

## Klasifikasi Jenis Buah-Buahan Menggunakan Citra Digital Dengan Metode Convolutional Neural Networks

Yuwan Jumaryadi<sup>1</sup>, Alif Muhammad Ihsan<sup>2\*</sup>, Bagus Priambodo<sup>2</sup>

<sup>1</sup> Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Mercu Buana, Jakarta, Indonesia

<sup>2</sup> Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Mercu Buana, Jakarta, Indonesia

Email: <sup>1</sup>yuwan.jumaryadi@mercubuana.ac.id, <sup>2\*</sup>41519210017@student.mercubuana.ac.id, <sup>3</sup>bagus.priambodo@mercubuana.ac.id

Email Penulis Korespondensi: 41519210017@student.mercubuana.ac.id

**Abstrak**—Dampak pandemi COVID-19 sangat terasa pada perubahan pola konsumsi masyarakat. Masyarakat dan pola konsumsi tidak dapat dipisahkan sangat berpengaruh pada kualitas hidup masyarakat. Pola konsumsi masyarakat yang meningkat pada masa pandemi yakni buah-buahan. Buah-buahan merupakan makanan kaya vitamin yang dibutuhkan tubuh untuk membangun imunitas pada masa pandemi ini. Jenis buah dikelompokkan menjadi 2, yaitu buah kering dan buah berdaging. Untuk mempermudah konsumen dalam mengetahui citra data jenis buah tersebut menggunakan metode CNN. Convolutional Neural Network (CNN) adalah algoritma deep learning yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara. Berdasarkan hasil klasifikasi yang telah dilakukan didapatkan nilai yaitu Precision sebesar 94% dan 85%, Recall sebesar 85% dan 95%, F1-score sebesar 90%, Accuracy sebesar 90%, Macro Average sebesar 90%, dan Weighted Average sebesar 90%.

**Kata Kunci:** Convolutional Neural Network; Deep Learning; Image Processing; Classification

**Abstract**—The impact of the COVID-19 pandemic has been felt in changes in consumption patterns. Society and consumption patterns are inseparable, greatly people's consumption patterns that increase during the pandemic are fruits. Fruits are foods rich in vitamins that the body needs to build immunity during this pandemic. Types of fruits are grouped into 2, namely dried fruits and fleshy fruits. To make it easier for consumers to find out the image data of these types of fruit using the CNN method. Convolutional Neural Network is a deep learning algorithm designed to process data in two-dimensional form, for example images or sounds. Based on the classification results that have been carried out obtained values, namely Precision of 94% and 85%, Recall of 85% and 95%, F1-score of 90%, Macro Average of 90%, and Weighted Average of 90%, Weighted Average of 90%.

**Keywords:** Convolutional Neural Network; Deep Learning; Image Processing; Classification

### 1. PENDAHULUAN

Pandemi Covid-19 yang tidak kunjung menurun memberi dampak amat besar pada pola konsumsi masyarakat. Masyarakat dan pola konsumsi tidak dapat dipisahkan karena apa yang dikonsumsi sehari-hari akan berpengaruh pada kualitas hidup Masyarakat [1]. Salah satu pola konsumsi yang marak terjadi yakni pola konsumsi buah-buahan yang merupakan salah satu makanan yang kaya akan vitamin yang dibutuhkan oleh tubuh untuk membangun imunitas pada masa pandemi ini. Sejak merebaknya pandemi di Indonesia, keputusan konsumen dalam memilih jenis buah tentunya tergantung akan kebutuhan tubuh, alternatif penguatan imun, dan tentunya kepuasan konsumen [2]. Jenis buah-buahan di Indonesia dikelompokkan menjadi 2, yakni jenis buah kering dan buah berdaging [3]. Umumnya buah kering termasuk ke dalam jenis buah tunggal, yaitu buah yang terbentuk dari satu bunga dengan satu bakal buah dan berisi 1 biji atau lebih. Sedangkan buah berdaging mempunyai daging yang pada umumnya dapat dimakan.

Dalam penelitian ini, penulis menerapkan metode Convolutional Neural Network (CNN) Metode CNN merupakan teknik deep learning yang dapat diterapkan pada citra resolusi tinggi dengan model distribusi nonparametrik dan melakukan proses pembelajaran mandiri untuk pengenalan objek, ekstraksi objek, dan klasifikasi [4][5][6]. Penelitian ini bertujuan mengimplementasikan algoritma CNN untuk melakukan klasifikasi dari objek jenis buah-buahan yang difokuskan pada pembuatan model dalam melakukan ekstraksi fitur dan klasifikasi objek penelitian dengan melakukan pengolahan citra digital (Digital Image Processing). Selain itu, Penelitian ini bertujuan agar mempermudah konsumen untuk mengetahui citra data jenis buah kering dan buah berdaging yang tentunya berbeda dengan memanfaatkan metode Convolutional Neural Network (CNN). Selain itu, konsumen juga dapat memiliki pertimbangan untuk membeli buah-buahan sesuai dengan kebutuhan tubuhnya dengan menentukan jenis vitamin, tingkat kesesuaian tubuh dan dampak positifnya terhadap tubuh yang bertujuan membangun imunitas pada masa pandemi.

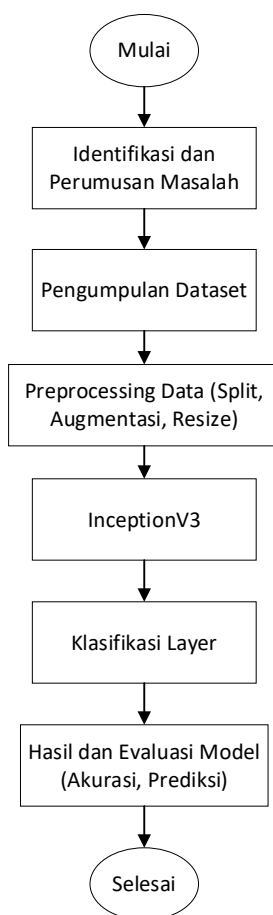
Penelitian ini didukung oleh beberapa penelitian terkait penggunaan metode Convolutional Neural Network (CNN) dalam proses pengenalan objek berdasarkan citra digital yang telah dilakukan sebelumnya, diantaranya penelitian yang dilakukan oleh Dewi dan Ismawan (2021) yang bertujuan merancang sistem pengenalan wajah dengan menggunakan metode Convolutional Neural Network dengan mengaplikasikan library OpenFace untuk memperdalam pengetahuan tentang deep learning dan mengetahui tingkat akurasi yang dihasilkan. Hasilnya adalah dalam melakukan proses pengenalan wajah, algoritma CNN berhasil mendapatkan tingkat akurasi 98% [7]. Selanjutnya penelitian Santony, dkk (2021) menerapkan Convolutional Neural Networks untuk mengklasifikasikan teks pada citra/gambar untuk diterjemahkan dari Bahasa Indonesia ke Bahasa Daerah Minangkabau [8]. Hasilnya adalah penelitian ini telah berhasil mengimplementasikan metode Convolutional Neural Networks (CNN) dalam mengidentifikasi teks pada citra teks dan metode Leveinstan Distance dalam menerjemahkan teks bahasa Indonesia ke dalam teks bahasa daerah. Selanjutnya Yuliany, dkk (2022) melakukan penelitian dengan pembagian data training dan data testing juga penggunaan beberapa parameter yang berfungsi mengurangi masalah overfitting. Pada penelitian tersebut, model pelatihan terbaik memiliki nilai akurasi validasi 83,02% sedangkan pada pengujiannya, hasil akurasi yang didapat yaitu 69,33% [9]. Berdasarkan

beberapa penelitian yang telah dilakukan sebelumnya, maka pada penelitian ini akan mengklasifikasi jenis buah-buahan menggunakan citra digital dengan metode Convolutional Neural Networks.

## 2. METODOLOGI PENELITIAN

Jenis penelitian yang digunakan adalah penelitian terapan. Penelitian terapan ialah setiap penelitian yang bertujuan untuk meningkatkan pengetahuan ilmiah dengan suatu tujuan praktis yang hasilnya diharapkan segera dapat dipakai untuk keperluan praktis [10]. Penelitian ini tidak berfokus pada ide, teori, maupun gagasan, tetapi lebih berfokus kepada penerapan penelitian dalam kehidupan sehari-hari. penelitian yang dilakukan merupakan salah satu solusi untuk menjawab permasalahan para konsumen dalam memudahkan pencarian jenis buah-buahan yang jumlahnya banyak.

Tahapan penelitian merupakan tingkatan dalam penelitian yang digambarkan secara terstruktur, logis dan sistematis [11][12]. Untuk memahami tahapan penelitian yang dilakukan, penelitian ini menggunakan flowchart atau diagram alir. Tahapan penelitiannya menggunakan Applied Research yang menghasilkan solusi dari suatu masalah yang terjadi. Gambaran tahapan penelitian dapat dilihat pada gambar 1.



**Gambar 1.** Tahapan Penelitian

Penjelasan terkait tahapan penelitian pada gambar 1, dapat dijelaskan sebagai berikut:

a. Identifikasi dan Perumusan Masalah

Identifikasi masalah merupakan langkah awal yang sangat penting dalam suatu proses penelitian. Mengidentifikasi masalah dapat dianggap sebagai Langkah awal dalam penelitian yang dilakukan [13].

b. Pengumpulan Dataset

Teknik pengumpulan data yang digunakan dalam penelitian ini, yaitu crawling pada search engine google. Data yang dikumpulkan dalam melakukan penelitian klasifikasi jenis buah-buahan adalah jenis buah kering dan jenis buah berdaging. Dalam melakukan pengumpulan data peneliti menggunakan extensions dari google, yaitu fatkun batch download image.

c. Preprocessing Data

Setelah menyelesaikan tahapan pengumpulan dataset berupa dataset gambar jenis buah kering dan jenis buah berdaging, maka selanjutnya akan dilakukannya tahapan pre-processing data. Pengelolaan citra pada penelitian ini dilakukan dengan cara split dataset menjadi train, val, dan test. Proses pembagian dataset dilakukan untuk memastikan

bahwa model yang telah dibangun dapat bekerja dengan baik pada data yang baru dilihat. Pada tahap selanjutnya adalah melakukan augmentasi data yang merupakan trik umum untuk dapat mengurangi over-fitting. Dataset juga melakukan tahapan resize agar dapat meningkatkan akurasi model. Apabila dataset memiliki ukuran yang berbeda tentunya akan menyulitkan dalam menemukan pola yang sama dalam bentuk yang berbeda. Dengan melakukan resize, model akan lebih mudah dalam mempelajari pola yang ada pada data sehingga dapat memberikan hasil yang akurat. Pada penelitian ini proses resize dilakukan agar dapat menyesuaikan model InceptionV3, menjadi 224x224x3.

d. InceptionV3

Tahapan selanjutnya adalah membuat arsitektur dari CNN untuk proses klasifikasi jenis buah kering dan buah berdaging, dengan menggunakan library TensorFlow keras untuk memanggil pre-trained model yang bernama InceptionV3 sebagai konvolusi layer dan pooling layer. Arsitektur InceptionV3 ini memakai beberapa filter pada layer convolutionalnya dan arsitektur ini merupakan sebuah model dari deep convolutional network yang dikembangkan oleh google untuk mengikuti ILSRVC (ImageNet Large Visual Recognition Challenge) di tahun 2012 [14].

InceptionV3 dikembangkan berdasarkan arsitektur Convolutional Neural Network (CNN) [15][16]. Dalam arsitektur InceptionV3 terdapat berbagai macam prosedur/langkah dilakukan diantaranya adalah konvolusi, Average Pool, MaxPool, dropout, fully connected, softmax. Pada dasarnya dalam penelitian ini menggunakan langkah yang sama. Namun tidak semua fungsi aktivasi dilakukan.

e. Klasifikasi Layer

Selanjutnya adalah klasifikasi layer yang berfungsi untuk melakukan pooling pengurangan dimensi pada suatu tensor, agar dapat dilakukan proses klasifikasi atau prediksi. Kemudian melakukan teknik dropout layer untuk dapat mengurangi overfitting pada saat train model. Selanjutnya melakukan aktivasi sigmoid untuk melakukan binary classification.

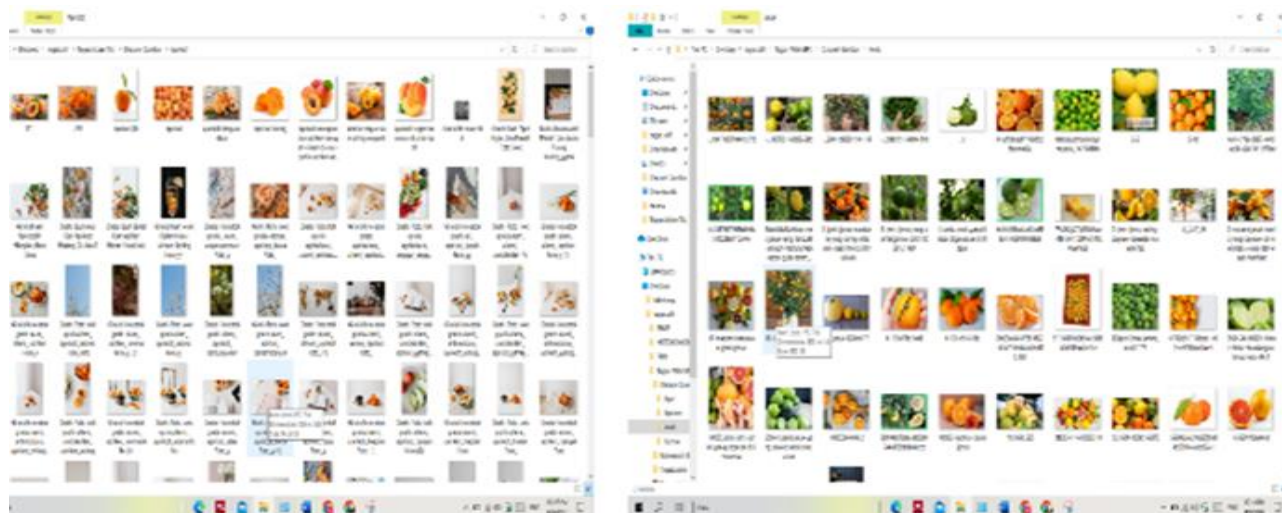
f. Hasil dan Evaluasi Model

Pada tahap terakhir ini akan disimpulkan hasil analisis dan evaluasi terhadap metode klasifikasi CNN yang telah dilakukan. Dan juga mengetahui akurasi data dan visualiasi data setelah melakukan tahapan training.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Dataset

Teknik pengumpulan data yang digunakan dalam penelitian ini, yaitu crawling pada search engine google dan potret langsung dengan kamera smartphone. Data yang dikumpulkan dalam melakukan penelitian klasifikasi jenis buah-buahan adalah jenis buah kering dan jenis buah berdaging. Data yang dikumpulkan berupa data gambar citra jenis buah kering dan buah berdaging yang dikumpulkan dengan cara melakukan crawling pada search engine google dengan extensions dari google, yaitu fatkun batch download image. Dataset tersimpan seperti gambar 2.



**Gambar 2.** Dataset Tersimpan

#### 3.2 Pre-processing

Setelah menyelesaikan proses pengumpulan dataset berupa data citra gambar pada jenis buah kering dan jenis buah berdaging, langkah selanjutnya adalah melakukan tahapan preprocessing image. Dalam proses preprocessing image tahapan-tahapan yang dilakukan adalah sebagai berikut:

a. Pembagian Dataset

Setelah mengumpulkan dataset, data tersebut harus melalui tahapan processing terlebih dahulu. Pre-processing dilakukan dengan cara membagi 3 dataset yang sudah tersedia dengan cara melakukan split Folders menjadi 3 bagian yang diubah menjadi data train, data validasi, dan data testing. Kode program dapat dilihat pada gambar 3.

```
[ ] def buat_folder_split():
import os
import shutil

# Step 1: Tentukan jalur folder dan rasio yang diinginkan
data_folder = dataset_folder # Ganti dengan jalur folder data Anda
output_folder = 'output' # Ganti dengan jalur folder output yang diinginkan
ratios = [0.6, 0.2, 0.2] # Rasio pelatihan, validasi, dan pengujian masing-masing

# Step 2: Dapatkan jumlah gambar di setiap folder
folder_counts = {}
for folder_name in os.listdir(data_folder):
    folder_path = os.path.join(data_folder, folder_name)
    if os.path.isdir(folder_path):
        file_count = len(os.listdir(folder_path))
        folder_counts[folder_name] = file_count

# Step 3: Membuat folder output
training_folder = os.path.join(output_folder, 'Training_Set')
validation_folder = os.path.join(output_folder, 'Validation_Set')
testing_folder = os.path.join(output_folder, 'Testing_Set')

os.makedirs(training_folder, exist_ok=True)
os.makedirs(validation_folder, exist_ok=True)
os.makedirs(testing_folder, exist_ok=True)

# Step 4: Menyalin gambar ke setiap folder yang ditetapkan
for folder_name, count in folder_counts.items():
    folder_path = os.path.join(data_folder, folder_name)
    training_count = int(count * ratios[0])
    validation_count = int(count * ratios[1])
    testing_count = int(count * ratios[2])

    # Menyalin gambar ke set pelatihan
    file_names = os.listdir(folder_path)
    for i in range(training_count):
        file_name = file_names[i]
        src_path = os.path.join(folder_path, file_name)
        dest_path = os.path.join(training_folder, folder_name, file_name)
        os.makedirs(os.path.join(training_folder, folder_name), exist_ok=True)
        shutil.copy(src_path, dest_path)

    # Menyalin gambar ke set validasi
    for i in range(training_count, training_count + validation_count):
        file_name = file_names[i]
        src_path = os.path.join(folder_path, file_name)
        dest_path = os.path.join(validation_folder, folder_name, file_name)
        os.makedirs(os.path.join(validation_folder, folder_name), exist_ok=True)
        shutil.copy(src_path, dest_path)

    # Menyalin gambar ke set pengujian
    for i in range(training_count + validation_count, count):
        file_name = file_names[i]
        src_path = os.path.join(folder_path, file_name)
        dest_path = os.path.join(testing_folder, folder_name, file_name)
        os.makedirs(os.path.join(testing_folder, folder_name), exist_ok=True)
        shutil.copy(src_path, dest_path)

# Step 5: Verifikasi jumlah yang dihitung
training_set_count = sum([len(files) for _, _, files in os.walk(training_folder)])
validation_set_count = sum([len(files) for _, _, files in os.walk(validation_folder)])
testing_set_count = sum([len(files) for _, _, files in os.walk(testing_folder)])

print(f"Training Set: {training_set_count} images")
print(f"Validation Set: {validation_set_count} images")
print(f"Testing Set: {testing_set_count} images")

buat_folder_split()

Training Set: 682 images
Validation Set: 227 images
Testing Set: 229 images
```

Gambar 3. Pembagian Dataset

Adapun dataset yang digunakan dalam penelitian ini sebanyak 1138, dengan data training sebanyak 60%, data validasi sebanyak 20%, dan data testing sebanyak 20%. Pada Tabel 1 merupakan detail mengenai dataset yang digunakan.

Tabel 1. Perbandingan Dataset

Dataset	Training (60%)	Validasi (20%)	Testing (20%)
Kering dan Berdaging	682	227	229
Total		1138	

b. Augmentasi Data

Proses bersamaan selanjutnya yang dilakukan dalam Pre-processing adalah augmentasi data. Proses augmentasi data dilakukan untuk menambah data agar lebih banyak pada saat proses pelatihan. Pada proses ini menggunakan library TensorFlow image data generator. Kode Program dapat dilihat pada gambar 4.

```
# Melakukan Pengolahan Data

# melakukan filter pada pada gambar
def blur_gambar(gambar):
    gambar = cv2.medianBlur(gambar,5)
    return gambar

# Malkukan preprocessing

def preprocessing(training_dataset,validation_dataset,testing_dataset ):

    buat_data_gambar = ImageDataGenerator(
        rescale=1/255,
        vertical_flip=True,
        horizontal_flip=True,
        width_shift_range=0.2,
        height_shift_range=0.2,
        rotation_range=5,
        shear_range=0.02,
        zoom_range=0.02,
        preprocessing_function=blur_gambar
    )
```

Gambar 4. Augmentasi Data

### c. Resize Dataset

Tahapan yang tidak kalah penting pada proses pre-processing adalah resize dataset atau menyamakan ukuran citra jenis buah kering dan jenis buah berdaging. Proses resize bertujuan untuk menyamakan semua ukuran gambar agar proses pelatihan atau training model menjadi lebih cepat dan efisien karna membutuhkan waktu yang lebih sedikit untuk mengelolanya. Setelah di resize semua ukuran gambar berubah menjadi 224x224 agar sesuai dengan input model deep learning. Beberapa model deep learning yang populer seperti ResNet, VGG, dan InceptionV3 telah melakukan train dengan diresize menjadi 224x224 karena dianggap sebagai ukuran standar dataset gambar pada deep learning.

## 3.3 Pembuatan model

### a. Pre-Trained Model

Pada penelitian ini model CNN yang digunakan dalam proses pre-trained model adalah inceptionV3. Arsitektur inceptionV3 ini merupakan pengembangan arsitektur Inception v1 dari GoogLeNet yang dikembangkan lebih lanjut oleh Christian Szegedy dan kawan-kawan [17]. Untuk mengurangi jumlah operasi matematika dalam Arsitektur inceptionV3 memanfaatkan teknik faktorisasi convolution asimetris yang memungkinkan untuk memproses gambar dengan cepat dan efisien. Kode program dapat dilihat pada gambar 5.

```
[ ] import tensorflow as tf
    from tensorflow.keras.applications import MobileNetV3Large,MobileNetV3Small,InceptionV3
    from tensorflow.keras import layers, models
    import datetime

    mobile_net = InceptionV3(input_shape=(tinggi_gambar, luas_gambar, warna), weights='imagenet', include_top=False)
    x = mobile_net.output
    x = layers.Dropout(0.2)(x)
    x = layers.Flatten()(x)
    predictions = layers.Dense(len(nama_class), activation='softmax', kernel_regularizer=tf.keras.regularizers.L2(0.01))(x)

    mobile_net = models.Model(inputs=mobile_net.input, outputs=predictions)
```

Gambar 5. Pre-trained Model

### b. Fully Connected Layer

Lapisan fully connected layer adalah kumpulan dari sebuah proses konvolusi. Lapisan ini mendapatkan input dari proses sebelumnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu [18]. Pada lapisan fully connected layer kita perlu mengimpor library tensorflow lalu memanggil Mobile NET V3 Large, Mobile Net V3 Small, Inception V3, Layers dan Models. Kode program dan keseluruhan layer dapat dilihat pada gambar 6.

```
import tensorflow as tf
from tensorflow.keras.applications import MobileNetV3Large,MobileNetV3Small,InceptionV3
from tensorflow.keras import layers, models
import datetime

mobile_net = InceptionV3(input_shape=(tinggi_gambar, luas_gambar, warna), weights='imagenet', include_top=False)
x = mobile_net.output
x = layers.Dropout(0.2)(x)
x = layers.Flatten()(x)
predictions = layers.Dense(len(nama_class), activation='softmax', kernel_regularizer=tf.keras.regularizers.L2(0.01))(x)

mobile_net = models.Model(inputs=mobile_net.input, outputs=predictions)

# Set the layers to be trainable
for layer in mobile_net.layers[:151]:
    layer.trainable = False
for layer in mobile_net.layers[151:]:
    layer.trainable = True
```

```
[ ] # Melihat Banyak Layer di Model
len(mobile_net.layers)

314

[ ] hasil = [
    TruePositives(name='tp'),
    FalsePositives(name='fp'),
    TrueNegatives(name='tn'),
    FalseNegatives(name='fn'),
    BinaryAccuracy(name='accuracy'),
    Precision(name='precision'),
    Recall(name='recall'),
    AUC(name='auc')
]

learning_rate = 0.001
optimizer = tf.keras.optimizers.legacy.RMSprop(learning_rate=learning_rate)
```

Gambar 6. Pre-trained Model

### 3.4 Pengujian

#### a. Compile Model

Sebelum model CNN dilakukan pelatihan, kita perlu menggunakan metode compile dari library keras yang digunakan untuk melakukan kompilasi dengan menambahkan beberapa parameter dalam menentukan bagaimana model tersebut akan dikompilasi. Kode Program dapat dilihat pada gambar 7.

```
learning_rate = 0.001
optimizer = tf.keras.optimizers.legacy.RMSprop(learning_rate=learning_rate)

# Compile the model within the strategy scope
mobile_net.compile(optimizer=optimizer,
                  loss='categorical_crossentropy',
                  metrics=hasil)
```

Gambar 7. Compile Model

#### b. Proses Training Model

Pada tahapan training model, penelitian ini menggunakan `mobile_net.fit` pada library Tensorflow. Saat method tersebut dipanggil, model akan melakukan iterasi (putaran) melalui data latih yang disediakan sebelumnya lalu memperbarui bobot (parameter) sesuai dengan algoritma. Sebuah model diharapkan mampu memprediksi ataupun mengambil keputusan dengan hasil akurasi yang lebih tinggi daripada sebelumnya setelah proses training selesai. Kode program dapat dilihat pada gambar 8.

```
learning_rate = 0.001
optimizer = tf.keras.optimizers.legacy.RMSprop(learning_rate=learning_rate)

# Compile the model within the strategy scope
mobile_net.compile(optimizer=optimizer,
                  loss='categorical_crossentropy',
                  metrics=hasil)
```

Gambar 8. Training Model

Hasil dari proses training dapat dilihat pada gambar 9 dan tabel 2.

```
Epoch 1/10
11/11 [=====] - 57s 3s/step - loss: 6.0208 - tp: 400.0000 - fp: 280.0000 - tn: 400.0000 - fn: 280.0000 - accuracy: 0.5882 - precision: 0.5882 - recall: 0.5882
Epoch 2/10
11/11 [=====] - 28s 3s/step - loss: 1.2398 - tp: 399.0000 - fp: 281.0000 - tn: 399.0000 - fn: 281.0000 - accuracy: 0.5868 - precision: 0.5868 - recall: 0.5868
Epoch 3/10
11/11 [=====] - 28s 3s/step - loss: 0.7738 - tp: 523.0000 - fp: 157.0000 - tn: 523.0000 - fn: 157.0000 - accuracy: 0.7691 - precision: 0.7691 - recall: 0.7691
Epoch 4/10
11/11 [=====] - 28s 3s/step - loss: 0.4370 - tp: 581.0000 - fp: 99.0000 - tn: 581.0000 - fn: 99.0000 - accuracy: 0.8544 - precision: 0.8544 - recall: 0.8544
Epoch 5/10
11/11 [=====] - 28s 3s/step - loss: 0.3743 - tp: 601.0000 - fp: 79.0000 - tn: 601.0000 - fn: 79.0000 - accuracy: 0.8838 - precision: 0.8838 - recall: 0.8838
Epoch 6/10
11/11 [=====] - 28s 3s/step - loss: 0.2312 - tp: 623.0000 - fp: 57.0000 - tn: 623.0000 - fn: 57.0000 - accuracy: 0.9162 - precision: 0.9162 - recall: 0.9162
Epoch 7/10
11/11 [=====] - 29s 3s/step - loss: 0.1889 - tp: 639.0000 - fp: 41.0000 - tn: 639.0000 - fn: 41.0000 - accuracy: 0.9397 - precision: 0.9397 - recall: 0.9397
Epoch 8/10
11/11 [=====] - 27s 3s/step - loss: 0.1331 - tp: 652.0000 - fp: 28.0000 - tn: 652.0000 - fn: 28.0000 - accuracy: 0.9588 - precision: 0.9588 - recall: 0.9588
Epoch 9/10
11/11 [=====] - 31s 3s/step - loss: 0.2201 - tp: 626.0000 - fp: 54.0000 - tn: 626.0000 - fn: 54.0000 - accuracy: 0.9206 - precision: 0.9206 - recall: 0.9206
Epoch 10/10
11/11 [=====] - 28s 3s/step - loss: 0.0966 - tp: 659.0000 - fp: 21.0000 - tn: 659.0000 - fn: 21.0000 - accuracy: 0.9691 - precision: 0.9691 - recall: 0.9691
```

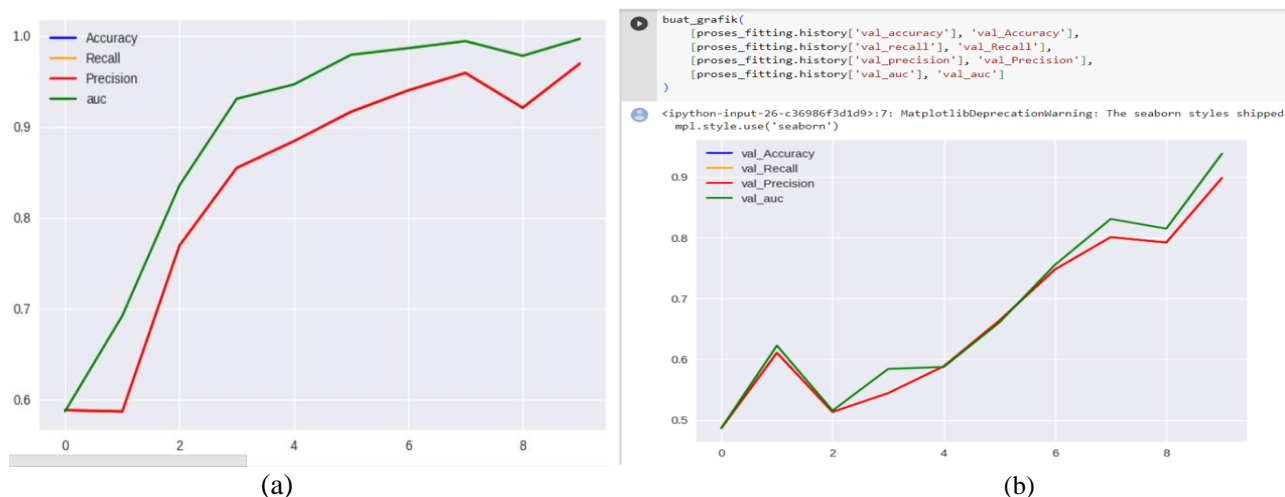
Gambar 9. Hasil Proses Training

Tabel 2. Hasil Akurasi dan Jumlah Epoch

Jumlah Epoch	Parameter	Nilai
10	accuracy	0.9691
10	precision	0.9691
10	Recall	0.9691
10	auc	0.9964

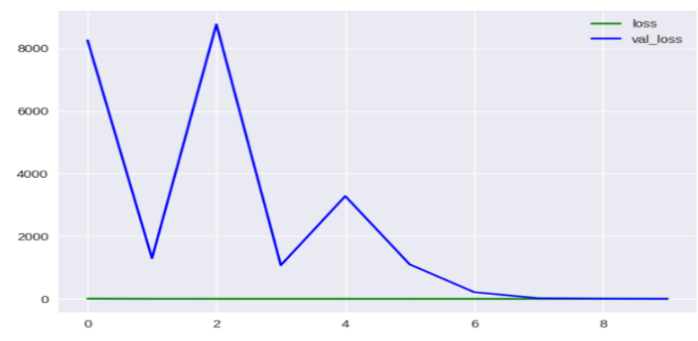
### 3.5 Visualisasi Data

Setelah semua proses training dilakukan, selanjutnya adalah menggambarkan visualisasi datanya. Visualisasi data merupakan proses penggunaan elemen visual seperti diagram, grafik atau peta untuk merepresentasikan data [19]. Pada penelitian ini penggambaran plot dari nilai akurasi menggunakan library matplotlib.pyplot as plt dengan menggunakan parameter proses\_fitting.history yang berisi tentang informasi pelatihan CNN sebelumnya. Grafik hasil akurasi dapat dilihat pada gambar 10 (a) dan 10 (b).



Gambar 10. Grafik Hasil Akurasi

Sedangkan grafik untuk nilai loss selama proses pelatihan dapat dilihat pada gambar 11.



Gambar 11. Grafik Hasil Akurasi

### 3.6 Analisis Hasil

#### a. Mengevaluasi Model

Evaluasi model dilakukan untuk mengetahui kemampuan model dalam mengklasifikasi gambar [20]. Evaluasi model pada penelitian ini menggunakan metode mobile\_net.evaluate. Pada metode ini dataset dibagi menjadi 3 fold yang dalam proses nya training akan dilakukan sebanyak 3 kali putaran, dengan jumlah pada setiap fold terdiri dari 229 data. Untuk hasil terbaik dari percobaan 3 kali putaran adalah seperti gambar 12.

```
3/3 [=====] - 5s 1s/step - loss: 1.0994 - tp: 174.0000 - fp: 18.0000 - tn:
174.0000 - fn: 18.0000 - accuracy: 0.9062 - precision: 0.9062 - recall: 0.9062 - auc: 0.9427
[1.09940767288208,
174.0,
18.0,
174.0,
18.0,
0.90625,
0.90625,
0.90625,
0.942654013633728]
```

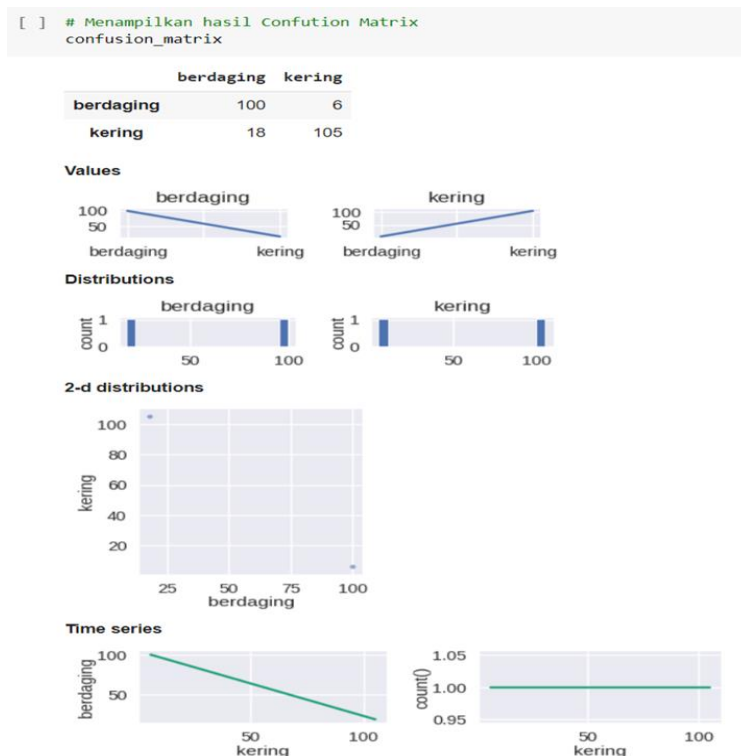
Gambar 12. Hasil Evaluasi Model

#### b. Melakukan Testing Terhadap Model

Testing model pada penelitian ini menggunakan metode Y\_pred = mobile\_net.predict. Dalam prosesnya penelitian ini menggunakan 2 parameter, parameter pertama adalah test\_dataset yang merupakan data yang sudah di training dalam







Gambar 16. Confusion Matrix

f. Hasil Dari Klasifikasi

Setelah melewati beberapa proses dalam pelatihan maka akan mendapatkan hasil dari klasifikasi yang dilakukan. Hasil klasifikasi adalah proses mengelompokkan sesuatu dengan cara melakukan proses membedakan dan mendistribusikan jenis ke dalam kelompok yang berbeda. Hasil dari proses klasifikasi dapat dilihat pada gambar 17.

	precision	recall	f1-score	support
berdaging	0.94	0.85	0.89	118
kering	0.85	0.95	0.90	111
accuracy			0.90	229
macro avg	0.90	0.90	0.90	229
weighted avg	0.90	0.90	0.90	229

Gambar 17. Hasil Klasifikasi

Dari hasil klasifikasi diatas, diperoleh nilai untuk beberapa kategori yaitu:

- Precision: akurasi 2 data yaitu data yang diinginkan dan prediksi yang diberikan model. Diperoleh nilai 94% untuk buah berdaging dan 85% untuk buah kering.
- Recall: gambaran model untuk menemukan informasi. Diperoleh nilai 85% untuk buah berdaging dan 95% untuk buah kering.
- F1-Score: perbandingan nilai precision dan recall. Diperoleh nilai sebesar 90%.
- Accuracy: gambaran seberapa akurat sebuah model mengklasifikasikan data tersedia dengan benar. Diperoleh nilai sebesar 90%.
- Average: untuk menghitung matrix secara bebas dari tiap kelas dan menghitung rata-ratanya. Diperoleh nilai sebesar 90%
- Wighted Average: untuk menghitung rata-rata dengan memperhitungkan bobot datanya. Diperoleh nilai sebesar 90%.

#### 4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan untuk mengklasifikasikan jenis buah berdaging dan buah kering menggunakan metode Convolutional Neural Network berdasarkan citra digital dengan InceptionV3. Pembagian dataset sebesar 60% untuk data training atau sebanyak 682 data citra, 20% untuk data validasi atau sekitar 227 data citra dan 20% untuk data testing atau sekitar 229 data citra. Setelah melakukan proses training model dengan 10 epoch dapat disimpulkan bahwa Convolutional Neural Network (CNN) mampu mengklasifikasikan dan memprediksi pada data citra yang telah diujikan dengan hasil nilai yang dapat dijabarkan menjadi nilai Precision sebesar 94% dan 85%, Recall sebesar

85% dan 95%, F1-score sebesar 90%, Accuracy sebesar 90%, Macro Average sebesar 90%, dan Weighted Average sebesar 90%. Hal tersebut merupakan hasil dari keseluruhan proses pelatihan yang dilakukan.

## REFERENCES

- [1] A. A. Mutezar and Umniy Salamah, "Pengembangan Sistem Manajemen Event Pameran Karya Mahasiswa Menggunakan Metode Extreme Programming," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 809–819, 2021, doi: 10.29207/resti.v5i4.3249.
- [2] Y. R. Jannah, S. Edwina, and A. Rifai, "Analisis Kepuasan Konsumen Berbelanja Buah Di Pasar Modern Saat Pandemi Covid-19 Di Kota Pekanbaru (Studi Kasus Jumbo Mart Delima)," *SEPA J. Sos. Ekon. Pertan. dan Agribisnis*, vol. 20, no. 1, p. 96, 2023, doi: 10.20961/sepa.v20i1.58485.
- [3] I. Q. Lailaty, Y. Suhendri, and M. Efendi, "Seed conservation of Begonia spp. from West Sumatra, Indonesia," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1192, no. 1, 2023, doi: 10.1088/1755-1315/1192/1/012002.
- [4] V. Ayumi and I. Nurhaida, "Klasifikasi Chest X-Ray Images Berdasarkan Kriteria Gejala Covid-19 Menggunakan Convolutional Neural Network," *JSAI (Journal Sci. Appl. Informatics)*, vol. 4, no. 2, pp. 147–153, 2021, doi: 10.36085/jsai.v4i2.1513.
- [5] V. Ayumi, I. Nurhaida, and H. Noprisson, "Implementation of Convolutional Neural Networks for Batik Image Dataset," *Int. J. Comput. Sci. Appl. Math.*, vol. 8, no. 1, p. 5, 2022, doi: 10.12962/j24775401.v8i1.5053.
- [6] R. I. Komaraasih, I. S. Sitanggang, Annisa, and M. A. Agmalaro, "Sentinel-1A image classification for identification of garlic plants using decision tree and convolutional neural network," *IAES Int. J. Artif. Intell.*, vol. 11, no. 4, pp. 1323–1332, 2022, doi: 10.11591/ijai.v11.i4.pp1323-1332.
- [7] N. Dewi and F. Ismawan, "Implementasi Deep Learning Menggunakan Cnn Untuk Sistem Pengenalan Wajah," *Fakt. Exacta*, vol. 14, no. 1, p. 34, 2021, doi: 10.30998/faktorexacta.v14i1.8989.
- [8] M. M. Santoni, N. Chamidah, D. S. Prasvita, H. N. Irmada, R. Astriratma, and R. A. Prayoga, "Penerapan Convolutional Neural Networks untuk Mesin Penerjemah Bahasa Daerah Minangkabau Berbasis Gambar," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1153–1160, 2021, doi: 10.29207/resti.v5i6.3614.
- [9] S. Yuliany, Aradea, and Andi Nur Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *J. Buana Inform.*, vol. 13, no. 1, pp. 54–65, 2022, doi: 10.24002/jbi.v13i1.5022.
- [10] A. Tzanetos and M. Blondin, "A qualitative systematic review of metaheuristics applied to tension/compression spring design problem: Current situation, recommendations, and research direction," *Eng. Appl. Artif. Intell.*, vol. 118, no. October 2022, p. 105521, 2023, doi: 10.1016/j.engappai.2022.105521.
- [11] M. Fasounaki, E. B. Yüce, S. Öncül, and G. Ince, "CNN-based Text-independent Automatic Speaker Identification Using Short Utterances," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 2021, pp. 413–418. doi: 10.1109/UBMK52708.2021.9559031.
- [12] R. Prayoga, A. T. Suci, T. Kristanto, S. Lutfi, - Desyanti, and Y. Pernando, "Kombinasi Metode Multi-Objective Optimization on the Basis of Ratio Analysis (MOORA) dan Pembobotan Rank Order Centroid (ROC) dalam Pemilihan Tablet PC Terbaik," *J. Informatics Manag. Inf. Technol.*, vol. 3, no. 2, pp. 63–69, 2023, doi: 10.47065/jimat.v3i2.256.
- [13] A. Gultom, T. Kristanto, Y. Pernando, J. Kuswanto, N. Putra, and - Amsar, "Penerapan Metode Weighted Product dalam Penyeleksian Supervisor Terbaik," *Bull. Informatics Data Sci.*, vol. 2, no. 1, pp. 42–47, 2023.
- [14] N. Ramadhani, J. Hendryli, and D. E. Herwindianti, "Pencarian Objek Wisata Bersejarah Di Pulau Jawa Menggunakan Convolutional Neural Network," *J. Ilmu Komput. dan Sist. Inf.*, pp. 87–91, 2019.
- [15] N. W. Wulansari, "Implementasi Metode Convolutional Neural Network Untuk Deteksi Kematangan Buah Pisang Menggunakan Inception V3," *JOINS (Journal Inf. Syst.)*, vol. 8, no. 2, pp. 147–155, 2023, doi: 10.33633/joins.v8i2.9074.
- [16] A. H. Nasrullah and H. Annur, "Implementasi Metode Convolutional Neural Network Untuk Identifikasi Citra Digital Daun," *J. Media Inform.*, vol. 7, no. 2, pp. 726–736, 2023, doi: 10.30865/mib.v7i2.5962.
- [17] J. E. Widyaya and S. Budi, "Pengaruh Preprocessing Terhadap Klasifikasi Diabetic Retinopathy dengan Pendekatan Transfer Learning Convolutional Neural Network," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 110–124, 2021, doi: 10.28932/jutisi.v7i1.3327.
- [18] M. H. Romario, E. Ihsanto, and T. M. Kadarina, "Sistem Hitung dan Klasifikasi Objek dengan Metode Convolutional Neural Network," *J. Teknol. Elektro*, vol. 11, no. 2, p. 108, 2020, doi: 10.22441/jte.2020.v11i2.007.
- [19] B. Priambodo, B. Jokonowo, Samidi, A. Ahmad, and R. A. Kadir, "Predict Traffic State Based on PCA-KMeans Clustering of neighbouring roads," in *Advances in Visual Informatics. IVIC 2023*, Selangor: Springer, 2023. doi: 10.1007/978-981-99-7339-2\_36.
- [20] R. Munarto and A. Darma, "Klasifikasi Gender dan Usia Berdasarkan Citra Wajah Manusia Menggunakan Convolutional Neural Network," *J. Ilm. Setrum*, vol. 10, no. 2, pp. 30–43, 2021, doi: 10.36055/setrum.v10i2.12991.
- [21] D. Normawati and S. A. Prayogi, "Implementasi Naive Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J. Sains Komput. Inform.*, vol. 5, no. 2, pp. 697–711, 2021.
- [22] R. Tuntun, K. Kusriani, and K. Kusnawi, "Analisis Perbandingan Kinerja Algoritma Klasifikasi dengan Menggunakan Metode K-Fold Cross Validation," *J. Media Inform. Budidarma*, vol. 6, no. 4, p. 2111, 2022, doi: 10.30865/mib.v6i4.4681.