1. Darunee WATNAKORNBUNCHA[1], 2. Noppadol AM-DEE[2], 3. Adisak SANGSONGFA[3]

Doctoral Program in Industrial Technology Management, Muban Chom Bueng Rajabhat University, Thailand (1) (2) (3)
ORCID: 1. 0009-0006-4616-5873; 2. 0000-0002-2413-1430; 3. 0000-0002-6231-4000

doi:10.15199/48.2024.03.36

Adaptive Deep Learning with Optimization Hybrid Convolutional Neural Network and Recurrent Neural Network for Prediction Lemon Fruit Ripeness

***Abstract**. Lemon is a valuable fruit in the citrus family; optimal usage requires careful selection. The study categorized lemon suitability prediction int 4 classes based on image data. A hybrid neural network, combining Convolutional and Recurrent Neural Networks, was optimized with the Particle Swarm Optimization algorithm. Experimental results were compared to using Convolutional Neural Network alone. The prediction yielded 89.83% training accuracy and 66.06% testing accuracy. However, combining the results increased training accuracy to 91.58% and testing accuracy to 86.76%.*

***Streszczenie.** Cytryna to owoc należący do bardzo pożytecznej rodziny cytrusów, ale aby można było z niej korzystać w celu maksymalizacji korzyści płynących z cytryny, konieczne jest wybranie zachowania przydatności do spożycia. Dlatego w tym badaniu przewidywanie przydatności cytryny jest podzielone na 4 klasy przy użyciu obrazów jako danych do badań. Wyniki predykcji w badaniach z wykorzystaniem połączonej sieci neuronowej pomiędzy Convulotinal Nerual Network i Recurrent Nerual Network z optymalizacją parametrów algorytmem Particle Swarm Optimization, wyniki eksperymentalne porównano z wykorzystaniem wyłącznie Convulotinal Nerual Network. Dla predykcji wynik treningu to 89,83%, a wynik testu to 66,06%, natomiast wynik kombinacji wyników treningu to 91,58% i wynik testu to 86,76%. (**Adaptacyjne głębokie uczenie się z optymalizacją Hybrydowa splotowa sieć neuronowa i rekurencyjna sieć neuronowa do przewidywania dojrzałości owoców cytryny**)*

Keywords: Lemon, Convolutional Neural Network, Recurrent Neural Network
Słowa kluczowe: Cytryna, Konwolucyjna sieć neuronowa, Rekurencyjna sieć neuronowa

1. Introduction

Neural networks are data mining techniques that use mathematical models for information with connectionist computation to simulate the functioning of neural networks in the human brain. The ability to learn pattern recognition and the ability to create new knowledge is similar to that of the human brain [1]. Most researchers are widely used and agree that the neural network has a different structure from the structure of the human brain. But it is also similar to the human brain in that a neural network is a parallel clustering of multiple subprocesses, and this clustering is an essential part of the network's intelligence. While the human brain is larger than a neural network and neurons are more complex, important brain functions such as learning can also be easily modeled with this neural network as well.

However, building a single-learned neural network in very complex learning may not be enough to do the job. Recognize and understand the input data, which may use supervised or unsupervised learning techniques so that the model can classify and process the data itself without defining data properties for each neural model set. The higher the number of layers, the better the ability to classify and analyze the data and learn the meaning of the relationships between the data. This type of neural network model is called a "deep learning neural network [2]". Deep learning can be categorized into several types depending on how researchers apply it, such as Feedforward neural networks, Recurrent neural networks (RNNs), Long short term memory neural networks (LSTMs), Convolutional neural networks (CNNs), etc. In general, deep learning, which is widely used by researchers in research involving predictive image analysis or image classification, is the most effective use of neural networks of CNNs. Because this type of neural network was originally developed for extracting features from data that is not very organized or unstructured data, such as images. The example research of CNNs is used for the prediction or classification of images as follows: Harmandeep Singh Gill and et. al. [3] presented research using fruit classification using deep learning CNNLSTM-RNN for classifying fruits based on optimal attributes such as shape, size, color, texture, and texture. In this regard, the researcher used CNN-LSTM-RNN to control the quality of fruit with many factors affecting fruit quality. For example, the deterioration of the highest quality due to bad weather and environmental conditions, etc. The results showed that the quality can be classified at an average confidence of 96.08% by comparing it with other methods such as SVM, ANFIS, and FFNN. Mehenag Khatun and et. al. [4] presented their research on automatic fruit classification as an interesting application of computer vision. The researchers mentioned that traditional fruit classification methods often rely on manual operations based on visual acuity. And such methods are tedious, time-consuming, and inconsistent. Therefore, this paper used a CNN-based image processing technique to classify fruit, which could classify up to 7 categories at an average accuracy of 98.74% from training in 10 epochs. Rupali Pathak [5] demonstrated the use of CNNs for fruit spoilage discrimination of 10,901 images of 3 fruit species using data from kaggle.com. The initial parameters of the CNNs were set and the efficiency of classification accuracy was tested. The data type was compared to the performance of data classification between CNNs defined by researchers and ready-made models called transfer learning models. The result was higher than the accuracy of 98.23%. Mimma [6] presents research using the classification of 30 fruit species using the data contained in the first FIDS-30 dataset of 971 images which is publicly available and developed into a real-time detection application using a mobile phone camera. Artificial neural networks of YOLOv7 type CNNs, which can detect motion in real-time and combine other techniques for comparison, such as those of VGG16, ResNet50, etc. Accuracy of 96.1% which is greater than ResNet50's 89.16%. Similarly, some researchers use other types of neural networks such as RNNS, and LSTM for data clustering, data classification, and data prediction. Well, examples of the mentioned research can be presented as follows. Kathiresan Shankar [7] presents a research paper that uses a neural network of type RNNs to classify fruit data by automatically adjusting parameters using a total of 75 fruits and 2633 images of all fruits. The average accuracy was 97.78%. In this research [8], the researcher develops a hybrid deep learning-based fruit image classification approach, which uses a convolution autoencoder to pre-train the images and uses an attention-based DenseNet to extract the features of an image. The researcher used two datasets of 26 fruit species to use all fruit images. 124,212 images divided into 85,260 images used for learning, and 38,952 images used for testing, an average accuracy of 95.86%. Chiagoziem C. Ukwuoma [9] proposed using faster region-based convolutional neural networks (FastRCNN) as the most widely used architecture in modifying pre-trained models for sensing tasks. It includes a separate feature model and a regional offering model. Most researchers find this model easier to use by changing the features individually according to their tastes. Therefore, object detection is followed. This research noticed that most researchers used ResNet or VGG as feature extractors while deploying modified Faster-RCNN and Masked CNNs. The results of the research showed that the average accuracy was 95.00%, some were up to 99.00% in some fruits.

From the research mentioned above, most of the fruits used are fresh fruits, but there is another fruit that is interesting and is an economic fruit. A growing trend in the food industry is that fruit is a lemon. The reason is that ”Lemon” is a yellow lemon full of benefits Water and lemon pulp is suitable for diabetics. high in vitamin C Help reduce blood sugar levels, high in fiber, which lemon is different from Lime, and green lemon. But the benefits are not the same contains antioxidants that strengthen the immune system well [10]. Because Lemon has many benefits from flowers, oil from lemon peel, and many others, most of which are beneficial to health is important, so in the world, there is value in production and export each year worth. For example, ”Production of lemon oil is dominated by Argentina, Spain, Italy, the USA, and South Africa. Argentina and Spain dominate world production, with approximately 70% of global output. The USA, Turkey, and Italy are also important lemons. growers. These six countries combined account for approximately 84% of world lemon production. ” etc. The value of the export market is quite high, for example, Argentina in 2017 estimated that the value of Argentina’s lemon industry was \$1.178 million, more than \$1 billion [11]. Lamon has many benefits, both directly and indirectly, by being processed into many medical and industrial products. Therefore, many research studies on the benefits of lemon have come. that can cause added value, for example, Yueh-Hao Ronny Hung [12] presented a research paper where this study aimed to investigate how the antimicrobial activity of lemon essential oil (EOs) benefits the shelf life of peeled shrimp. Eight essential oils were selected for their inhibitory effect on shrimp putrefactive bacteria for comparison. The results showed that lemon oil slowed the deterioration of peeled shrimp during storage. Overall, the study supports the potential use of citrus EOs in microbial control of peeled shrimp to help extend shelf life. Cheng-Yu Hsieh [13] presents research on the use of lemon to compare the antioxidant capacity of unpasteurized lemons and sterilized lemon products. This research indicates that Lactic acid fermented lemon juice can greatly increase bioactive compounds and antioxidant capacity. From research examples of the utilization of lemons, it was found that lemons can be used for a variety of purposes, but to maximize the benefits of lemons, they must come from lemons that are suitable for harvesting. Different harvesting periods for lemons have different uses, so harvesting lemons for optimal use requires expert intervention. The lemon grower’s expertise is difficult and requires a long learning curve if the grower is to master it. Similarly, using a system of intelligence was Invented to help as a tool to predict the fit of the harvest to be the most appropriate, thus allowing farmers to harvest with quality.

So, to solve the problem of the shortage of experts in lemon harvesting that produces the best lemons and from the researcher’s study of the previous research above, it was found that using the best export predictor tool from image analysis, most researchers prefer to use parameterized deep learning to fit the input values. Therefore, in this paper, parameterized deep learning is used in this research so that the prediction result is the best value with the maximum accuracy while the error is the minimum. But the model images for the learning of each class of lemons are similar, it is difficult and complex to learn that using deep learning alone may make predictions. The result for output may not be the best, and Harmandeep Singh Gill’s [3] study found that combining more than one type of deep learning results in better output than using only one type of deep learning for this research to have the best output value, a combination of powerful image analysis advantages of deep learning was used. Convolutional neural networks and iterative deep learning provide a deeper understanding of the recurrent neural network, which maximizes learning and testing efficiency for prediction. However, another important problem is the use of tools to optimize the deep learning parameters used in this research and the study of the tools used to optimize them. According to widely used research, [14], [15], [16], [17] and [18], it is found that swarm intelligence is very popular and effective in performance. Therefore, the researchers have the inspiration to bring swarm intelligence which is particle swarm optimization algorithm (PSO) to help. Optimizing the parameters to obtain the best output for this research will result in greater accuracy and less error in the application.

2. Related work

A. Convolutional Neural Networks (CNNs)

In this section, we will explain in detail and get to know each other. Convolutional Neural Network (CNN), which is a special neural network structure, has a much better ability to classify image data than conventional neural networks. Remove parts of the image, such as the borders of objects. So that the model can learn image characteristics efficiently and accurately. In CNN, convolution layers are used to combine with other types of layers, such as Pooling layers, and then bring those layer groups together, which may change. some hyperparameters, such as the size of the filter layer (which is part of the convolution layer) and the number of channels of the layer. let’s assemble this It is called the structure of the CNN, which has many types such as LeNet, AlexNet, VGG, ResNet, Inception Network, etc. First of all, one should understand the components of CNN, which is the basis to be assembled, starting with the Convolution layer first [19].

1) Convolution layer: In this process, the input matrix $a^{[l-1]}$, while the filter matrix is the matrix of the weight $W^{[l]}$, which has values to be learned from the backward propagation process. The result of convolving these 2 matrices together is the result matrix, $W^{[l]}a^{[l-1]}$. The system then adds intercept $b^{[l]}$ (bias) to form the resultant straight-line equation $Z^{[l]} = W^{[l]}\ a^{[l-1]} + b^{[l]}$, and then feeds this $Z^{[l]}$ into the activation function $g(Z^{[l]})$, resulting in the activation output of that layer, which is $A^{[l]}$.

2) Padding: Most convolutions make the output matrix smaller in size, which means that if multiple convolutions are performed, the final image will be much smaller. Convolutions also tend to cause information along the edges of the image to be ignored, to be fully computed as

the data in the middle of the image because the filter has less chance of capturing information along the edge of the image than in the middle of the image. For example, if the Input is 6×6, padding p=1 will make Input 8×8, etc. The researchers can calculate what the output dimension will be when padding is done, but first, let's compare it to the normal.

2.1) "Valid" convolution, without padding.

The output dimension is:

(1) $$(n - f + 1), (n - f + 1)$$

Where *n* is the Input dimension and *f* is the Filter dimension.

2.2) The "Same" convolution is padding so that the output dimension is equal to the input dimension.

(2) $$(\boldsymbol{n + 2p - f + 1}), (\boldsymbol{n + 2p - f + 1})$$

(3) $$p = \frac{f-1}{2}$$

Where *p* is the pixel size of the padding on each side. Incidentally, *f* should be an odd number, so that achieve symmetry and have a center point.

2.3) Stride: Stride means the number of channels to be shifted in each convolution, e.g. if moving one channel Stride *s* = 1, but moving 2 channels is *s* = 2, etc.

The formula for calculating the output size when using Stride is:

(4) $$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor, \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

2.4) Pooling layer: After the data has passed through the convolution layer, it is often sent to another layer called the pooling layer. The function of the pooling layer is to extract the most important part of the data. The pooling layer mechanism is very simple. It is to extract only the maximum value of the Grid and store it in the Output, for example, from Figure 1, showing *a* 2×2 Pooling layer with Stride *s*=2:

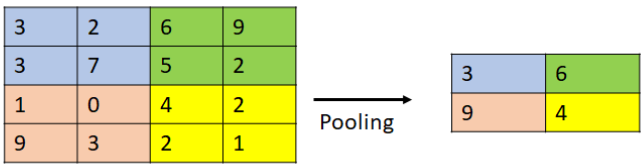

Fig. 1. Pooling layer size = 2×2 with Stride *s*=2

The pooling layer that extracts only the maximum value of the Grid is called Max pooling, which is the most commonly used form. There is also Average pooling, which finds the average of the Grid but is much less used than Max pooling.

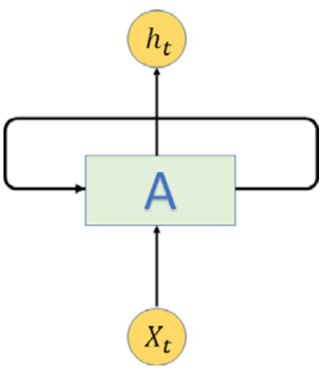

Fig. 2. The image shows the RNN Engine

B. Recurrent Neural Networks (RNNs)

A recurrent Neural Network [20] is a network that combines the output from the previous state as input. Its operation is similar to loop operation. which can show the operation as shown in Figure 2. If when bringing the engine writes in full, it will come out as in Figure 3 as follows, the behavior that appears is similar to that of a normal neural network with many neural connections output to the next new network. Therefore, the equation of RNN can be written as follows.

(5) $h(t) = \sigma(x(t)W_{in} + h(t-1)W + b_n)$

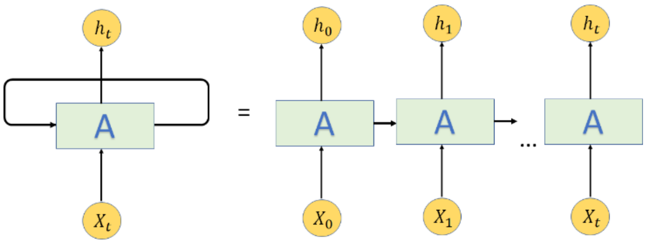

Fig. 3. The image shows the Full RNN Engine

When *σ* is activated function, *bh* is bias value and *W* is weight value. Where this equation represents the use of the output of *x*(*t*) in conjunction with the output of *h*(*t* - 1) (or the output of the previous network) with two weights of *x*(*t*) and *h*(*t* - 1).

However, because RNNs use data from previous networks, they work well with time series data (taking previous time data into the same network). Let's continue with the current time series, which includes text and audio data. But the disadvantage of RNN is that in theory, RNN can solve longterm problems very well if selecting weight well, but in actual use, it doesn't come out as well as expected.

C. Particle Swarm Optimization (PSO)

PSO [21] is a simulation of a flock of birds foraging because it is instinctive that all birds do not know where the food is when looking for food. How far is it from food? As they fly at different stages, they determine their distance from the food source and the entire flock chooses to fly according to the bird closest to the food source. They then proceed to the next phase of flight where the entire flock repeats this process until they reach a food source. From the feeding method of a flock of birds, we can see that in determining the distance of the bird from the food source. And the decision to follow the bird closest to the food source at each stage takes so little time that it looks like they are flying in a continuous group. Thus, in this kind of optimization, each bird in the flock is represented by a particle in which each particle has fitness. The value indicates its distance from the food source, with all particles flying after the particle with the best fitness value in each iteration period. The PSO concept begins with randomly determining the position of the particle (where the positions of those particles are a set of possible solutions) then it optimizes by improving the values in each round of decision, where each particle is updated by changing its position accordingly. The two best values are:

1) The best value of a particle (pbest: particle best) is the best position of its previous motion.

2) The global best (gbest: global best) is the value that best prevails over the movement of the entire group (swarm).

The operation of PSO is a cyclic process (Iteration), in which in each cycle of work, the velocity of each particle is adjusted by three important variables:

1) The particle's current speed (velocity).
2) The information that the particle contains (pbest).
3) Aggregate data of whole groups of particles (gbest).

After that, each particle adjusts its position using the newly calculated velocity. Based on the following equation, the new velocity of the particle is calculated according to Equation 6, and adjust the new position value of the particle according to Equation 7.

$$V_{ik}(t+1) = V_{ik}(t) + [c_1 * r_1 * (p(t) - x_{ik}(t))] + [c_2 * r_2 * (g(t) - x_{ik}(t))] \tag{6}$$

$$X_{ik}(t+1) = X_{ik} + V_{ik}(t+1) \tag{7}$$

where $c1$ and $c2$ are constants it is sometimes called the learning factor (to weight memory). This is usually equal to 2, but other values can be used, ranging between [0; 4] or 0 to 4. From the concept of PSO, the algorithm with pseudo-code can be shown as follows in Figure 4.

```
Step1: Randomly initialize Swarm population of N particles Xi (i=1, 2, …, n)
Step2: Select hyperparameter values
    w, c1 and c2
Step 3: For Iter in range(max_iter):  # loop max_iter times
        For i in range(N):  # for each particle:
          a. Compute new velocity of ith particle
             swarm[i].velocity =
                w*swarm[i].velocity +
                r1*c1*(swarm[i].bestPos - swarm[i].position) +
                r2*c2*( best  pos  swarm - swarm[i].position)
          b. Compute new position of ith particle using its new velocity
             swarm[i].position += swarm[i].velocity
          c. If position is not in range [minx, maxx] then clip it
             if swarm[i].position < minx:
                swarm[i].position = minx
             elif swarm[i].position > maxx:
                swarm[i].position = maxx
          d. Update new best of this particle and new best of Swarm
               if swaInsensitive to scaling of design variables.rm[i].fitness <
swarm[i].bestFitness:
                  swarm[i].bestFitness = swarm[i].fitness
                  swarm[i].bestPos = swarm[i].position

               if swarm[i].fitness < best_fitness_swarm
                  best  fitness  swarm = swarm[i].fitness
                  best  pos  swarm = swarm[i].position
         End-for
      End -for
Step 4: Return best particle of Swarm
```

Fig. 4. The image shows the pseudo-code of PSO Algorithm

3. Methodologt

Figure 5 shows an overview of the whole research framework starting with data collection from lemon fruit. The data were categorized by quality and once the data was obtained, deep learning was developed to be used for prediction. Therefore, in this research, the detailed steps in working according to the framework can be described as follows.

A. Data Preparation

In preparing this lemon data, a total of 1,6414 images of lemons were used, and the images of lemons were divided into 4 classes of images of lemons, with class 1 as Grade A lemons are the most suitable lemon for the extraction of essential oils used in the food industry and cosmetic industries. The extracted lemon juice in this class has an average sweetness of 7◦brix and pH= 2. In class 1 Use 394 images. Class 2 is a lemon whose quality is not much different from class 1. It can also be used in the food and cosmetic industry, but the quality may not be equal to class 1. Essential oils from lemon have lower quality when Compared with class 1, however, the average sweetness was 7◦brix and pH = 2, same as class 1, and 525 images were used in this class 2. Class 4 is a lemon with lower quality than Class 1 and 2 because it is a lemon that has not yet reached harvest time. When separated from the juice, the lemon juice is of inferior quality, with an average sweetness of 7◦brix and pH = 2. In class, this contains a total of 261 images. Class 3 is lemon with the lowest quality of all classes. Because it is an over-harvested lemon, the advantage of lemon in this class is that it will give lemon juice more than all classes, which is suitable for bringing lemon juice to use in various industrial applications. While the average sweetness in this class was 8◦brix and pH = 3, in this class 3 there were 461 images in total. However, this research determined that 70% of all learning lemon images were shared and 30% were used for testing, while class names were assigned for prediction as follows.

- Class 1 is good.
- Class 2 is fair.
- Class 3 is poor quality.
- Class 4 still can not use.

Figure 6 shows the number of images used in each class and Figure 7 shows examples of images used in this research divided into 4 classes according to lemon quality.

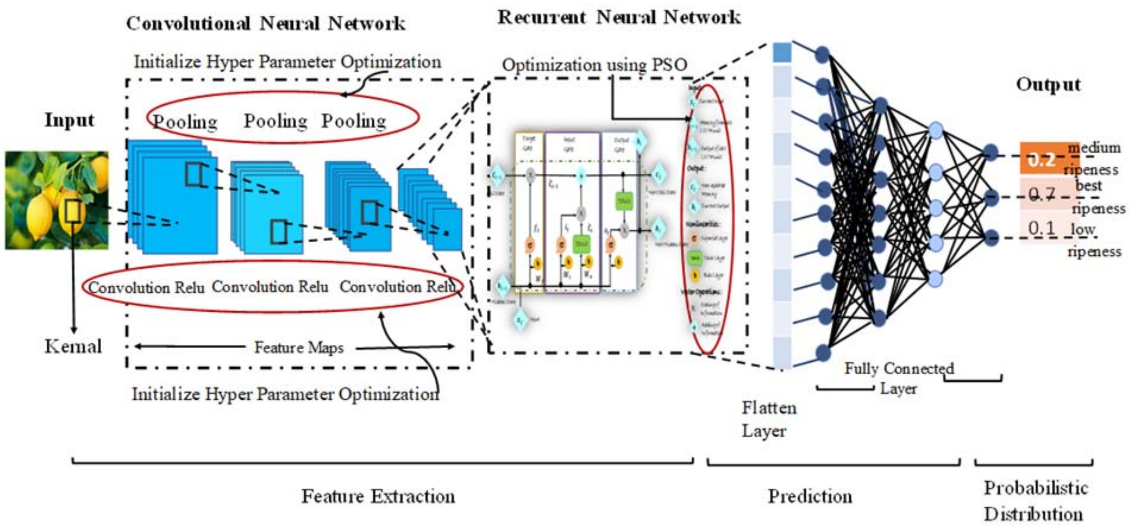

Fig. 5. The image shows the framework of Adaptive Deep Learning with Optimization Hybrid Convolutional Neural Network and Recurrent Neural Network

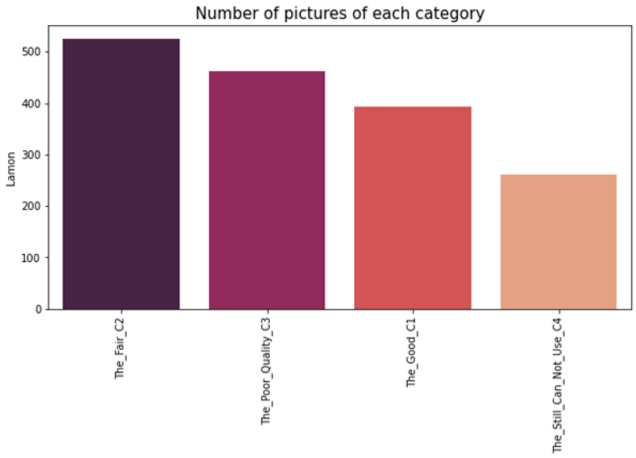

Fig. 6. Graph showing the number of pictures of lemon classified by 4 classes, the total number of pictures is 1,641 pictures

Fig. 7. The pictures show the sample information of Lemon in each class of 4 classes

B. Measuring the efficiency of the Neural Network

To measure the effectiveness of prediction in this research, Confusion Matrix was used. The Matrix, an important tool for estimating the predicted exclusion from the developed model, is based on measuring how the experimental exclusion (using the developed model) applies to observation events. What are the actual proportions and other measurements besides using Confusion Matrix to find the accuracy, these are two measurements that must be performed in parallel. Where both types of measurement formulas can be represented by the formulas below.

Table 1. Table showing the confusionmatrix formulation.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positive (TPs)	False Positives(FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

From table 1, When True Positive (TP)= what is predicted matches what actually happened, in case the prediction is true and what happened is true. True Negative (TN)= The prediction matches what happened in the case of the prediction not being true and what happened is not true. False Positive (FP)= Prediction does not match what happened, i.e. Prediction is true, but what happens is not true. False Negative (FN)= Prediction does not match what happened. is a prediction that is not true But what happens is that it is true that TP, TN, FP, and FN in the table are represented by frequency values. In this research, the Confusion Matrix can be used to calculate the evaluation of the prediction efficiency with the developed Accuracy model. (The accuracy that we guess matches what happened), which has the following formula in equation 8:

(8) $$Accuracy = \frac{(TP_s+TN_s)}{(TP_s+TN_s+FP_s+FN_s)}$$

At the same time, we can calculate the prediction accuracy percentage, which is the ratio of the number of prediction accuracy to the total number of predictions. The equation is as follows in equation 9:

(9) $$prediction\ accuracy = \frac{correct\ prediction}{total\ predictions} \times 100$$

```
Model: "sequential_15"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_90 (Conv2D)          (None, 400, 300, 32)      896

 max_pooling2d_90 (MaxPoolin  (None, 200, 150, 32)     0
 g2D)

 conv2d_91 (Conv2D)          (None, 200, 150, 64)      18496

 max_pooling2d_91 (MaxPoolin  (None, 100, 75, 64)      0
 g2D)

 conv2d_92 (Conv2D)          (None, 100, 75, 64)       36928

 max_pooling2d_92 (MaxPoolin  (None, 50, 37, 64)       0
 g2D)

 conv2d_93 (Conv2D)          (None, 50, 37, 64)        36928

 max_pooling2d_93 (MaxPoolin  (None, 25, 18, 64)       0
 g2D)

 conv2d_94 (Conv2D)          (None, 25, 18, 64)        36928

 max_pooling2d_94 (MaxPoolin  (None, 12, 9, 64)        0
 g2D)

 conv2d_95 (Conv2D)          (None, 12, 9, 64)         36928

 max_pooling2d_95 (MaxPoolin  (None, 6, 4, 64)         0
 g2D)

 flatten_15 (Flatten)        (None, 1536)              0

 dense_30 (Dense)            (None, 256)               393472

 activation_30 (Activation)  (None, 256)               0

 dropout_15 (Dropout)        (None, 256)               0

 dense_31 (Dense)            (None, 4)                 1028

 activation_31 (Activation)  (None, 4)                 0

=================================================================
Total params: 561,604
Trainable params: 561,604
Non-trainable params: 0
_________________________________________________________________
```

Fig. 8. The summarised model of CNN for prediction of Lemon

C. Defining and optimizing hyper-parameters of CNN deep learning

To determine the hyperparameters of the CNNs used in this research, the researcher used the kernel size =3×3, the size of the polling=2×2, padding=same number of neural size 5, activation function=relu,optimizer=adam and batch size=128 a total number of learning cycles 100 epochs where these hyperparameters are obtained from repeated experiments. And many adjustments were found to get the best accuracy for prediction. The results of the overall hyperparameters picture are shown in Figure 8.

D. Hybrid CNN and optimization RNN using PSO for prediction Lemon

Because the image input data of lemons in each class is very close, especially in the 1st and 2nd classes, it is not very good to learn using CNN neural network. The experimental results of using only CNN are shown in the next section. Therefore, the researcher takes advantage of the Recurrent Neural Network (RNN), which is a network that takes the output from the previous state to be used as input. The operation is similar to the operation. It is a loop as shown in Figure 2, using the principle of feeding the model's internal state back as a new input coupled with normal input, called hidden state, internal state, and memory, allowing the model to recognize the pattern of the input sequence. The sequence is better suited to recognition in cases where input data are similar and sequenced. But RNNs have problems with vanishing gradients or exploding gradients, and their performance deteriorates with long sequences. GRUs are designed to address these issues. Gated Recurrent Units (GRUs) are an on and off-state update mechanism within the Recurrent Neural Network. Similar to Long Short-Term Memory (LSTM) with forget gate but with fewer parameters than LSTM due to lack of output gate, GRU performs similarly to LSTM in many tasks but due to fewer parameters makes training easier, faster, and in In some jobs with smaller datasets, GRUs have been found to perform better.

Pseudocode of PSO optimization Recurrent Neural Network (RNN)
Input: RNN Weights from random distribution ,x_trainCNN, y_trainCNN
Output: best Hyper Parameters of RNN
For each particle **do**
Initialize particle **(Initialize Particle)**
End for
For each particle **do**
Calculate fitness value (Accuracy) **Equations 8, 9**
If the fitness value is better than the best fitness value (pBest) in history
set current value as the new pBest
End if
Choose the particle with the best fitness value of all the particles as the gBest
For each particle **do**
Calculate particle velocity according to **Equation 6**
Update particle position according to **Equation 7**
End for
RNN Training
Initialize weight matrices hidden connections parameterized by a weight matrix U, hidden-to-hidden recurrent connections parameterized by a weight matrix W, and hidden-to-output connections parameterized by a weight matrix V, and all these weights (U, V, W) from output PSO and bias b, c with zeros
For epochs and batch_size **do**
Forward propagation to compute predictions
$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)}$
$h^{(t)} = \tanh(a^{(t)})$ (hidden state at time t and acts as memory)
$o^{(t)} = c + Vh^{(t)}$ (illustrates the output of the network)
$\hat{y}^{(t)} = softmax(o^{(t)})$ (vector of normalized probability over the output)
Compute the loss
$CE = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$
M is number of possible class labels
y is a binary indicator (0 or 1) of whether class label c is the correct classification for observation o
p is the model's predicted probability that observation
Backpropagation to compute gradients
$W \leftarrow W - \propto \frac{\partial L}{\partial W}$ (The equation updates the weights W to minimize loss)
$\nabla_{h^{(T)}} L = V^T \nabla_{o^{(T)}} L$
$\nabla_{h^{(t)}} L = \left(\frac{\partial h^{(t+1)}}{h^{(t)}}\right)^T (\nabla_{h(t+1)} L) + \left(\frac{\partial o^{(t)}}{\partial h^{(t)}}\right)^T (\nabla_{o^{(t)}} L)$
$= W^T diag\left(1 - (h^{(t+1)})^2\right)(\nabla_{h^{(t+1)}} L) + V^T(\nabla_{o^{(t)}} L)$
Update weights based on gradients
Using back-propagation through time, it calculated the gradient for each parameter of the model
End for
End for

Fig. 9. The Pseudocode of PSO optimization Recurrent Neural Network (RNN)

The researcher will encounter the Vanishing Gradient problem in training Artificial Neural Networks with Gradient Descent and Backpropagation algorithms in Machine Learning. During training, the model will update the Weight and Bias from the Partial Derivative of the Loss Function depending on that Weight, Bias every training cycle. The gradient gets smaller and smaller until it reaches 0, so the Weight is no longer updated. The Exploding Gradient Problem is the opposite of the Vanishing Gradient Problem. The researcher find that during training the gradient gets bigger and bigger until it reaches Infinity or Not a Number (NaN), meaning the number is beyond the system's capacity. This makes the model unable to continue training. The researcher has to reset the model and start training again. The researcher can solve Vanishing Gradient Problem and Exploding Gradient Problem by weighting initialization and batch normalization. In order to optimize these initial values, in this research, the researcher used a particle swarm optimization algorithm to optimize parameters. And to make the optimal lemon prediction output value for each task have the best value, the accuracy is the highest starting from the combination of neural networks with CNN-type neural networks with optimal parameters and RNNtype neural networks with optimal parameters are also used. Therefore, in order to be able to use this model clearly in the work, the researcher will present a pseudocode to work in Figure 9 and Figure 10 as follows.

Pseudocode Hybrid Between CNN and RNN
Input: Image Lemon Dataset
Output: Prediction the best Results
Data: Training Data, epochs, batch size
Initialization hyperparameter
while *epochs ≤epochs(max)* **do**
extract features:
for each epoch **do**
train CNN();
Function Conv (activations, weights)
for (l=0;l<L;l++) do // Loop of output feature maps
for (m=0;m<M;m++) do
for (n=0;n<N;n++) do //Loop of filter number
sum=bias[l];
for (k=0;k<K;k++) do //Loop of filter channel
for (s1=+;s1<S1;s1++) do //Loop of 2D convolution
for (s2=0;s2<S2;s2++) do
sum+=weight[k][l][s1][s2]× input[k][m+s1][n+s2]
end for
end for
end for
output[l][m][n]=activation function(sum)
end for
end for
end for
return (Conv)
where N= Number of 3D kernels /of output feature map channel
K= Number of input feature map/kernel channels
M= Input feature map width/height
L= Output feature map width/height (=1 in FCN layer)
S1= Kernel width/height (=M in FCN layers), S2 = Batch size of 3Dh feature maps
Function Pooling(activation (l^{th} layer))
$h_{xy}^{l} \leftarrow max_{(i=0,1,2,\dots,s, j=0,1,2\dots,s)} h_{(x+i)(y+i)}^{l-1}$
return(h_{xy}^{l})
Function FC(activation, learnable parameters)
$z^{l} \leftarrow W^{l} \cdot h^{l-1}$
return(z^{l})
Pool←ConV, FC←Pool, Output←FC
Calculate average loss over each epoch in minibatch
Backpropagation applied to every iteration
end for // each epoch forCNN
output from CNN :x train, y train, x test, y test, optimization model of CNN
Pseudocode of PSO optimization Recurrent Neural Network (RNN)
output ←(Best Lemon Prediction)
end while

Fig. 10. The Pseudocode of Hybrid between CNN and RNN which is
optimization by PSO

4. Experimental results

The researcher divided the lemon visual data into 70% for learning and 30% for use testing from the framework part of the methodology and the pseudocode part of the same part as discussed earlier. The computer resources are as follows, the researcher used a workstation computer type Lenovo intel core i7 gen 10*th* GPU, speed 2.60 GHz, 32 GB ram, OS Microsoft Windows 11 Pro. The image of the sampled lemon is shown in Figure 7. The results obtained from the experiment start from the results of using a CNN-type artificial neural network to predict the fruit of lemons in each class. The best summary for prediction using a neural network type CNN as shown in Fig. 8 uses a total number of training cycles of 100. Performance measurements use a confusion table. The average training accuracy was 89.83%, and the average test accuracy was 61.00%, according to Figure 11, the average training time was 723 seconds. The prediction testing images using an artificial neural network CNN-type can be shown in Figure 12, and when the various architectures of model CNN were tested to find the best accuracy, it was found that it can be used with DenseNet169 to get good accuracy. The highest at 67.27% (testing accuracy), the results obtained using various architectures can be shown in Figures 13,14,15 and 16 are predicted results using the DenseNet169 architecture respectively.

```
Classification Report:

              precision    recall  f1-score   support

           0       0.63      0.65      0.64        26
           1       0.68      0.57      0.62        30
           2       0.50      0.86      0.63        14
           3       0.67      0.33      0.44        12

    accuracy                           0.61        82
   macro avg       0.62      0.60      0.58        82
weighted avg       0.63      0.61      0.60        82

Confusion Matrix:

[[17  6  1  2]
 [ 2 17 11  0]
 [ 1  1 12  0]
 [ 7  1  0  4]]

Accuracy: 0.60976
```

Fig. 11. The confusion matrix and accuracy of CNN by testing the Lemon dataset

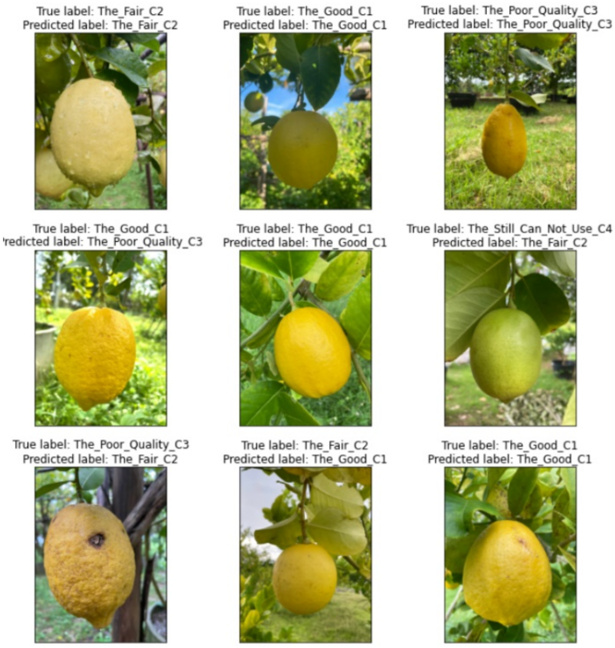

Fig. 12. The image shows the results obtained from the prediction using the artificial neural network CNN, the accuracy is 61.00%

From the experimental results mentioned above, using only CNN neural network with optimized parameters, it was also found that the accuracy of the test was still not very high because the original images used for training were taken in the real situation and without editing the image to make it the most natural image. Therefore, it was found that the colors in each class were very similar, which is the nature of the original lemon color. Therefore, the researcher uses another type of neural network with repetitive learning properties to combine, called the RNN type neural network. The parameters of the RNN neural network were optimized using the PSO algorithm to obtain the optimal parameters corresponding to the parameters of the CNN. The population used for tuning the RNN consisted of a learning rate with a lower bound value of 0.00001 and an upper bound value of 0.01. The weight used in each neuron was the lower bound value of 4 and the upper bound value of not more than 64. A total population of 100 individuals was divided into 5 particles, totaling 20 swarms used in training rounds of 100 epochs and used the number of generations to train in the beginning at 5 generations, the values are not stable when increasing to 10 generations, the values are stable and have a good training value, the training accuracy is 91.58%, the testing value is 86.76%, as shown in Figure 17,18 and 19 respectively.

	model	val_accuracy	accuracy	Training time (sec)
0	DenseNet169	0.7279	0.6606	282.61
1	ResNet101V2	0.5782	0.5818	654.64
2	MobileNetV2	0.5986	0.5758	65.90
3	DenseNet121	0.6327	0.5212	188.37
4	MobileNet	0.6327	0.5091	68.53
5	Xception	0.5918	0.4970	224.16
6	ResNet50V2	0.6122	0.4909	346.73
7	DenseNet201	0.6054	0.4909	346.03
8	InceptionV3	0.5170	0.4848	334.85
9	ResNet152V2	0.6190	0.4788	1055.10
10	VGG16	0.4762	0.4242	1735.77
11	InceptionResNetV2	0.5782	0.4121	609.93
12	VGG19	0.4966	0.3758	1529.70
13	EfficientNetB3	0.3265	0.3333	110.78
14	ResNet50	0.3265	0.3333	385.08
15	ResNet101	0.3265	0.3333	713.02
16	EfficientNetB2	0.3265	0.3333	84.58
17	MobileNetV3Large	0.3333	0.3333	73.54
18	EfficientNetB0	0.3265	0.3333	70.07
19	EfficientNetB1	0.3265	0.3333	81.79
20	EfficientNetB4	0.3265	0.3333	161.72
21	MobileNetV3Small	0.3265	0.3333	61.40
22	EfficientNetB7	0.3197	0.2424	207.48
23	EfficientNetB6	0.3197	0.2424	191.33
24	ResNet152	0.2449	0.2061	1119.11
25	EfficientNetB5	0.2041	0.1939	175.79

Fig. 13. The image shows a summary of the different architectures used in CNNs showing the run time and accuracy of the tests in descending order.

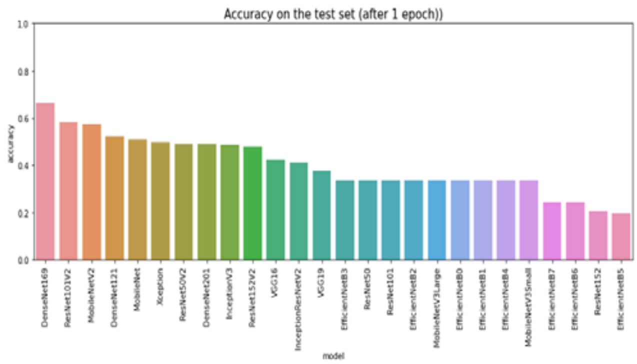

Fig. 14. The graph shows a summary of the accuracy of different architectures when learning 1 epoch

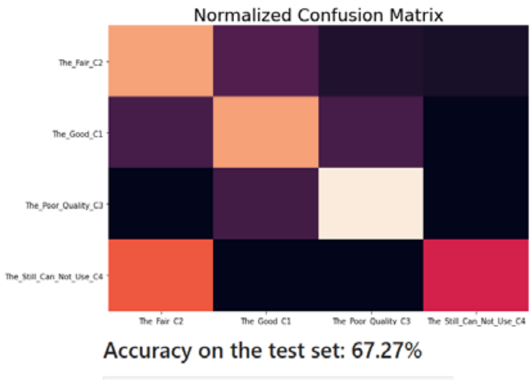

Fig. 15. The confusion matrix and accuracy of CNN-DenseNet169 by testing Lemon dataset

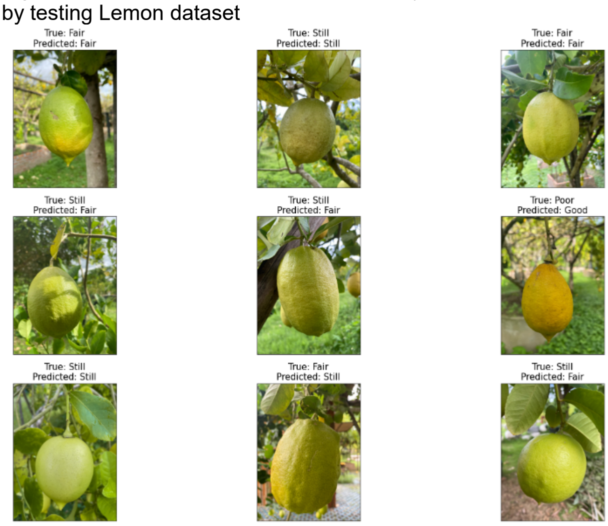

Fig. 16. The image shows the results obtained from the prediction using the CNN-DenseNet169, the accuracy is 67.27%

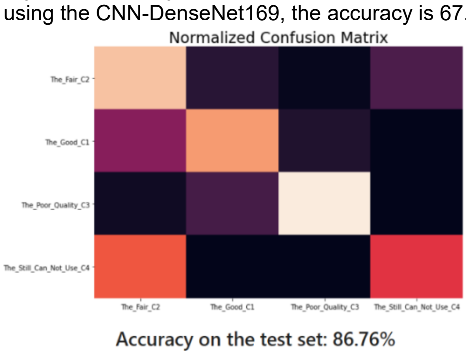

Fig. 17. The confusion matrix and accuracy of hybrid between CNN and RNN by testing Lemon dataset

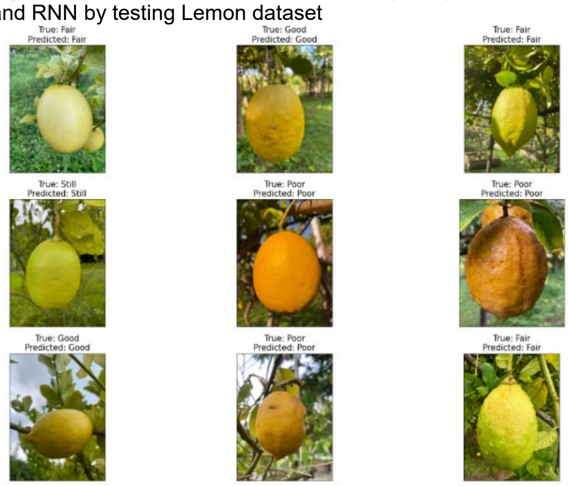

Fig. 18. The image shows the results obtained from the prediction using the hybrid between CNN and RNN, the accuracy is 86.76%

```
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (InputLayer)          [(None, 256, 64, 1)]      0

 conv1 (Conv2D)              (None, 256, 64, 32)       544

 batch_normalization_18 (Bat  (None, 256, 64, 32)      128
 chNormalization)

 activation_18 (Activation)  (None, 256, 64, 32)       0

 max1 (MaxPooling2D)         (None, 128, 32, 32)       0

 conv2 (Conv2D)              (None, 128, 32, 64)       32832

 batch_normalization_19 (Bat  (None, 128, 32, 64)      256
 chNormalization)

 activation_19 (Activation)  (None, 128, 32, 64)       0

 max2 (MaxPooling2D)         (None, 64, 16, 64)        0

 dropout_12 (Dropout)        (None, 64, 16, 64)        0

 conv3 (Conv2D)              (None, 64, 16, 128)       131200

 batch_normalization_20 (Bat  (None, 64, 16, 128)      512
 chNormalization)

 activation_20 (Activation)  (None, 64, 16, 128)       0

 max3 (MaxPooling2D)         (None, 64, 8, 128)        0

 dropout_13 (Dropout)        (None, 64, 8, 128)        0

 reshape (Reshape)           (None, 64, 1024)          0

 dense1 (Dense)              (None, 64, 64)            65600

 RNN1 (Bidirectional)        (None, 64, 512)           657408

 RNN2 (Bidirectional)        (None, 64, 512)           1574912

 dense2 (Dense)              (None, 64, 30)            15390

 softmax (Activation)        (None, 64, 30)            0

=================================================================
Total params: 2,478,782
Trainable params: 2,478,334
Non-trainable params: 448
_________________________________________________________________
```

Fig. 19. The summarised model of the hybrid between CNN and RNN for the prediction of Lemon

However, the researcher has used the results of using a combination of CNN and RNN for predicting the accuracy of lemon and using the accuracy of other studies that also predicted fruit from images. Comparative results show that the prediction using the method used in this research differs from other studies in that it gives better accuracy as shown in Table 2 as follows.

Table 2. Table showing the comparison accuracy between this research and the other research.

Author	Accuracy %
Yuzhen Lu [22]	74.00
Hou et. al. [23]	83.51
Nagnath et al. [24]	81.75
Tamoor Khan et. al. [25]	76.30
Darunee Watnakornbuncha and et. al.	**86.76**

5. Conclusion

From the experimental results mentioned above, it was found that when using a CNN with optimized parameters, the accuracy result was found to be in a not very high range with a learning accuracy of 89.83%, while a test accuracy of 61.00% using the number of learning cycles of 100 epochs. When the optimized parameterized CNNs were used together with different architectures as shown in Fig. 13 and compared, it was found that DenseNet169 architecture could give the highest accuracy among all 25 architectures, 66.06%, but still a value that is not very high that can be accepted. When considering the imported data, which is a picture of all four lemon classes, it is found that the color of each type of image, especially in Class 1 and Class 2, has very little difference almost similar, making the prediction or difficulties in prediction, coupled with the color of the background, and the distance of the original images for learning are different, some of the images were taken at close range, showing the oil glands on the surface of lemons, which differ in each class, with a small number of images. Therefore, to make very limited learning, one can learn better, this research uses a combination (hybrid) of RNN-type neural networks and CNNtype neural networks. The combination (hybrid) of RNNs uses the advantages of RNNs with repetitive learning to better remember the original image.

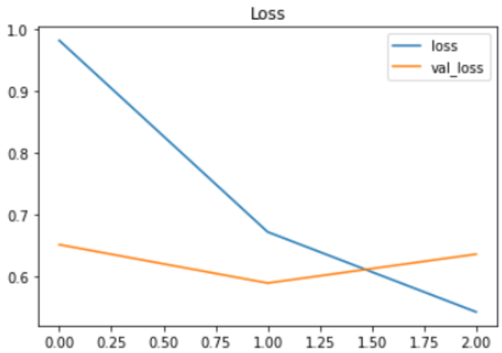

Fig. 20. Traning and Validation Loss of hybrid between CNN and RNN optimized parameter by PSO

Moreover, in optimizing the RNN neural network parameters with weight and learning rate adjustment, the PSO algorithm was used to optimize the values according to the CNN neural network performance. The results of the combined neural network showed that the accuracy was significantly higher than that of the single CNN neural network. It was found that the learning accuracy was 91.58% and the test accuracy was 86.76% as shown in the experimental results shown earlier (Fig. 15, 16, and 17). And a well-optimized and consistently parameterized RNN was able to make the predictions of difficult-to-learn images more accurate according to the confusion matrix shown in the results section (Fig. 17). If the analysis of the cause of the learning accuracy is very high while the test accuracy is not very high from the learning curve graph in Fig. 20, it is found that the loss and value loss graphs occur. The overfitting learning curve shows that the model initially learns too well from the training dataset, including variations in noise or fluctuations in the training dataset, until it learns from the disturbance and details of the data that are not. Correctly, the model would not be suitable for predicting the data, but when the learning curve presented in this paper was adjusted, at some point good fitting occurred, indicating that the system was adjusted. The adaptive for learning to be more efficient, thus making the output of the prediction with more accuracy as well.

6. Recommendation

From the aforementioned research conclusions, it was found that using a combination of the advantages of both types of artificial neural networks (CNN and RNN) was found to increase the accuracy, but when Working for a period of time, found that the learning curve graph is underfitting, it indicates that our model does not work because it cannot capture the trend of the data because our model is not suitable or the data is too small. This model has a high bias (high bias), which means that the model starts to decline further learning at a certain time. In this case, it is necessary to arrange the learning period appropriately. That is, future research may have to optimize the learning time to be the most appropriate by adapting an algorithm to support the optimization as mentioned, which may make the accuracy of prediction higher than that shown in this research.

Acknowledgment

This research was supported by Pasutara Co., Ltd. 317 No.1 Ta Nao Si Subdistrict, Suan Phueng District, Ratchaburi, Thailand. Tel: +66 09-2254-4199
E-mail: pasutara.thailand@gmail.com

Authors: *Darunee Watnakornbuncha, Ph.D. candidate., Muban Chom Bueng Rajabhat University, Ratchaburi, Thailand, E-mail: Daruneewat@mcru.ac.th; Asst. Prof. Dr. Noppadol Am-Dee, Ph.D., Muban Chom Bueng Rajabhat University, Ratchaburi, Thailand, E-mail: noppadolamd@mcru.ac.th; Dr. Adisak Sangsongfa, Ph.D. Muban Chom Bueng Rajabhat University, Ratchaburi, Thailand, E-mail: adisaksang@mcru.ac.th*

REFERENCES

[1] R. Sharma, “A detailed guide on the meaning importance,j and future of neural networks,” Blog of Artificial Intelligence and Machine Learning, (2022), Available on https://emeritus:org/in/learn/ai-ml-neural-networks/
[2] P. Anurak, “What is a deep learning neural network?” Thai Config Company Limited, (2023), Available on https://thaiconfig:com/artificial-intelligence-ai/what-is-deeplearning/
[3] H. S. Gill and B. S. Khehra, “Fruit image classification using deep learning,” *Research Square*, vol. 1, pp. 1–9, (2022), available on https://doi:org/10:21203/rs:3:rs-574901/v1
[4] M. Khatun, J. Nine, P. Sarker, and et.al., “Fruits classification using convolutional neural network,” *Journals- Global Research and Development Journal for Engineering*, (2020), vol. 5, no. 8, pp. 1–5.
[5] R. Pathak and H. Makwana, “Classification of fruits using convolutional neural network and transfer learning models,” *Journal of Management Information and Decision Sciences*, (2021). vol. 24, no. S3, pp. 1–12.
[6] M. Aznin, S. Ahmed, and et.al., “Fruits classification and detection application using deep learning,” *Hindawi Scientific Programming*, vol. 2022, pp. 1–16, (2022), https://doi:org/10:1155/ 2022/4194874
[7] K. Shankar, S. Kumar, and et.al., “An automated hyperparameter tuning recurrent neural network model for fruit classification,” *Mathematics (MDPI)*, (2022), vol. 10, no. 13, pp. 1–18. Available on: https://doi:org/10:3390/math10132358
[8] G. Xue, S. Liu, and Y. Ma, “A hybrid deep learning-based fruit classification using attention model and convolution autoencoder,” *Complex and Intelligent Systems*, (2023), vol. 9, pp. 2209–2219. Available on: https://doi:org/10:1007/s40747-020-00192-x
[9] C. C. Ukwuoma, Q. Zhiguang, and et.al., “Recent advancements in fruit detection and classification using deep learning techniques,” *Hindawi Mathematical Problems in Engineering*, (2022), vol. 2022, pp. 1–29. Available on: https://doi:org/10:1155/2022/9210947
[10] M. Ware, “How can lemons benefit your health?” Medical News Today, (2019), Available on https://www:medicalnewstoday:com/articles/283476
[11] P. Greenhalgh, “Lemon,” IFEAT Socio-Economic Report, (2021), Available on https://ifeat:org/wp-content/ uploads/2021/09/Socio-Economic-Report-on-Lemon-2021:pdf
[12] Y. H. R. Hung, H. J. Lin, and et.al., “Effect of lemon essential oil on the microbial control, physicochemical properties, and aroma profiles of peeled shrimp,” *LWT Food Science and Technology*, (2023), vol. 173, pp. 1–9.
[13] C. Y. Hsieh, S. Hsieh, J. Y. Ciou, and et.al., “Lemon juice bioactivity in vitro increased with lactic acid fermentation,” *International Journal of Food Properties*, (2020), vol. 24, no. 1, pp. 28–40. Available on: https://doi:org/10:1080/10942912:2020:1861008

[14] D. C. F. Rayner, "Optimization for heuristic search," ERA, (2014), Available on https://era:library:ualberta:ca/items/57ecc087-1494-41b0-aa98-8f3bedcaa25f

[15] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial bee colony (abc) algorithm," (2011), vol. 11, pp. 652–657.

[16] X.-S. Yang, "Nature-inspired algorithms," in *Nature-Inspired Optimization Algorithms,* Ed. Oxford: Elsevier, (2014), p. i. ISBN 978-0-12-416743-8. [Online]. Available on: https://www:sciencedirect:com/science/article/pii/B9780124167438000166

[17] V. Kenny. "Heuristic algorithms," Optimization, (2014), Available on: https://optimization:mccormick:northwestern:edu/index:php/Heuristic/Algorithms

[18] F. E. Fernandes and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification," *Swarm and Evolutionary Computation*, (2019), vol. 49, pp. 62–74. Available on https://doi:org/10:1016/j:swevo 05:010

[19] A. Ghosh, A. Sufian, F. Sultana, and et.al., *Fundamental Concepts of Convolutional Neural Network*. ResearchGate, Jan (2020), DOI:10:1007/978-3-030-32644-9 36

[20] Aishwarya, "Introduction to recurrent neural network," Machine Larning, (2021), Available on: https://www:geeksforgeeks:org/introduction-to-recurrent-neural-network/

[21] Shaadk7865, "Particle swarm optimization (pso) an overview," GeeksforGeeks, (2023), Available on : https://www:geeksforgeeks:org/particle-swarm-optimizationpso-an-overview/#article-meta-div

[22] Y. Lu, "Food image recognition by using convolutional neural networks (cnns)," *Computer Vision and Pattern Recognition (cs arXiv:1612.00983)*, (2019), vol. 2, pp. 1–6. Available on: https://doi:org/10:48550/arXiv:1612:00983

[23] S. Hou, Y. Feng, Z. Wang, and Vegfru, "A domain-specific dataset for fine-grained visual categorization," *In Proceedings of the 2017 IEEE International Conference on Computer Vision*, (2017), pp. 541–549.

[24] N. Aherwadai, U. Mittai, J. Singla, and et. al., "Prediction of fruit maturity, quality, and its life using deep learning algorithms," *Electronics 2022*, (2022), vol. 11, no. 24, pp. 1–13. https://doi:org/10:3390/electronics11244100

[25] T. Khan, J. Qiu, M. A. A. Qureahi, and et. al., "Agricultural fruit prediction using deep neural networks," *Procedia Computer Science*, (2022), vol. 174, pp. 72–78. Available on: https://doi:org/10:1016/j:procs:2020:06:058