Identification of the Maturity Level of Roasted Coffee Beans Using Architecture of InceptionV3 Convolutional Neural Networks

Jodion Siburian[1] Pradita Eko Prasetyo Utomo[*2], Benedika Ferdian Hutabarat [2], Natasya Yonike Ambarita[2], Robby Yehezkiel Pardomuan[2]

*Corresponding author
*ORCHID IDs: https://orcid.org/0009-0008-4787-4609

[1]*Department of Biology Education, Universitas Jambi, Indonesia*
[2]*Department of Information System, Universitas Jambi, Indonesia*
**email: pradita.eko@unja.ac.id*

Abstract

Coffee is a drink made from coffee beans that have gone through a roasting process and been ground into powder. Each type of coffee has significant differences in shape, texture, color according to the roasting system used and taste. Identification of types of coffee is difficult to differentiate with the naked eye so special skills are required. One method that can be used to identify types of coffee is to use digital image processing. One technology that utilizes digital image processing is deep learning. The Deep Learning method that currently has the most significant results in image processing is Convolutional Neural Network (CNN). This research is expected to produce a system that can classify coffee beans, especially Robusta coffee beans based on the level of maturity of the roasted coffee beans. This research classifies the maturity level of roasted coffee beans in the form of light roast or dark roast. For detection modeling, the proposed pre-trained model InceptionV3 is used. From the results obtained with accuracy reaching 99.84%, it can be concluded that the model created can perform classification well. With a loss value of only 0.0035, which is a very small loss value.

Keyword: CNN, InceptioanV3, beans of coffe, light roast, dark roast, roasted, maturity, deep learning

I. Introduction

Coffee is one of Indonesia's leading plantation commodities, Indonesia ranks fourth as the largest coffee producer in the world [1]. In 2017, based on data from the Central Statistics Agency (BPS) website, Indonesia was one of the largest coffee exporters in the world with 464 thousand tonnes exportable and an average of 450 thousand tonnes exported each year. [2]. As one of the largest coffee producers in the world, Indonesia must be able to maintain the quality of coffee in accordance with established standards to be able to compete with other countries which are also coffee producing countries. With 70% of total national production being used as an export commodity, there is a need to standardize the quality of coffee beans. In assessing the quality of coffee beans, various methods can be used, one of which is assessing the quality of coffee beans. To produce coffee that tastes good, there are several processes, one of which is the roasting process.[3] Roasting coffee beans is a very important process in the coffee industry which really determines the quality of the coffee drink obtained. This process turns unpalatable raw coffee beans into a delicious drink with a delicious aroma and taste. Coffee bean processing needs to be adjusted to consumer demand and preferences. The roasting levels consist of light roast (moderate roast), medium roast (medium roast), dark roast (mature roast) [4]. The roasting process will be relatively easier to control if the coffee beans have uniform size, texture, specific gravity, chemical structure and water content. However, in reality, there are big differences in each coffee bean, so that the roasting process becomes an art that requires skills and experience as requested by consumers [5].

So far, to be able to determine the level of maturity of coffee roasting, this has been done manually and requires an expert or specialist in this field[6]. Therefore, the qualification and classification process for coffee beans cannot run well and efficiently without the help of qualified technology. It is hoped that this alternative technological assistance can help increase the effectiveness of the process of classifying roasted coffee beans to make it better and more efficient[7]. There have been several studies discussing similar themes, but so far nothing has shown the effectiveness and sustainability of this system for determining the classification and qualification of roasted coffee beans. As in the research example [8], in this research the level of accuracy produced was very minimal, and also lacked efficiency and flexibility. This is proven by the system used being less up to date and the lack of training data. This research is expected to produce modeling that can classify roasted coffee beans. This research focuses on robusta coffee based on the maturity level of roasted coffee beans with the help of the CNN model using the PreTrained InCeptionV3 method.

II. Method

2.1 Coffee

Coffee beans that have been harvested will undergo a process that will determine their characteristics. Coffee cherries that have been harvested will first go through a process of separating the beans from the flesh and skin. Then the resulting seeds will be fermented, dried, until the epidermis is released. There are two popular processes carried out, namely natural drying (dry process) and washing or wet process [9]. Coffee beans that have gone through this process are called green beans which will then undergo a roasting process. Roasting is the process of roasting coffee beans at a temperature that can reach 250 degrees Celsius which will cause changes in the fat, sugar and water levels in the coffee. A high enough temperature will cause the moisture content in the coffee beans to decrease and then the sugar content in them will experience caramelization. The color of the coffee beans will also transform from green to black. The most popular coffee bean maturity levels as a benchmark in Indonesia are three classes, namely: light, medium and dark roast.

2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is developed based on the multilayer layer perceptron (MLP) which is designed or intended for processing two dimensional data in the form of images. CNN is a variant of the deep neural network because of its high network depth and is widely applied to complex image data [10]. The image classification process can basically only use MLP, but one of the weaknesses of this MLP method is that it is not suitable because it cannot store spatial information from the mind data and each pixel is considered an independent or large feature, thus allowing to get unfavorable results [11].

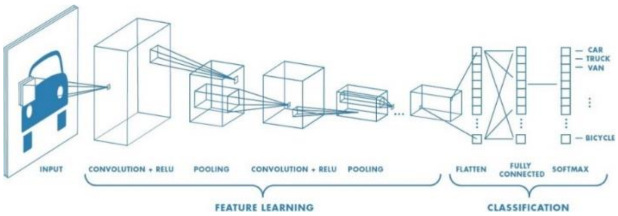

Figure 1. Architecture of CNN

The CNN method is the same as other neural network theories which are trained using the backpropagation algorithm. CNN is designed to recognize visual patterns directly from image pixels by minimizing preprocessing. CNN can recognize patterns with a wide variety, resistant to distortion and simple geometric transformations. The architecture of CNN is divided into major parts, namely the Feature Extraction Layer and the Fully Connected Layer. Each stage consists of three layers, namely the convolutional layer, the layer activation function, and the pooling layer, which can be seen in Figure 1 which is the CNN network architecture.

.

2.3 Design Research

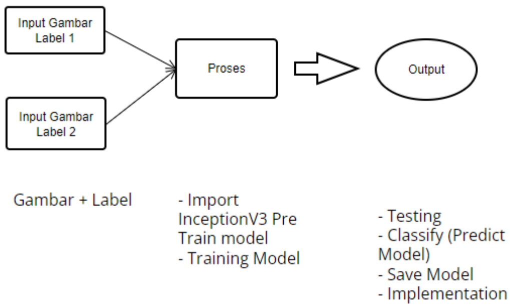

Figure 2. Illustration of System Model

In this research, the CNN model uses the Pre-Trained Model InceptionV3 method to solve problems related to Image Classification. Image detection modeling requires several images that have been obtained as input datasets that will be used in the image classification process. The number of images required is 300 images which are then divided into 2 classes according to the classification label. Then, after the image is obtained, the image must be converted to an image size ratio of 1:1 to make the training process easier. The process scheme for this modeling can be described using a flowchart scheme as follows.

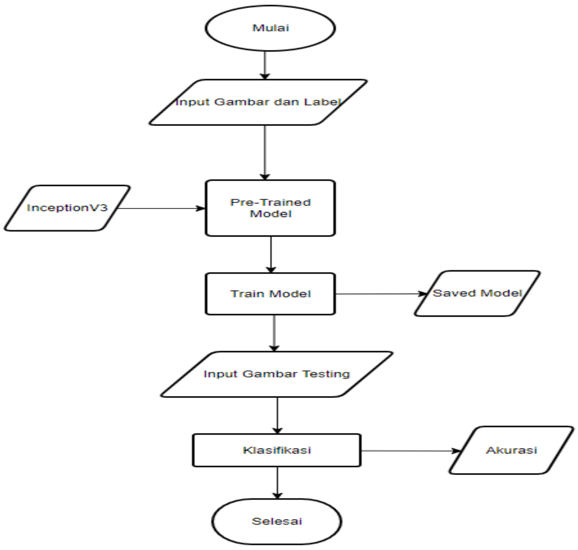

Figure 3. Modelling

As can be seen in Figure 3, modeling begins by inputting an image and inputting a label that corresponds to Image 1, then re-inputting Image 2 and Label 2 that corresponds to Image 2. After inputting the image and label, the image will be processed, which is called preTrained. InceptionV3 model. After that, the model will be trained and produce a training model. After getting the training model, the training model can be saved. After that, carry out the classification process with the training model, then upload the image files used for testing. When the image is finished uploading, it is generated and the classification process begins and produces output in the form of a probability diagram.

III. Results and discussion

3.1 Data Collection

The initial step of this research involves collecting data in the form of images divided into two classes, namely dark roasting and light roasting, with each consisting of 150 images. This data will be the main material in the training and evaluation of the model to be developed.

3.2 Remove Background

Once the data was collected, the next step was to perform background removal on each image so that only the main object, the coffee beans, remained visible. In its implementation on Google Colab, the Rembg library was used to remove the background on the images. Here are examples of images before (left) and after the background removal process (right):

Figure 4. Remove Background Image

This process helps to separate the main object, in this case the coffee beans, from its background so that the focus on the main object becomes clearer. The image after background removal will be easier for further processing in image analysis, such as coffee bean recognition and measurement.

3.3 Preprocessing and Resize

The preprocessing and resizing stages are important steps in image data processing. The aim is to ensure that the image data is in a suitable and high-quality format for model training. In this stage, the images are resized to 224x224 pixels, as this has become a common standard in image processing for convolutional neural network (CNN) model training. In addition, images that were not originally in RGB colour mode will be converted into RGB mode. The images that have been resized and processed are then stored in the folder corresponding to their class of origin. The results of the preprocessing and resizing image processes are as follows:

Figure 5. Preprocessing and resize

3.4 Augmentation Data

Data augmentation is a key stage in the development of a model for coffee bean roasting level detection and identification based on image features. The goal is to improve the quality and diversity of the dataset so that the resulting model has a better ability to recognize a wide variety of coffee beans. By using various augmentation techniques, such as zoom, rotation, horizontal flip, and vertical flip, the augmented images are stored in a new folder corresponding to their original class. The following is the result of the data augmentation process:

The initial data amount of 300 images increased dramatically to 1442 images after going through the data augmentation process.

Figure 6. Augmentation Data

```
Total number of photos before augmentation: 300
Total number of photos after augmentation: 1442
```

Figure 7. Total Data

3.5 Split Data

In this stage, the dataset is divided into two different subsets: training data and validation data. The aim is to prepare the data that will be used in training and validating the Convolutional Neural Network (CNN) model. This process is important to prevent overfitting and ensure that the model can generalize well to data that has never been seen before. The percentage of dataset division used is 70% for training data and 30% for validation data. Here are the results:

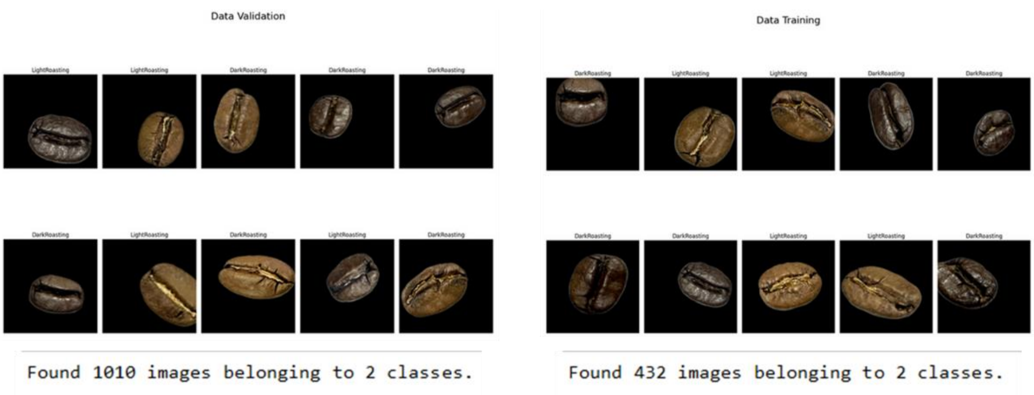

Figure 8. Split Data

3.6 Modelling

In this research, CNN method modelling is performed using the TensorFlow library in the Python programming language. The CNN model uses a pre-existing architecture, namely Inception V3, which has been pre-trained. Then, additional layers were added to the model. The following is a summary of the model that has been built.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 5, 5, 2048)        21802784

 global_average_pooling2d (  (None, 2048)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 128)               262272

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 2)                 130

=================================================================
Total params: 22073442 (84.20 MB)
Trainable params: 270658 (1.03 MB)
Non-trainable params: 21802784 (83.17 MB)
_________________________________________________________________
None
```

Figure 9. Summary of Model

Next, the CNN model was compiled using the ADAM optimizer, the loss function "categorical_crossentropy," and the accuracy metric. In the model training process, we implemented the "ReduceLROnPlateau" callback to reduce the learning rate if the 'val_loss' value does not decrease by a factor of 0.2 for 3 consecutive epochs. This aims to help the model converge better. In addition, we use the "EarlyStopping" callback to stop training if 'val_loss' does not decrease for 5 consecutive epochs, returning the best saved weights. The CNN model was trained by calling the "fit()" method using training data (train_generator) for 20 epochs while using validation data (valid_generator) for evaluation. Callbacks "reduce_lr" and "early_stopping" were used during training to optimize the model's performance. Training results and metrics such as accuracy and loss are stored in history variables. Here is the program code:

```
inception_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-6)
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

history = inception_model.fit(train_generator, epochs=10, validation_data=valid_generator,
                              verbose=2, callbacks=[reduce_lr, early_stopping])
```

Figure 10. Model Training

The output of the model training code provides information on loss, accuracy, time per epoch, and learning rate. This output helps monitor the model's progress from epoch to epoch, assess the loss reduction and accuracy performance on training and validation data, as shown in the following figure:

```
Epoch 1/10
64/64 - 140s - loss: 0.3030 - accuracy: 0.8792 - val_loss: 0.0851 - val_accuracy: 0.9653 - lr: 0.0010 - 140s/epoch - 2s
/step
Epoch 2/10
64/64 - 99s - loss: 0.0976 - accuracy: 0.9663 - val_loss: 0.2282 - val_accuracy: 0.9282 - lr: 0.0010 - 99s/epoch - 2s/s
tep
Epoch 3/10
64/64 - 97s - loss: 0.0881 - accuracy: 0.9653 - val_loss: 0.0343 - val_accuracy: 0.9815 - lr: 0.0010 - 97s/epoch - 2s/s
tep
Epoch 4/10
64/64 - 72s - loss: 0.0516 - accuracy: 0.9812 - val_loss: 0.0254 - val_accuracy: 0.9931 - lr: 0.0010 - 72s/epoch - 1s/s
tep
Epoch 5/10
64/64 - 67s - loss: 0.0234 - accuracy: 0.9931 - val_loss: 0.0221 - val_accuracy: 0.9884 - lr: 0.0010 - 67s/epoch - 1s/s
tep
Epoch 6/10
64/64 - 43s - loss: 0.0102 - accuracy: 1.0000 - val_loss: 0.0219 - val_accuracy: 0.9884 - lr: 0.0010 - 43s/epoch - 676m
s/step
Epoch 7/10
64/64 - 45s - loss: 0.0352 - accuracy: 0.9881 - val_loss: 0.0275 - val_accuracy: 0.9861 - lr: 0.0010 - 45s/epoch - 707m
s/step
Epoch 8/10
64/64 - 45s - loss: 0.0479 - accuracy: 0.9812 - val_loss: 0.0420 - val_accuracy: 0.9861 - lr: 0.0010 - 45s/epoch - 699m
s/step
Epoch 9/10
64/64 - 45s - loss: 0.0287 - accuracy: 0.9931 - val_loss: 0.0247 - val_accuracy: 0.9861 - lr: 0.0010 - 45s/epoch - 710m
s/step
Epoch 10/10
64/64 - 44s - loss: 0.0198 - accuracy: 0.9950 - val_loss: 0.0217 - val_accuracy: 0.9884 - lr: 2.0000e-04 - 44s/epoch -
693ms/step
```

Figure 11. Output of the training model

From the output, the development of accuracy and loss values during the model training process for 10 epochs can be observed. The peak performance is reached at the 10th epoch, with the training accuracy reaching 99% and the validation accuracy reaching 98%.

3.7 Evaluation Model

Model evaluation is an important step that aims to measure the performance and effectiveness of the model that has been built. In the evaluation stage, loss and accuracy values are calculated on the validation data. Loss is used to measure the extent to which the model can model the validation data well, while accuracy measures how well the model performs the correct classification on the data. The evaluation results give an idea of the extent to which the model is reliable for the actual task.
The model evaluation results on the validation data show that the model performs very well, with a loss value of about 0.0217. This indicates that the model very accurately modelled the validation data. In addition, the accuracy reached 98%, indicating that the model performed very well in classifying the data. This indicates that the model is very effective at detection. The accuracy and loss values are shown in the following visualization graph:

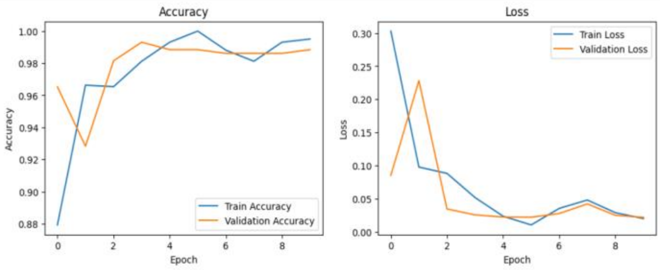

Figure 12. Visualization of accuracy and loss value

The next step is to evaluate the model using a matrix that provides an overview of the model's performance in the classification task. Precision, recall, and F1-Score all have values of around 51%, which indicates that the model performs reasonably well in classification. The following are the values:

	precision	recall	f1-score	support
DarkRoasting	0.51	0.51	0.51	216
LightRoasting	0.51	0.51	0.51	216
accuracy			0.51	432
macro avg	0.51	0.51	0.51	432
weighted avg	0.51	0.51	0.51	432

Figure 13. Matrix Evaluation Model

3.8 Model Detection On The New Data

The built model was tested to perform detection on new data. The results show that dark roast and light roast coffee bean images are detected with 99% accuracy. As seen in the image below:

```
1/1 [==============================] - 0s 59ms/step
{'DarkRoasting': 99.999985, 'LightRoasting': 8.004634e-06}
```

Figure 14. Dark roasting class detection

```
1/1 [==============================] - 1s 1s/step
{'DarkRoasting': 0.00019805509, 'LightRoasting': 99.999794}
```

Figure 15. Light roasting class detection

IV. Conclusion

The conclusion of this research is that by using a CNN model that has been trained with Inception V3 to identify the maturity level of roasted coffee beans, we managed to achieve our goal of implementing an alternative technology that improves the effectiveness of the roasted robusta coffee bean classification process, making it better and more efficient.

References

[1] K. Pertanian, “Outlook 2017 Komoditas Pertanian Sub Sektor Perkebunan,” 2019.

[2] H. Hervinaldy, “Strategi Pemerintah Indonesia Dalam Meningkatkan Pendahuluan,” vol. 4, no. 20500, pp. 1–15, 2017.

[3] Saputra, et.al, “Analisis Kualitas Sangraian Biji Kopi Berdasarkan Ekstraksi Fitur Bentuk dan GLCM”, 2023. Jurnal Teknik Informatika Unika ST. Thomas (JTIUST), Volume 08 Nomor 01, Juni 2023, pISSN : 2548-1916, eISSN : 2657-1501

[4] A. Afriliana, Teknologi Pengolahan Kopi. Yogyakarta: Deepublish, 2018.

[5] Joko, N., Lumbanbatu, J., & Sri, R. (2009). Pengaruh Suhu dan Lama Penyangraian Terhadap Sifat Fisik-Mekanis Biji Kopi Robusta. Seminar Nasional Dan Gelar Teknologi PERTETA, 6(2006), 217–225.

[6] Michael,E.H, et.all, 2020. "Design And Design Of An Android-Based Convolutional Neural Network" SAINSBERTEK Scientific Journal. Vol. 1 No. 1-T.INFORMATIKA, September 2020)

[7]. Ashari, YA and Wahyuni, ES. 2022, "Design of a Maturity Level Classification System for Roasted Balinese Arabica Coffee Beans image-based”. Islamic University of Indonesia.

[8] H. T. Putra, “Berdasarkan Hasil Roasting Menggunakan Algoritma Fuzzy C - Means Skripsi Oleh : Hayuangga Tinno Putra Kusuma Jurusan Teknik Informatika,” 2018.

[9] A.D.S.Putra, et.all. (2020). Penentuan Level Kematangan Kopi Berdasarkan Hasil Roasting Menggunakan Metode Deteksi Rgb Dan Klasifikasi Minimum Distance. Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA). Jakarta-Indonesia, 14 Agustus 2020

[10] Rismiyati and SN. Azhari, “Convolutional Neural Network implementation for imagebased Salak sortation,” ICST (International Conference on Science and TechnologyComputer, 27-28 Oct. 2016 [Online]. Available:https://ieeexplore.ieee.org/document/7877351.

[11] Juliansyah, L., Agustian, I dan Hadi, F., 2018. Deteksi Gangguan Ginjal Melaui Citra Digital Iris Mata Menggunakan Metode Convolutional Neural Network (CNN). Jurnal Amplifier Vol. 9 No. 1, Mei 2019 hal 1-9