameter Tuning Using GridSearchCV and Bayesian Optimization on an IoT-Based Weather Prediction System," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 13, no. 3. INSIGHT - Indonesian Society for Knowledge and Human Development, pp. 851–862, 2023.

[21] M. D. S. Antariksa, A. Y. Husodo, R. B. Huwae, and R. A. Nugraha, "Design and Development of Smart Farming System for Monitoring and Bird Pest Control Based on Raspberry Pi 4 with Implementation of YOLOv5 Algorithm," *2023 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS)*. pp. 1–6, 2023.

[22] N. James, L.-Y. Ong, and M. Leow, "Exploring Distributed Deep Learning Inference Using Raspberry Pi Spark Cluster," *Future internet*, vol. 14. p. 220, 2022.

[23] E. Prasetyo, R. Purbaningtyas, and R. D. Adityo, "Cosine K-Nearest Neighbor in Milkfish Eye Classification," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 3. Infonomics Society, Surabaya, Indonesia, p. 2020, 2020.

[24] P. Ramachandran, T. Eswarlal, M. Lehman, and Z. Colbert, "Assessment of Optimizers and their Performance in Autosegmenting Lung Tumors," *Journal of Medical Physics*, vol. 48, no. 2. pp. 129–135, Apr. 2023.

[25] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, "3D Deep Learning on Medical Images: A Review," *Sensors*, vol. 20, no. 18. MDPI, p. 5097, 2020.

[26] E. Boateng, J. Otoo, and D. Abaye, "Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review," *Journal of Data Analysis and Information Processing*, vol. 8, pp. 341–357, 2020.

[27] N. Mahony, S. Campbell, A. Carvalho, L. Krpalkova, G. Velasco-Hernández, D. Riordan, and J. Walsh, "Understanding and Exploiting Dependent Variables with Deep Metric Learning," in *Proceedings of the International Conference on Intelligent Systems and Computing (ISC) 2020*, pp. 97–113, 2020.

[28] G. Guo and Z. Zhang, "Road damage detection algorithm for improved YOLOv5," *Scientific Reports*, vol. 12, no. 1, p. 15523, 2022.

[29] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling Up Your Kernels to 31×31: Revisiting Large Kernel Design in CNNs," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11953–11965, New Orleans, LA, USA, 2022.

[30] L. Huang, Z. Xiang, J. Yun, Y. Sun, Y. Liu, D. Jiang, H. Ma, and H. Yu, "Target Detection Based on Two-Stream Convolution Neural Network with Self-Powered Sensors Information," *IEEE Sensors Journal*, vol. 23, pp. 20681–20690, 2023.

[31] W. Bismi, D. Riana, and A. S. Hewiz, "Disease Identification on Fig Leaf Images Using Deep Learning Method," International Journal of Advanced Science Computing and Engineering, vol. 6, no. 2, pp. 57–63, 2024.