JOURNAL LA MULTIAPP

VOL. 05, ISSUE 04 (424-439), 2024
DOI: 10.37899/journallamultiapp.v5i4.1461

Implementation of a Convolutional Neural Network Algorithm in Classifying Vegetable Freshness Based on Image

Dysa Handira[1], Muhammad Siddik Hasibuan[1]

[1]*Department of Computer Science, Faculty of Science and Technology, Universitas Islam Negeri Sumatera Utara*

**Corresponding Author: Dysa Handira*
Email: handiradysa12@gmail.com

Article Info

Article history:
Received 3 July 2024
Received in revised form 27 July 2024
Accepted 12 August 2024

Keywords:
Convolutional Neural Network (CNN), Image classification, Phyton, Keras/Tensorflow.

Abstract

The purpose of this work is to apply CNN algorithm to a real problem of vegetable freshness identification using image data. Quantitative approach was used for this study and the data source was obtained from Kaggle; it is referred to as Fresh and Stale Images of Fruits and Vegetables with 2,604 images, four categories in total. The CNN model architecture consisted of a basic organization of four successive convolutional layers with associated max-pooling layers that aimed at capturing hierarchical feature representations of the input images. This model was trained using the Adam's optimizer for 20 iterations with the batch size of 32. Pre-processing of data included image augmentations such as scaling, rotation, flipping which improved the performance of the model. The assessment was done using Confusion Matrix approach and the results show that the proposed system achieved an accuracy of 95%, with a precision of 94%, recall of 93% and F1-score of 93%. From this it can be concluded that the CNN model proposed has achieved the objective of distinguishing fresh and non-fresh vegetables with enough precision to assist in the automation of quality control in agriculture. The conclusion that can be drawn from this study is that AI especially CNNs could be of big help in increasing accuracy and decreasing human factors in the large scale production of food.

Introduction

Freshness vegetable is very important factor in guard quality, taste and benefits nutrition from vegetables. When fresh vegetables are consumed, we can feel tenderness, richer taste, and gain benefit optimal nutrition. Apart from that, fresh vegetables also provide life and energy new to in life We. For guard freshness vegetable, important For choose fresh vegetables with signs quality like color bright, firm texture, and fresh aroma. Not a little Difficult buyers differentiate type, type, and freshness vegetable. So, buyers only follow instruction from seller (Lubis, 2020).

Artificial Intelligence (AI) especially technology *machine learning*, has interesting attention big in a number of year final with aim For possible computer For simulate intelligence human and helpful handle tasks in the real world. Learning process specific and detailed known as *Deep Learning*. *Deep Learning* is a learning process that uses algorithms that refer to law the math works like brain. *Deep Learning* used For various type work like predict opportunity or events, recognize object or object, and also can diagnose disease (Xie et al., 2021). *Deep learning* own a number of one method *Convolutional Neural Network* (CNN) (Efendi et al., 2024). *Convolutional Neural Networks* can used or implemented For introduction image with rivaling accuracy humans on a certain dataset and *Convolutional Neural Network* can learn

ISSN: 2716-3865 (Print), 2721-1290 (Online)

type features that produce more accuracy high and therefore use it in the classification process (Paraijun et al., 2022) .

Convolutional Neural Network (CNN) incl in kind of deep learning because depth the network (Ketkar et al., 2021). Deep learning is branch from machine learning that can teach computer For do work appropriately human, like computer can Study from the training process. CNN is operation combining convolutions a number of layer processing , using a number of operating elements in a way parallel and inspired by the system nerve biological (Romario et al., 2020). Development Technology Continuous current information develop, can help overcome various problem (Chen, 2017). Because related matters with progress Technology Information has spread to almost all over layer Indonesian society. One of them can used in matter classification freshness vegetable. Where is one the problem that is in classify freshness vegetables, vegetable distributor need system that can classify freshness vegetable with amount Vegetables are plentiful and needed fast time (El-Ramady et al., 2015).

In Indonesia itself quality vegetable national seen from side consistency size, color, and freshness Still Far from hope consumer Because many vegetables that don't can classified its freshness in a way whole (Paraijun et al., 2022). According to Hidayat & Lusiana (2022), CNN was used For classify type vegetables based on features that have been extracted. After through layer convolution and pooling, features the submitted to layers connected full (fully connected layers) which allows the model to learn connection between features these and categories the right vegetables. This allows the model to predict with accurate type vegetables from given image.

Based on description above, research This aim For apply algorithm *Convolutional Neural Networks* in classification freshness in the vegetables represented in something pictures, and for prove level success method *Convolutional Neural Networks* in do classification object. Researcher hope that with exists study this, got it lighten up work public in introduction quality vegetable based on image with big amount For know freshness vegetable is vegetable is it fresh or not fresh (Paraijun et al., 2022).

Methods

This research work adopted the quantitative research design whereby the primary data source was in the form of numbers, statistics, and quantitative measures to investigate the classification of vegetable freshness with the CNN algorithm. The research design and methodology was well guided to enable smooth running of the research process through data collection and preprocessing, model implementation, and evaluation. The data for this study were collected from Kaggle site and the data is called Fresh and Stale Images of Fruits and Vegetables. It contains 12 different classes of fruit and vegetable images and from that, 4 classes were chosen for the best performance of the system. The selected classes were bitter gourd – raw and cooked and capsicum – raw and cooked. The training dataset included 2082 images while 522 images were used in the testing dataset and the complete dataset was 2604 images. This division of data was made in a systematic way so as to maintain proportionality of each class as the models were being trained and tested.

This preprocessing was used in enhancing the image data in readiness for the model training exercise. The images were then adjusted to a standard target dimension of 150x150 pixels so as to have a uniform input size. Several augmentations were performed to improve the model performance by feeding it with different images hence making it more adaptive. The specific augmentation parameters she found is shown in the Table 1 below.

Table 1. Preprocessing Parameters

Process Name	Description
Rescale	Normalization of pixel values to the range [0,1] by dividing by 255.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Rotation Range	Images were randomly rotated within a range of -40° to 40° to simulate different angles of image capture.
Shear Range	A shear transformation was applied with a shear intensity of 0.2, altering the image's geometry.
Zoom Range	A zoom range of 0.2 was applied to enhance the model's ability to recognize features at different scales.
Horizontal Flip	Images were randomly flipped horizontally to increase variability in the dataset.
Fill Mode	Nearest neighbor interpolation was used to fill in any missing pixels created during transformations.

CNN was used by establishing an architecture that followed a deep learning format comprised of four convolutional layers and subsequently followed by max-pooling layers. The convolutional layers were utilized to obtain the hierarchical features from the input images and the feature maps were downsampled by the max-pooling layers to reduce computational cost and to avoid overfitting. CNN architecture details are shown in Table 2:

Table 2. CNN Model Architecture

Layer	**Configuration**	**Description**
Layer 1	32 filters, 3x3 kernel size, ReLU activation, 2x2 max-pooling	Extracts low-level features, downsampled by max-pooling.
Layer 2	64 filters, 3x3 kernel size, ReLU activation, 2x2 max-pooling	Doubles filters to capture more complex patterns, followed by downsampling.
Layer 3	128 filters, 3x3 kernel size, ReLU activation, 2x2 max-pooling	Further increases filter count to refine feature extraction.
Layer 4	256 filters, 3x3 kernel size, ReLU activation, 2x2 max-pooling	Final layer of feature extraction and downsampling.

CNN was set up by designing an architectural model that was inherently a deep learning model consisting four convolutional layers that in turn were followed by max-pooling layers. An application of the convolutional layers was used to obtain the hierarchical features of the input images and the feature maps are downsampled by the max-pooling layers to reduce the computations and to prevent overfitting. The CNN architecture of details are presented in table two below.

Table 3. Model Training and Evaluation Parameters

Parameter	**Details**
Optimizer	Adam
Batch Size	32
Epochs	20
Evaluation Metrics	Accuracy, Loss, Recall, Precision, F1-score
Accuracy	95%
Average Precision	94%
Average Recall	93%
F1-score	93%

The data was formed by inserting a random shift into training data and evaluated by performing a random crop into the training data; the mini-batches with 32 samples for the training data were made by the Adam optimizer; the total numbers of epochs for the training and validation data were set as 20. The specifically chosen optimizer is known as the Adam optimizer namely for the convenience that originates in the capacity to automatically manage the learning rates during the training process, thus giving a boost to convergence and enhancing the overall performance. For the purpose of evaluation, such fundamental

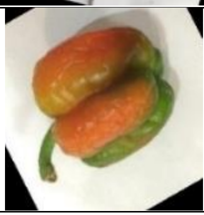
ISSN: 2716-3865 (Print), 2721-1290 (Online)

indicators as accuracy, loss, recall, precision, and F1-score were used. These metrics were computed based on confusion matrix obtained in the testing phase of the model. Evaluation parameters of the method are presented in the Table 3 below.

Results and Discussion

Collected data For study This through a dataset taken on the Kaggle website https://www.kaggle.com/datasets/raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables where there are 4 classes type vegetable where the total number of vegetables that will be tested is 2604 vegetable images to see the images of the vegetables being tested as following:

Table 4. Vegetable Image Data

No	Vegetable Image Data	Label Name
1		fresh_bitter_gourd
2		fresh_capsicum
3		stale_bitter_gourd
4		stale_capsicum

Pre - processing

In this research, the pre-processing process was carried out by randomly dividing the raw/original vegetable image data into two parts, 80% for *training data* and 20% for *testing data*. The raw data used totaled 2604 images. Thus, 2082 images were used as *training data*, while the remaining 522 images were used as *testing data*. This random distribution aims to ensure that the model developed has sufficient data for learning and sufficient data for testing and evaluating the model's performance in a representative manner (Efendi, 2024). Following This is division of training data and testing data used in this research :

Table 5. Results of Distribution of *Training* and *Testing Data*

Class	Total Class	Amount of Training Data	Amount of Testing Data
Fresh Bitter Gourd	316	252	64
Fresh Peppers	1031	824	206
Bitter Gourd Not Fresh	357	285	72
Bell Peppers Are Not Fresh	901	721	180
Total	**2,604**	**2,082**	**522**

ISSN: 2716-3865 (Print), 2721-1290 (Online)

After dividing *the training* and *testing data*, the data will be processed to implement the use of Tensorflow/Keras to load image data from the directory and use it as input for training and testing the model.

```
[54] # Set seed untuk reproducibility
     seed = 20
     np.random.seed(seed)
     tf.random.set_seed(seed)
     random.seed(seed)

[55] # Preprocess the Data training & testing
     train_datagen = ImageDataGenerator(
         rescale=1./255,
         rotation_range=40,
         shear_range=0.2,
         zoom_range=0.2,
         horizontal_flip=True,
         fill_mode='nearest'
     )
     train_generator = train_datagen.flow_from_directory(
         '/content/drive/MyDrive/SAYUR/Train',
         target_size=(150, 150),
         batch_size=32,
         class_mode='categorical'
     )
     test_datagen = image.ImageDataGenerator(rescale=1./255,)
     test_generator = test_datagen.flow_from_directory(
         '/content/drive/MyDrive/SAYUR/Test',
         target_size=(150, 150),
         batch_size=32,
         class_mode='categorical'
     )
```

Figure 6. Code for Vegetable Data Augmentation and Feature Extraction

Based on *code feature extraction* on rass data, it can be explained as follows:

Table 6. Explanation *of Code* for *Augmentation* and *Feature Extraction of Vegetable Data*

Process Name	**Information**
rescale	rescale will do image data normalization becomes 0 – 1, with method share image data with amount maximum The RGB color pixel size is 255.
rotation_range	Augmentation *rotation_range* This used For rotate picture in a way random with mark rotation Can from 0 ˚ -180 ˚. However in study This rotation value used that is 4 0. So Later picture will be rotated in a way random from (- 4 0˚) to 4 0˚.
shear_range	Augmentation share_range This used For transform shift image, so picture Can shaped line up parallelogram. On research This share_range to use value 0.2
zoom_range	Augmentation this zoom_range used For Enlarge the image, so that the image can be seen more clearly. On research This zoom _range used value 0.2
horizontal_flip	*horizontal_flip* will carry out the process of reversing image data in a way random.
fill_mode	*fill_mode* This used For fill in point or pixels in space blank after do shift or rotation picture. On research This use nearest mode wich one will fill in pixels new with use mark pixels nearest available.
target_size	U target size for loaded image. Image will changed the size become size specified here
batch_siz e	Amount sample will given to the model on each iteration training.
class_mode	This shows that the model will uses categorical labels (one-hot encoded) for existing classes.
train_generator	For training
test_generator	For testing

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Convolutional Neural Network Modeling

After the *preprocessing process is carried out*, the image data will then be processed in the *convolutional neural network model* by going through 2 stages, including:

Feature Learning

Stages First from this CNN model is feature learning, in research This this learning feature itself Alone consists of 2 parts namely convolutional layers and max pooling layers. Following This is the code used For perform feature learning:

```
[7]  # Membuat CNN Model
     model = Sequential([
         #layer 1
         Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
         MaxPooling2D(2, 2),

         #layer 2
         Conv2D(64, (3, 3), activation='relu'),
         MaxPooling2D(2, 2),

         #layer 3
         Conv2D(128, (3, 3), activation='relu'),
         MaxPooling2D(2, 2),

          #layer 4
         Conv2D(256, (3, 3), activation='relu'),
         MaxPooling2D(2, 2),
```

Figure 7. Code in Feature Learning

Following This explanation about *code* For *learning features:*

Table 7. Explanation of *the Code* in *Feature Learning*

Layer 1		
Convolutional Layer 1	At the convolutional layer 1 stage, the filter size used is 32, with a kernel size of 3x3 matrix , using as many strides as possible 0 , and with use function Activation form ReLU . The input data for conv_1 is the input layer.	Convolutional Layer 1 Calculation: $n_{(out)} = \left[\frac{n_{in}+2p-k}{stride}\right]+1$ $=\left[\frac{150+2(0)-3}{0}\right]+1$ $= 147 + 1$ $= 148$ Result of Calculation *Convolutional Layer 1:* 148 x 148 x 32
Max Pooling layer 1	After passing through convolutional layer 1, next image will done compression size image on max pooling layer 1, with use size pooling matrix of 2 x 2.	$n_{(w,h)} = \frac{\left(n_{((w,h)-1)} - f\right)}{stride} + 1$ $=\frac{(148-2)}{2}+1$ $=\frac{146}{2}+1$ $= 74$ Result of Calculation *Max Pooling Layer 1:* 74 x 74 x 32
Layer 2		
Convolutional Layer 2	In *convolutional layer 2*, the input data will be processed in *Convolutional layer 2* originate from *max pooling layer 1*, in convolutional layer 2 is image data will return done processing with using a filter of 64, the kernel size is 3 x 3 matrix , and using stride size of 0 . Apart from that, convolutional *layer 2* this will too used function Activation form ReLU .	Convolutional Layer 1 Calculation: $n_{(out)} = \left[\frac{n_{in}+2p-k}{stride}\right]+1$ $=\left[\frac{74+2(0)-3}{0}\right]+1$ $= 71 + 1$ $= 72$ Result of Calculation *Convolutional Layer 2 :* 72 x 72 x 64

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Max Pooling layer 2	After pass *convolutional layer 2* , the input data will be processed in *max pooling layer 2* originate from *convolutional layer 2* . At *max pooling layer 2* This size matrix used equal to 2 x 2.	$n_{(w,h)} = \frac{\left(n_{((w,h)-1)} - f\right)}{stride} + 1$ $= \frac{(72-2)}{2} + 1$ $= \frac{70}{2} + 1$ = 36 Result of Calculation *Max Pooling Layer 2 :* 36 x 36 x 64
Layer 3		
Convolutional Layer 3	Next, convolutional *layer 3* image data return processed , with the input data originating from *max pooling layer 2.* In processing image in *convolutional layer 3* This is the filter used is 128 , with kernel size of 3 x 3 matrix , and using function Activation ReLU with *stri d es* equal to 0	$n_{(out)} = \left[\frac{n_{in}+2p-k}{stride}\right] + 1$ $= \left[\frac{36+2(0)-3}{0}\right] + 1$ = 33 + 1 = 34 Result of Calculation *Convolutional Layer 3 :* 34 x 34 x 128
Max Pooling layer 3	After passes through convolutional layer 3, then image data will return processed in max pooling layer 3 with the goal is that the CNN model can understand form image that will processed.	$n_{(w,h)} = \frac{\left(n_{((w,h)-1)} - f\right)}{stride} + 1$ $= \frac{(34-2)}{2} + 1$ $= \frac{32}{2} + 1$ = 17 Result of Calculation *Max Pooling Layer 3 :* 17 x 17 x 128
Layer 4		
Convolutional Layer 4	Next, convolutional *layer 4* image data return processed, with the input data originating from *max pooling layer 2.* In processing image in *convolutional layer 4* This is the filter used is 256, with kernel size of 3 x 3 matrix , and using function Activation ReLU with *stri d es* equal to 0	$n_{(out)} = \left[\frac{n_{in}+2p-k}{stride}\right] + 1$ $= \left[\frac{17+2(0)-3}{0}\right] + 1$ = 34 + 1 = 15 Result of Calculation *Convolutional Layer 3:* 15 x 15 x 256
Max Pooling layer 4	After passes through convolutional layer 3, then image data will return processed in max pooling layer 3 with the goal is that the CNN model can understand form image which will processed.	$n_{(w,h)} = \frac{\left(n_{((w,h)-1)} - f\right)}{stride} + 1$ $= \frac{(15-2)}{2} + 1$ $= \frac{13}{2} + 1$ = 7 Result of Calculation *Max Pooling Layer 3 :* 7 x 7 x 256

Classification

After passing through feature learning then image data will enter second stage namely classification. In stages This there is a number of layers , among others that is fully connected layer and output layer. Following This is a code contained in the classification of testing image of fresh and non-fresh vegetables :

ISSN: 2716-3865 (Print), 2721-1290 (Online)

```
    #layer 4
    Conv2D(256, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),

    #Classification Process
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax')
])

# Compile model with Adam optimizer
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Figure 8. Code at the Vegetable Image Classification Stages

Based on the vegetable image classification code in Figure 8, it can be explained as follows:

Table 8. Explanation of Vegetable Image *Classification* Stages

Process Name	Information
Flatten	*Flatten layers* are used For make *input* that has Lots dimensions become A *vector*, so that later results from *flatten layer* can used as mark *input* from *fully connected layer*.
Dense	After *fully connected layer* has *input* data, next process is connected *input* that is on *the flatten layer* to in *layer classification*, with use as *dense* as 128 units and function Activation that is ReLU. Then at the final stage of classification because there are 4 classes to be tested, *the output layer* is *a softmax* function Activation *softmax* will displays results based *output probability* biggest from results classification.
Dropouts	Before enter *output layer*, moreover formerly results from *fully connected layer* will pass *dropout*, *dropout* this Alone used For delete *overlifting* occurs happen during the *fully connected layer* process.

Model Evaluation

Based on results tests carried out from the *Convolutional Neural Network* model on image data of fresh and non-fresh vegetables , obtained results for *training data* as following:

Table 9. Accuracy and *Loss Results* with Adam Optimization on *Training Data*

Batch_size	Epoch	Accuracy	Losses
32	20	0.9393	0.2399

training data testing using Adam optimization, with an epoch size of 20 epochs, and a batch size of 32 batches, the accuracy was 93% and the loss was 0.2399.

```
# Train the Model
history = model.fit(
            train_generator,
            steps_per_epoch=8,
            epochs=20,
    )
```


ISSN: 2716-3865 (Print), 2721-1290 (Online)

```
8/8 ━━━━━━━━━━━━━━━━━━━━ 14s 2s/step - accuracy: 0.8165 - loss: 0.5156
Epoch 9/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 213ms/step - accuracy: 0.7500 - loss: 0.5831
Epoch 10/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 19s 2s/step - accuracy: 0.8080 - loss: 0.4878
Epoch 11/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.8486 - loss: 0.4289
Epoch 12/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.8952 - loss: 0.2390
Epoch 13/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.8982 - loss: 0.2350
Epoch 14/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.9483 - loss: 0.1315
Epoch 15/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.9612 - loss: 0.1192
Epoch 16/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.9756 - loss: 0.1409
Epoch 17/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 13s 2s/step - accuracy: 0.9347 - loss: 0.1685
Epoch 18/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 213ms/step - accuracy: 0.8203 - loss: 0.6564
Epoch 19/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 30s 2s/step - accuracy: 0.9541 - loss: 0.2446
Epoch 20/20
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.9393 - loss: 0.2399
```

Figure 9. Train Data Evaluation Results

Based on results training data *training* the so can We shown in the plot/ graph of course in Figure 10 below.

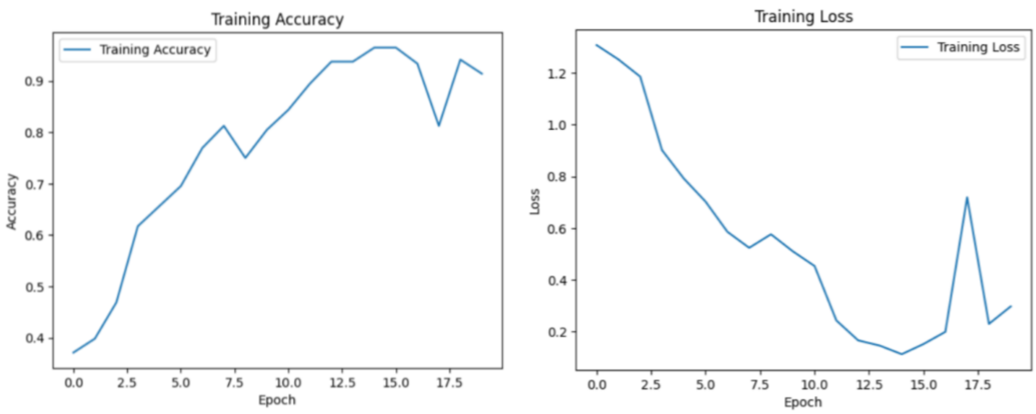


Figure 10. Accuracy and Loss Data Train Plot

From figure 10, we can analyze exists connection between mark accuracy and loss value on training data with number of epochs or iterations carried out. Observed relationship to grades accuracy show correlation positive, which means that the more Lots number of epochs used in training, grades accuracy of training data tends increase. On the other hand, the loss value shows correlation negative with number of epochs, meaning the more many epochs are used, the loss value in the training data tends to be decrease. Based on observation this, got it concluded that for reduce desired loss value, add number of epochs in the training process Can be one solution effective.

Confusion Matrix

After we train the model, then we evaluate the performance of the model on the test set. Evaluation is done to get the probability of failure of an image object that is read in the classification process, then the accuracy value with the highest probability will be obtained from the entire test model.

The classification results obtained from training data and test data are shown in Figure 11 showing the prediction results for training data. Vegetable images that were successfully classified were all Fresh Bitter Gourd images, namely 61 out of 64 images, Fresh Capsicum images that were successfully classified were 197 out of 206 images used, Satle Bitter Gourd

ISSN: 2716-3865 (Print), 2721-1290 (Online)

images that were successfully classified were 60 out of 72 images used and Stale Capsicum images that were successfully classified were 178 out of 180 images.

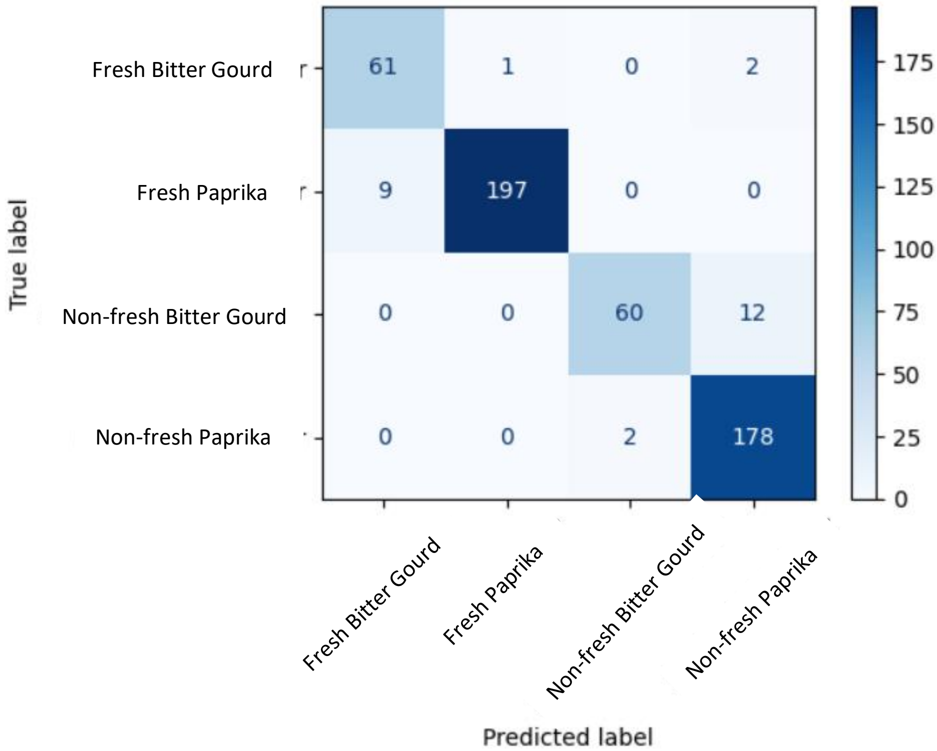


Figure 11. Confusion Testing Data Matrix

From the *confusion matrix* can obtained results form mark predictions *recall* and *precision* of the CNN model, below This results accuracy , *recall* and *precision* of vegetable data :

Accuracy

From the results of the confusion matrix, the overall average accuracy results are:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \text{ x } 100\%$$

$$Accuracy = \frac{61 + 197 + 60 + 178}{61 + 197 + 60 + 178 + 1 + 2 + 9 + 12 + 2} \text{ x } 100\%$$

$$Accuracy = 95\%$$

Recall

From the results *confusion matrix* can is known results *recall* , which is where mark this *recall* used For know how much the precision of the model when matched return with use different image . Following This results calculation mark *recall* for data on fresh vegetables and non-fresh vegetables :

Fresh Bitter Gourd

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{61}{61 + 1 + 2}$$

$$Recall = 0.953$$

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Based on results calculation , value *recall* from fresh bitter gourd the size of 0.953 which means level accuracy model fit when matched return with different image own level accuracy amounting to 95 %.

Fresh Capsicum

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{197}{197 + 9}$$

$$Recall = 0.956$$

Based on results calculation , value *recall* from fresh capsicum as big as 0.956 which means level accuracy model fit when matched return with different image own level accuracy by 97%.

Stale Capsicum

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{60}{60 + 12}$$

$$Recall = 0{,}833$$

Based on the calculation results, the recall value of the stale bitter gourd is 0.833, which means that the accuracy of the model match when matched again with a different image has an accuracy rate of 83%.

Stale Capsicum

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{178}{178 + 2}$$

$$Recall = 0.988$$

Based on the calculation results, the recall value of the stale capsicum is 0.988, which means that the accuracy of the model match when matched again with a different image has an accuracy rate of 99%

Precision

In the confusion matrix, results are also obtained precision, which is where results mark precision works for know how much the precision of the CNN model when tested with another image. Following This results calculation mark precision for fresh and non-fresh vegetable data :

Fresh Bitter Gourd

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{61}{61 + 9}$$

$$Precision = 0.871$$

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Based on results calculation, value precision of fresh bitter gourd vegetables as big as 0.871 which means level accuracy of deep CNN models identify amounting to 87%.

Fresh Capsicum

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{197}{197 + 1}$$

$$Precision = 0.994$$

Based on results calculation, value precision of fresh capsicum vegetables as big as 0.994 which means level accuracy of deep CNN models identify the vegetable as big as 99%.

Stale Bitter Gourd

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{60}{60 + 2}$$

$$Precision = 0.967$$

Based on the calculation results, the precision value of the stale bitter gourd vegetable is 0.967 which means that the accuracy of the CNN model in identifying the vegetable is 97%.

Stale Capsicum

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{178}{178 + 12 + 2}$$

$$Precision = 0.927$$

Based on the calculation results, the precision value of stale capsicum vegetables is 0.927 which means that the accuracy of the CNN model in identifying these vegetables is 93 %.

From evaluation the so obtained *classification report* data on picture following .

Classification Report:

	precision	recall	f1-score	support
Labu Pahit Segar	0.87	0.95	0.91	64
Paprika Segar	0.99	0.96	0.98	206
Labu Pahit Tidak Segar	0.97	0.83	0.90	72
Paprika Tidak Segar	0.93	0.99	0.96	180
accuracy			0.95	522
macro avg	0.94	0.93	0.93	522
weighted avg	0.95	0.95	0.95	522

Figure 12. Classification Report Data Testing Results

On picture on is results *classification report* from evaluate this model , can seen there is mark *recall, fi-score* and *precision* of each existing vegetable label in the *test* data. *Recall* (sensitivity) is ratio predictions Correct positive compared to with entire data correct positive. *F1-Score* is comparison of average precision and *recall* weight it. And *precision* is ratio predictions Correct positive compared to n overall predicted results positive.

Classification Results

After produce CNN configuration and architecture then the parameters obtained implemented through help Language Python programming to test whether the classification results on

ISSN: 2716-3865 (Print), 2721-1290 (Online)

vegetable images are well predicted after going through the CNN model testing phase which has been tested previously.

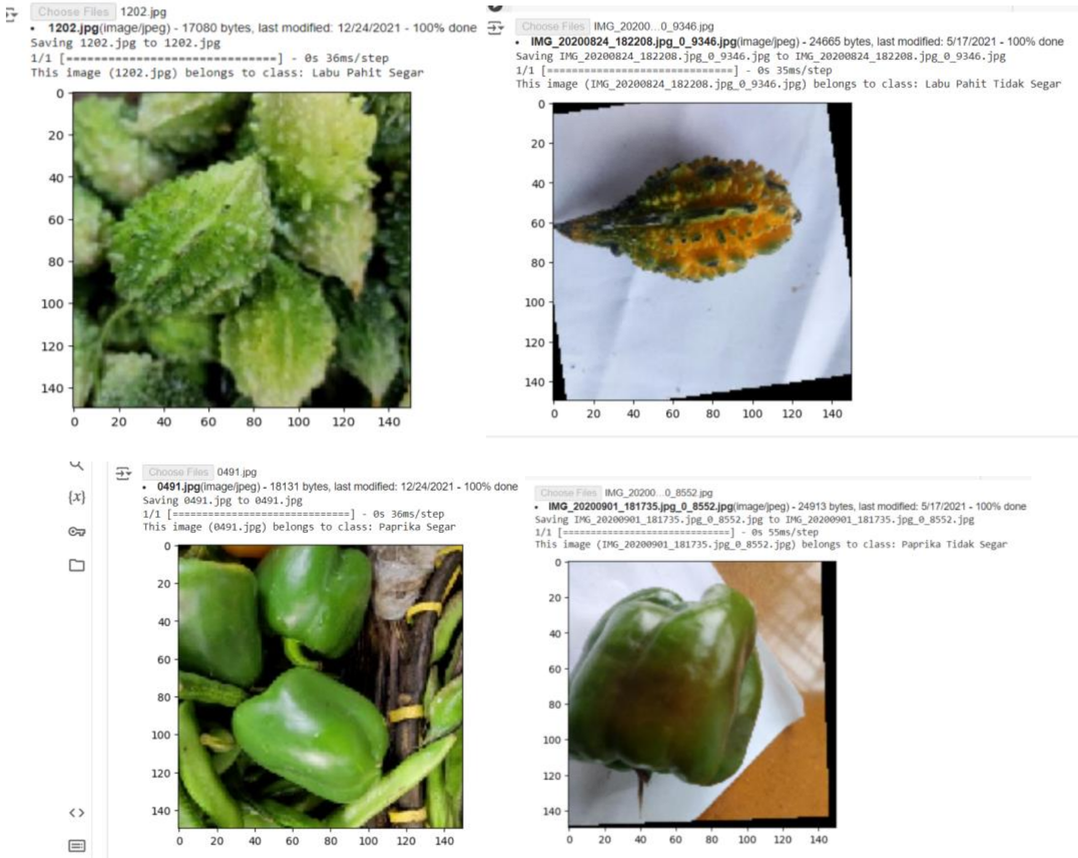


Figure 12. Classification Results

The obtained high accuracy also proves that the application of CNNs, which incorporate the mechanism of hierarchical feature extraction, is relevant for tasks connected with image classification in the sphere of agriculture. This finding is consistent with other current studies indicating that deep learning is efficient in agriculture thus improve precision and automation of quality control. This brings operability since CNNs help to reduce human errors and labor costs which are paramount in large-scale food production and distribution by facilitating the assessment of the freshness of produces in question.

However, this study's use of CNNs demonstrates AI's capacity to be incorporated into the vast concept of smart agriculture. Recent developments in AI and ML have indicated that such technologies can improve the living standards and the productivity of agriculture through efficiency in the usage of such resources and reduction of wastage (Gill et al., 2022; Zhang & Aslan, 2021; Cioffi et al. 2020; Li et al., 2021). Accordingly, the practical contribution of the current study goes far beyond the particular context of vegetable classification and can be viewed as the conceptualisation of how the design of an informative space of AI models may provide solutions for decision-making agents in the agricultural sector.

The first and major advantage of this study relates to the methodological part, including preprocessing and data augmentation of image data. Besides, when implemented the model scaling, rotation and flipping techniques not only improved its ability to generalize from previous training data but also helped train a more robust CNN on images that were filtered distorted in various ways. This kind of approach is in line with general practice in deep learning, wherein authors tend to make use of data augmentation so as to emulate actual use and reducing overfitting (Zhou et al., 2021; Chaudhari et al., 2021; Kiyasseh et al., 2021).

ISSN: 2716-3865 (Print), 2721-1290 (Online)

However, the following are the limitations of the study. The given dataset, though good enough for the purposes of the current research, is not diverse enough. The consideration of only two vegetable varieties, namely the bitter gourd and the capsicum, also leads to query the model's applicability to other kinds of produce. This can be a serious limitation especially since there are so many types of vegetables and fruits which may have different colours and textures. has rightly pointed out that how effective CNN models are in the applications of agriculture all depends with the data used for training these models. Future research should, therefore, try to use a wider variety of produce to ensure that the model can be applied in different conditions because of its usefulness (Alzubaidi et al., 2021; Li et al., 2021; Taye, 2023; Kattenborn et al., 2021; Salehi et al., 2023).

However, there are two conspicuous drawbacks regarding the stimulus materials; the first is that the images were acquired in a highly controlled manner. Though, the given work successfully showcased the viability of the model, real-world applications would prove much more complex and include such factors as inconsistent illumination, differences in background, and quality of the captured image (Eller et al., 2022; Elhanashi et al., 2023; Akande et al, 2022; Zhu et al., 2024). These factors could go a long way in affecting the model in question. According to Shaikh et al. (2022), to be practical in farming AI models, they should be checked beneath variable environmental situations. Therefore, future research should focus on applying the model under different situations with the goal of establishing its practical usability.

Conclusion

The image classification process in this study researchers separated 2 types of fresh and non-fresh vegetable categories, then randomly separated the raw dataset from 2064 vegetable images into 80% training and 20% testing. Then enter data preprocessing, data augmentation used in the form of scaling, rotation, translation, and flipping. Then enter the CNN model stage, then the researcher compiles the model with Adam's optimizer and then trains the training model, after entering the training model stage, the researcher evaluates the model with a graph plot and confusion matrix. On research This is the model process that occurs in CNN algorithm consists of 2 parts, namely feature learning and classification. The data has been done feature extraction will process at convolution layer 1, after goes through the feature learning process at convolutional layer 1, then processes the image data will done processing back in the max pooling layer aiming for zoom out image data size, so that the CNN model is designed can Can understand more details regarding data on fresh and non-fresh vegetables. The feature learning process is carried out until 4 layer, after pass feature learning then image will changed become A vector in flatten layer with aim for results from feature learning can used as mark input on stage classification. To measure loss and accuracy performance, training and testing of vegetable image datasets using the CNN model is quite good. Based on the CNN model designed, using the number of batch_size as much as 32 and a total of 20 epochs, the accuracy of fresh vegetables and non-fresh vegetables is 93% with a loss of 0.2399. The test results using the Confusion Matrix method obtained an accuracy value of 95%, an average precision value of 94%, an average recall value of 93%, and an f1-score value of 93%.

References

Akande, T. O., Alabi, O. O., & Ajagbe, S. A. (2022). A deep learning-based CAE approach for simulating 3D vehicle wheels under real-world conditions. In *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA42021882

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges,

ISSN: 2716-3865 (Print), 2721-1290 (Online)

applications, future directions. *Journal of big Data*, *8*, 1-74. https://doi.org/10.1186/s40537-021-00444-8

Chaudhari, A. S., Sandino, C. M., Cole, E. K., Larson, D. B., Gold, G. E., Vasanawala, S. S., ... & Langlotz, C. P. (2021). Prospective deployment of deep learning in MRI: a framework for important considerations, challenges, and recommendations for best practices. *Journal of Magnetic Resonance Imaging*, *54*(2), 357-371. https://doi.org/10.1002/jmri.27331

Chen, L. (2017). Continuous delivery: overcoming adoption challenges. *Journal of Systems and Software*, *128*, 72-86. https://doi.org/10.1016/j.jss.2017.02.013

Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., & De Felice, F. (2020). Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. *Sustainability*, *12*(2), 492. https://doi.org/10.3390/su12020492

Efendi, A. M. A., Sriani, S., & Hasibuan, M. S. (2024). Classification of Watermelon Ripeness Levels Using HSV Color Space Transformation and K-Nearest Neighbor Method. *Journal of Computer Networks, Architecture and High Performance Computing*, *6*(3), 934-948. https://doi.org/10.47709/cnahpc.v6i3.3999

Elhanashi, A., Dini, P., Saponara, S., & Zheng, Q. (2023). Integration of deep learning into the iot: A survey of techniques and challenges for real-world applications. *Electronics*, *12*(24), 4925. https://doi.org/10.3390/electronics12244925

Eller, L., Svoboda, P., & Rupp, M. (2022). A deep learning network planner: Propagation modeling using real-world measurements and a 3D city model. *IEEE Access*, *10*, 122182-122196. https://doi.org/10.1109/ACCESS.2022.3223097

El-Ramady, H. R., Domokos-Szabolcsy, É., Abdalla, N. A., Taha, H. S., & Fári, M. (2015). Postharvest management of fruits and vegetables storage. *Sustainable Agriculture Reviews: Volume 15*, 65-152. https://doi.org/10.1007/978-3-319-09132-7_2

Gill, S. S., Xu, M., Ottaviani, C., Patros, P., Bahsoon, R., Shaghaghi, A., ... & Uhlig, S. (2022). AI for next generation computing: Emerging trends and future directions. *Internet of Things*, *19*, 100514. https://doi.org/10.1016/j.iot.2022.100514

Hidayat, A. R., & Lusiana, V. (2022). Deteksi Jenis Sayuran dengan Tensorflow Dengan Metode Convolutional Neural Network. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, *6*(2), 1032-1040. http://dx.doi.org/10.30645/j-sakti.v6i2.512

Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS journal of photogrammetry and remote sensing*, *173*, 24-49. https://doi.org/10.1016/j.isprsjprs.2020.12.010

Ketkar, N., Moolayil, J., Ketkar, N., & Moolayil, J. (2021). Convolutional neural networks. *Deep learning with Python: learn best practices of deep learning models with PyTorch*, 197-242. https://doi.org/10.1007/978-1-4842-5364-9_6

Kiyasseh, D., Zhu, T., & Clifton, D. (2021). A clinical deep learning framework for continually learning from cardiac signals across diseases, time, modalities, and institutions. *Nature Communications*, *12*(1), 4221. https://doi.org/10.1038/s41467-021-24483-0

Li, L., Rong, S., Wang, R., & Yu, S. (2021). Recent advances in artificial intelligence and machine learning for nonlinear relationship analysis and process control in drinking water treatment: A review. *Chemical Engineering Journal*, *405*, 126673. https://doi.org/10.1016/j.cej.2020.126673

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, *33*(12), 6999-7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lubis, C. (2022). Klasifikasi Buah Segar Dan Busuk Menggunakan Convolutional Neural Network Berbasis Android. *Jurnal Ilmu Komputer dan Sistem Informasi*, *10*(2). https://doi.org/10.24912/jiksi.v10i2.22551

Paraijun, F., Aziza, R. N., & Kuswardani, D. (2022). Implementasi Algoritma Convolutional Neural Network dalam Klasifikasi Kesegaran Buah Berdasarkan Citra Buah. *KILAT, 11*, No 1, 1–9. https://doi.org/10.33322/kilat.v10i2.1458

Putri, O. N. (2020). Implementasi Metode CNN Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing. *Dspc Repository*, 23-27. https://dspace.uii.ac.id/

Romario, M. H., Ihsanto, E., & Kadarina, M. T. (2020). Sistem Hitung dan Klasifikasi Objek dengan Metode Convolutional Neural Network. *Jurnal Teknologi Elektro, Universitas Mercu Buana*, *11* (02), 108–114. https://doi.org/10.24912/jiksi.v10i2.22551

Salehi, A. W., Khan, S., Gupta, G., Alabduallah, B. I., Almjally, A., Alsolai, H., ... & Mellit, A. (2023). A study of CNN and transfer learning in medical imaging: Advantages, challenges, future scope. *Sustainability*, *15*(7), 5930. https://doi.org/10.3390/su15075930

Shaikh, T. A., Rasool, T., & Lone, F. R. (2022). Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming. *Computers and Electronics in Agriculture*, *198*, 107119. https://doi.org/10.1016/j.compag.2022.107119

Taye, M. M. (2023). Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*, *11*(3), 52. https://doi.org/10.3390/computation11030052

Xie, S., Yu, Z., & Lv, Z. (2021). Multi-disease prediction based on deep learning: a survey. *Computer Modeling in Engineering & Sciences*, *128*(2), 489-522. https://doi.org/10.32604/cmes.2021.016728

Zhang, K., & Aslan, A. B. (2021). AI technologies for education: Recent research & future directions. *Computers and Education: Artificial Intelligence*, *2*, 100025. https://doi.org/10.1016/j.caeai.2021.100025

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., ... & Summers, R. M. (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, *109*(5), 820-838. https://doi.org/10.1109/JPROC.2021.3054390

Zhu, Z., Wang, X., Zhao, W., Min, C., Deng, N., Dou, M., ... & Huang, G. (2024). Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*. https://doi.org/10.48550/arXiv.2405.03520

ISSN: 2716-3865 (Print), 2721-1290 (Online)