

Penerapan Teknologi CNN Dalam Proses Pendeteksi Kematangan Buah Stroberi

by Zahrotul Ilmi Wijayanti

Submission date: 01-Jul-2024 02:35PM (UTC+0700)

Submission ID: 2411116717

File name: Uranus_vol_2_no_3_september_2024_hal_01-12.pdf (1.09M)

Word count: 3319

Character count: 21107



Penerapan Teknologi CNN Dalam Proses Pendeteksi Kematangan Buah Stroberi

²¹ Zahrotul Ilmi Wijayanti

Universitas Muhammadiyah Ponorogo

Alamat: Jl. Budi Utomo No.10, Ronowijayan, Kec. Ponorogo, Kabupaten Ponorogo

Korespondensi penulis: zahrotulwijayanti36@gmail.com*

Abstract. *The process of manually identifying fruits to determine ripe and unripe can affect the production and quality of food and beverages themselves. The CNN method is able to group images and analyze images based on objects. Therefore, it is necessary to conduct research using the CNN method on the ripeness of strawberries. This study aims to determine the level of maturity of strawberries during harvest time. The accuracy graph shows that the model is not only capable of learning the training data well but can also generalize well to the validation data. In contrast, the validation accuracy graph starts from 0.825 in the 0th epoch and rises consistently until it reaches 0.975 in the 30th epoch. Both charts remained stable above those values throughout the training period. Overall, the development of the CNN model for the detection of strawberry ripeness resulted in excellent performance. The model achieved the lowest loss of 0.0383 and an accuracy as high as 98% on the validation data, demonstrating a strong ability to accurately predict between ripe and unripe strawberries.*

Keywords: Strawberry Fruit, Image Processing, CNN (Convolution Neural Network)

Abstrak. Proses identifikasi buah secara manual untuk menentukan matang dan tidak matang dapat mempengaruhi pada produksi dan kualitas pada makanan serta minuman itu sendiri. Metode CNN mampu dalam mengelompokkan gambar serta menganalisis gambar berdasarkan objek. Maka perlu dilakukan penelitian yang menggunakan metode CNN pada kematangan buah stroberi. Penelitian ini bertujuan untuk mengetahui tingkat kematangan pada buah stroberi saat waktu pemanenan. Grafik akurasi menunjukkan bahwa model tidak hanya mampu mempelajari data pelatihan dengan baik tetapi juga dapat menggeneralisasi dengan baik untuk data validasi. Sebaliknya, grafik akurasi validasi memulai dari 0.825 pada epoch ke-0 dan naik secara konsisten hingga mencapai 0.975 pada epoch ke-30. Kedua grafik tetap stabil di atas nilai tersebut sepanjang periode pelatihan. Secara keseluruhan, pengembangan model CNN untuk deteksi kematangan buah stroberi menghasilkan performa yang sangat baik. Model mencapai loss terendah sebesar 0.0383 dan akurasi setinggi 98% pada data validasi, menunjukkan kemampuan yang kuat dalam memprediksi dengan tepat antara buah stroberi yang matang dan belum matang.

Kata kunci: Buah Stroberi, Pengolahan Citra, CNN (Convolution Neural Network)

LATAR BELAKANG

Buah Stroberi merupakan buah yang populer dan banyak diminati di Indonesia. Warna buahnya yang merah merona sangat menggoda bagi siapa saja yang melihatnya. Buah stroberi umumnya dapat tumbuh di daerah dataran tinggi contohnya, di daerah Jawa Tengah ada di Kecamatan Tawangmangu Kabupaten Karanganyar. Rasa dari buahnya yang manis membuat sebagian orang ingin menikmatikannya ketika berkunjung ke daerah Tawangmangu Karanganyar (Mahardika et al., 2023).

Kematangan dari buah Stroberi kaya akan manfaatnya karena buah ini dapat dijadikan bahan olahan untuk membuat makanan dan minuman. Makanan dan minuman yang terbuat dari buah stroberi antara lain selai stroberi, pie stroberi, jus stroberi, yogurt stroberi dan masih

Received: Juni 30, 2024; Accepted: Juli 01, 2024; Published: September 30, 2024;

* Zahrotul Ilmi Wijayanti, zahrotulwijayanti36@gmail.com

banyak lagi makanan dan minuman yang terbuat dari buah stroberi. Kematangan menjadi pokok utama dalam membuat olahan berbahan dasar buah stroberi (Puspaningtyas, Yulia, & Farazila, 2022).

Paparan sinar matahari yang tidak merata menjadikan buah stroberi tidak matang secara merata. Olahan buah stroberi ini justru memerlukan buah stroberi yang benar-benar matang secara merata untuk dijadikan bahan dasarnya. Kematangan yang tidak merata menjadikan proses panen buah stroberi membutuhkan waktu yang cukup lama. Proses identifikasi buah secara manual untuk menentukan matang dan tidak matang dapat mempengaruhi pada produksi dan kualitas pada makanan serta minuman itu sendiri.

Seiring dengan kemajuan teknologi pada bidang pertanian, teknologi pengolahan citra digital mampu berfokus dalam manipulasi gambar digital menggunakan komputer. Analisis citra sangat penting dalam proses pengolahan citra digital dikarenakan dapat untuk mengukur dan menganalisis fitur-fitur dalam gambar, seperti bentuk, tekstur, atau warna. CNN (Convolutional Neural Network) menjadi bagian dari pengolahan citra digital dan menjadi salah satu jenis neural network yang biasa digunakan pada data image. CNN mampu untuk mengidentifikasi visual dengan mengenali objek gambar (Lauw, Santoso, & Intan, 2020).

Berdasarkan uraian tersebut, didapatkan bahwa CNN mampu dalam mengelompokkan gambar serta menganalisis gambar berdasarkan objek (Aminullah, 2024). Maka perlu dilakukan penelitian yang menggunakan metode CNN pada kematangan buah stroberi. Penelitian ini bertujuan untuk mengetahui tingkat kematangan pada buah stroberi saat waktu pemanenan. Dengan harapan sistem pendeteksi ini mampu untuk membantu manusia dalam proses pemanenan dan dapat digunakan untuk memilah kematangan berdasarkan warna yang telah terdeteksi oleh sistem menggunakan teknologi CNN.

KAJIAN TEORITIS

Pengolahan citra digital adalah bidang ilmu yang menyelidiki dan menganalisis gambar digital dengan bantuan komputer. Teknologi ini telah berkembang pesat dan digunakan dalam banyak bidang, seperti kesehatan, militer, bisnis, dan pertanian. Pada dasarnya, pengolahan gambar digital terdiri dari beberapa langkah utama: mendapatkan gambar, meningkatkan kualitasnya, mengsegmentasi gambar, mengekstrak fiturnya, dan akhirnya menganalisis gambar (Jumadi, Yupianti, & Sartika, 2021).

Penelitian ini memerlukan rujukan dari penelitian terdahulu. Studi sebelumnya yang ditulis oleh (Areni, Amirullah, & Arifin, 2019) berjudul *Klasifikasi Kematangan Stroberi Berbasis Segmentasi Warna dengan Metode HSV* bertujuan untuk membuat sistem yang dapat

mengklasifikasikan secara otomatis buah stroberi menjadi tiga kategori: belum matang, setengah matang, dan matang. Algoritma Hue, Saturation, Value (HSV) dan Multi-Class Support Vector Machine (SVM) dengan kernel Radial Basic Function (RBF) digunakan untuk mengklasifikasikan kematangan stroberi dan mengsegmentasi warnanya. Data stroberi dapat diambil oleh kamera Logitech C920. Jumlah dataset yang digunakan adalah 158 gambar stroberi. Dengan parameter kernel RBF cost (C) = 10 dan gamma (γ) = 10⁻³, algoritma multiclass SVM memiliki akurasi tertinggi 97% dalam klasifikasi kematangan stroberi.

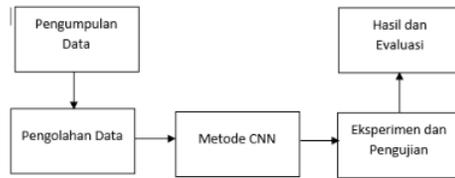
Penelitian sebelumnya juga dilakukan oleh (Hermawan, 2021) dengan judul Klasifikasi Pengenalan Wajah Menggunakan Masker atau Tidak Dengan Mengimplementasikan Metode CNN (Convolutional Neural Network). Dalam penelitian ini, berbagai pendekatan digunakan untuk mendeteksi dan memahami masker wajah. Untuk menentukan apakah metode Convolutional Neural Network (CNN) dapat mengidentifikasi ekspresi wajah menggunakan masker atau tidak, diperlukan tahapan kecermatan, ketelitian, dan kondisi pengoperasian. Kondisi pengoperasian adalah jumlah waktu yang dibutuhkan oleh sistem untuk mengidentifikasi ekspresi wajah. Dataset penelitian memiliki tingkat akurasi sistem sebesar 94%. Penelitian sebelumnya menunjukkan bahwa metode Convolutional Neural Network (CNN) bekerja dengan baik dengan sistem klasifikasi yang menggunakan pengenalan wajah. Metode ini mungkin akan meningkatkan.

(Ananda, Widyasari, Muttaqin, & Stefanie, 2023) melakukan penelitian dengan judul Identifikasi Tingkat Kematangan Buah Pepaya Menggunakan Metode Convolutional Neural Network (CNN). Tujuan penelitian ini adalah untuk memastikan proses pemilihan buah dilakukan secara optimal. Untuk melakukan ini, Deep Learning Convolutional Neural Network (CNN) digunakan untuk mengklasifikasikan pepaya ke dalam tiga kelas: tidak matang (yang berarti buah belum matang), setengah matang (yang berarti buah setengah matang), dan matang. Dataset sebanyak 300 foto buah pepaya digunakan untuk membangun model untuk proses pelatihan dan pengujian. Hasil menunjukkan nilai validasi sebesar 97% dan akurasi sebesar 99%.

METODE PENELITIAN

Metode pada penelitian ini menggunakan metode kuantitatif. Penelitian kuantitatif merupakan teknik pengumpulan data numerik yang kemudian dilakukan analisis pada data numerik sehingga didapat hasil dari temuan (Susanto, Arini, Yuntina, Soehaditama, & Nuraeni, 2024). Sama halnya dengan proses penelitian yang akan dilakukan dengan tahap awal

pengumpulan data lalu analisis data dan terakhir akan mendapatkan hasilnya. Terkait dengan langkah-langkah penelitian dapat dilihat pada gambar 1 berikut :



Gambar 1. Diagram Perancangan Sistem

Diagram perancangan sistem merupakan langkah-langkah yang digunakan dalam proses penelitian yang akan dilakukan dengan sebagai berikut:

1. Pengumpulan Data

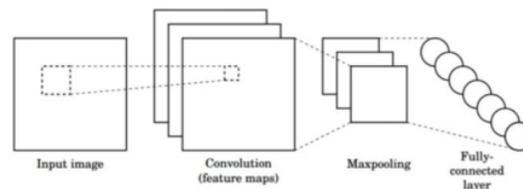
Dalam sebuah merancang sistem diperlukan pengumpulan data yang baik. Data yang berkualitas memungkinkan model membuat prediksi yang akurat dan tetap akan relevan dalam jangka panjang. Tanpa adanya data yang baik sistem tidak akan mencapai performa yang diinginkan atau memberikan nilai signifikan.

2. Pengolahan Data

Melalui tahapan pengolahan data, peneliti dapat memastikan bahwa data yang digunakan untuk analisis memang berkualitas tinggi, relevan dan siap digunakan untuk mendapatkan hasil yang valid dan dapat diandalkan dalam penelitian.

3. Metode CNN

Algoritma deep learning, Convolutional Neural Network (CNN), dapat digunakan untuk memproses input data video dan gambar. CNN dapat membedakan objek satu sama lain dengan mengidentifikasi aspek penting dalam gambar, seperti bias dan bobot yang dapat dipelajari. CNN menggunakan layer konvolusi dan max pooling (Herdianto, 2022).

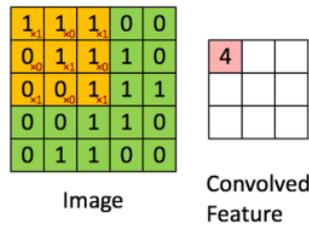


Gambar 2 Convolution Neural Network

Gambar 2 menunjukkan sebuah gambar input dengan menggunakan tiga filter konvolusi untuk mendeteksi fitur dan atributnya. Setelah proses konvolusi selesai, max pooling dilakukan, yang menghasilkan tiga gambar hasil konvolusi yang memiliki resolusi lebih kecil. Terakhir, hasil max pooling dapat dimasukkan ke dalam lapisan MLP yang tersembunyi.

4. Convolution Layer

Sebuah jaringan saraf yang menggunakan atribut dan piksel untuk mengenali objek. Layer convolution membantu mengidentifikasi karakteristik unik sebuah objek. Untuk ilustrasi, sifat mata, telinga, dan hidung membentuk sifat wajah. Konvolusi menggunakan filter pada gambar. Matriks dan area pada gambar diperkalikan selama proses ini (Azmi, Defit, & Sumijan, 2023).

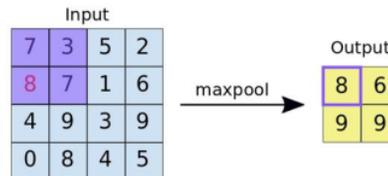


Gambar 3 Operasi convolution layer

Ketika proses konvolusi selesai, output konvolusi dapat digunakan sebagai masukan untuk sebuah MLP, seperti yang ditunjukkan pada gambar 3.

5. Max Pooling

Pooling adalah teknik untuk mengurangi resolusi gambar sambil mempertahankan informasinya. Karena pada praktiknya ada banyak filter yang digunakan dalam proses konvolusi, proses max pooling digunakan. Penggunaan 64 filter pada konvolusi menghasilkan 64 gambar baru, yang membantu mengurangi ukuran dari setiap gambar proses konvolusi (Bimanjaya, Handayani, & Darminto, 2021).



Gambar 4 Operasi Max Pooling

6. Eksperimen dan Pengujian

Uji model sistem sangatlah penting dikarenakan untuk penentu keberhasilan teknik yang diterapkan pada sebuah penelitian. Dirasa dari pengujian ada yang kurang atau tidak sesuai dengan harapan maka dilakukannya eksperimen dalam pengujian sistem untuk membantu dan mendorong agar sistem yang menerapkan CNN sebagai metodenya ini akan memberikan hasil dengan tingkat keberhasilan tinggi.

7. Hasil dan Evaluasi

Tahap terakhir pada perancangan sistem ini adalah tahap hasil dan evaluasi. Hasil akan diperoleh setelah melakukan uji coba pada sistem yang dibuat untuk mengetahui seberapa besar tingkat keberhasilan teknik yang digunakan pada sistem. Evaluasi akan dilakukan setelah ada hasil yang didapat. Evaluasi sendiri dilakukan untuk melihat tingkat keberhasilan sebuah sistem misalnya jika hasil akurasi yang didapat tinggi dan sebanding dengan data yang digunakan maka sistem yang dibuat berhasil sebaliknya, jika hasil akurasi rendah dan tidak sebanding dengan data yang digunakan maka perlu dilakukan perbaikan pada sistem yang dibuat.

HASIL DAN PEMBAHASAN

1. Penghimpunan Data

Dataset dalam setiap penelitian menjadi faktor utama. Dataset bisa berupa gambar, video, maupun angka dalam sebuah penelitian. Dataset yang digunakan dalam penelitian pendeteksi kematangan buah stroberi bersumber dari kaggle yang berjudul “Strawberry Classification”. Jumlah keseluruhan dataset dari kaggle sebanyak 521 data dengan data berupa gambar. Data gambar pada dataset ini berisi gambar buah stroberi yang matang dan belum matang. Jumlah data gambar tersebut termasuk dalam data yang relatif kecil. Klasifikasi data terbagi menjadi dua yakni buah stroberi yang matang dan buah stroberi yang belum matang.



Gambar 5 Dataset buah stroberi matang yang akan digunakan untuk train dan test pada sistem

Dataset yang digunakan pada penelitian ini tidak hanya yang matang saja tetapi yang belum matang akan digunakan. Perlu adanya 2 kelas dalam sebuah sistem untuk menghasilkan tingkat perbandingan dan juga akurasi yang tinggi dan relevan.



Gambar 6 Dataset stroberi yang belum matang dan digunakan untuk train dan test pada sistem.

2. Implementasi CNN

Tahap implementasi ini menjelaskan mekanisme Convolutional Neural Network (CNN) menggunakan Google Colab sebagai code editor. Dalam penerapan CNN ini memanfaatkan library TensorFlow dan Keras untuk membangun model jaringan syaraf tiruan. TensorFlow dan Keras merupakan library Python yang sangat populer dalam pengembangan dan pelatihan model deep learning, karena menyediakan abstraksi tingkat tinggi yang mempermudah pengembang untuk merancang dan melatih CNN.

```
[ ] import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount('/content/drive')

# Set path ke direktori dataset
train_dir = '/content/drive/MyDrive/Classroom/strawberryDataset/training'
val_dir = '/content/drive/MyDrive/Classroom/strawberryDataset/validation'

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=
```

Gambar 7 Mengimport library tensorflow dan keras lalu alokasi path directory

Pada gambar 7 Implementasi ini menunjukkan bagaimana integrasi antara Google Colab sebagai lingkungan pengembangan yang dapat diakses secara online dan TensorFlow-Keras sebagai alat utama untuk mengembangkan model deep learning. Dengan kombinasi ini, peneliti dapat dengan efisien membangun, melatih, dan mengevaluasi model CNN untuk berbagai aplikasi, mulai dari pengenalan pola dalam gambar hingga analisis citra medis yang lebih kompleks. Gambar 7 juga menunjukkan path directory. Pada tahap ini pengaturan jalur direktori sangat penting untuk mengorganisir dan mengelola **data yang digunakan untuk pelatihan dan pengujian model**. Jalur direktori menunjukkan lokasi atau alamat di mana data tersimpan dalam sistem file komputer. Dalam proses pengembangan model machine learning, terutama untuk Convolutional Neural Network (CNN), pengaturan jalur direktori memungkinkan peneliti dengan mudah mengakses dataset yang terdiri dari data yang dipelajari.

```
train_datagen = ImageDataGenerator(  
    [ ]  
    rescale=1./255,  
    rotation_range=20,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    fill_mode='nearest'  
)  
  
val_datagen = ImageDataGenerator(rescale=1./255)  
  
train_generator = train_datagen.flow_from_directory(  
    train_dir,  
    target_size=(150, 150),  
    batch_size=32,  
    class_mode='binary' # Menggunakan binary karena hanya ada dua kelas  
)  
  
val_generator = val_datagen.flow_from_directory(  
    val_dir,  
    target_size=(150, 150),  
    batch_size=32,  
    class_mode='binary'  
)
```

Gambar 8 Augmentasi Dan Preprocessing

Gambar 8 menunjukkan selama proses preprocessing kode ini, "ImageDataGenerator" digunakan untuk meningkatkan data pelatihan dengan berbagai transformasi, seperti mengubah skala piksel menjadi 0-1, rotasi hingga 20 derajat, pergeseran horizontal dan vertikal hingga 20%, shear, zoom, dan flip horizontal. Data validasi hanya di-rescale tanpa peningkatan tambahan. Kemudian, "train_generator" dan "val_generator" mengirimkan data dari direktori dengan ukuran target 150 x 150 piksel, ukuran batch 32, dan mode klasifikasi biner untuk digunakan dalam pelatihan model.

```
===== Total params: 3453121 (13.17 MB)  
Trainable params: 3453121 (13.17 MB)  
Non-trainable params: 0 (0.00 Byte)
```

Gambar 9 Hasil Membangun Model CNN

Pada gambar 9 merupakan hasil dari model CNN yang menunjukkan jumlah parameter total dalam model, termasuk bobot dan bias yang harus diubah selama pelatihan, disebut "total parameter", dan dalam kasus ini, semua 3,453,121 parameter memerlukan 13.17 MB memori. Parameter yang dapat dilatih adalah parameter yang akan diubah selama pelatihan, yaitu semua 3,453,121 parameter dalam model ini. Parameter yang tidak dapat dilatih adalah parameter yang tidak dapat diubah selama pelatihan, dan dalam model ini tidak ada parameter dalam kategori tersebut. Pemahaman tentang persyaratan total, dapat dilatih, dan tidak dapat dilatih membantu dalam mengelola kompleksitas model dan sumber daya komputasi.

```
Epoch 1/100: ..... - loss: 0.8084 - accuracy: 0.9755 - val_loss: 0.8106 - val_accuracy: 0.9765  
Epoch 2/100: ..... - loss: 0.8027 - accuracy: 0.9796 - val_loss: 0.8027 - val_accuracy: 0.9804  
Epoch 3/100: ..... - loss: 0.8028 - accuracy: 0.9794 - val_loss: 0.8043 - val_accuracy: 0.9790  
Epoch 4/100: ..... - loss: 0.8008 - accuracy: 0.9796 - val_loss: 0.8048 - val_accuracy: 0.9824  
Epoch 5/100: ..... - loss: 0.8028 - accuracy: 0.9796 - val_loss: 0.8043 - val_accuracy: 0.9805  
Epoch 6/100: ..... - loss: 0.8028 - accuracy: 0.9775 - val_loss: 0.8085 - val_accuracy: 0.9808  
Epoch 7/100: ..... - loss: 0.8028 - accuracy: 0.9755 - val_loss: 0.8106 - val_accuracy: 0.9805  
Epoch 8/100: ..... - loss: 0.8027 - accuracy: 0.9755 - val_loss: 0.8028 - val_accuracy: 0.9796  
Epoch 9/100: ..... - loss: 0.8028 - accuracy: 0.9801 - val_loss: 0.8106 - val_accuracy: 0.9820  
Epoch 10/100: ..... - loss: 0.8007 - accuracy: 0.9796 - val_loss: 0.8043 - val_accuracy: 0.9790  
Epoch 11/100: ..... - loss: 0.8028 - accuracy: 0.9794 - val_loss: 0.8028 - val_accuracy: 0.9801  
Epoch 12/100: ..... - loss: 0.8008 - accuracy: 0.9805 - val_loss: 0.8106 - val_accuracy: 0.9801  
Epoch 13/100: ..... - loss: 0.8008 - accuracy: 0.9811 - val_loss: 0.8028 - val_accuracy: 0.9824  
Epoch 14/100: ..... - loss: 0.8004 - accuracy: 0.9796 - val_loss: 0.8027 - val_accuracy: 0.9805  
Epoch 15/100: ..... - loss: 0.8002 - accuracy: 0.9827 - val_loss: 0.8048 - val_accuracy: 0.9824
```

Gambar 10 Hasil Training Model CNN

Hasil training model cnn pada gambar 10 menunjukkan pelatihan model dengan data sebanyak 521 sampel dan dilakukan percobaan dengan 30 epoch. Hasil menunjukkan bahwa pada beberapa step, model mencapai loss serendah 0.0462 dan akurasi setinggi 98%. Evaluasi validasi menunjukkan val_loss terendah 0.0383 dengan val_accuracy setinggi 98%. Meskipun ada fluktuasi pada beberapa epoch dengan val_loss mencapai 0.1204 dan val_accuracy serendah 0.9629, model secara keseluruhan menunjukkan performa yang baik dengan akurasi pelatihan dan validasi yang konsisten tinggi. Kinerja yang tinggi ini mengindikasikan bahwa model mampu belajar dengan baik dari data yang diberikan.

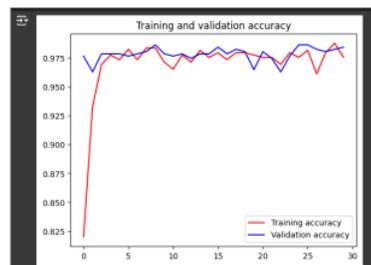
Setelah melatih model CNN, tahap berikutnya dilakukan dengan menggunakan data validasi untuk mengevaluasi kinerja model pada data yang belum pernah dilihat sebelumnya.

```
[ ] loss, accuracy = model.evaluate(val_generator)
print(' Validation Loss: {loss}')
print(' Validation Accuracy: {accuracy}')

17/17 [=====] - 16s 957ms/step - loss: 0.0318 - accuracy: 0.9846
Validation Loss: 0.031753141433000565
Validation Accuracy: 0.984644889831543
```

Gambar 11 Evaluasi Model

Pada gambar 11 Hasil evaluasi menunjukkan bahwa model berkinerja sangat baik dan mampu menggeneralisasi dengan baik pada data validasi dengan menghasilkan kehilangan sebesar 0.0318 dan akurasi sebesar 98%. Hasil dicapai dalam waktu 16 detik, dengan 957 milidetik per langkah evaluasi.



Gambar 12 Grafik Akurasi Model

Grafik akurasi pelatihan gambar 12 menunjukkan bahwa model tidak hanya mampu mempelajari data pelatihan dengan baik tetapi juga dapat menggeneralisasi dengan baik untuk data validasi. Sebaliknya, grafik akurasi validasi memulai dari 0.825 pada epoch ke-0 dan naik secara konsisten hingga mencapai 0.975 pada epoch ke-30. Kedua grafik tetap stabil di atas nilai tersebut sepanjang periode pelatihan.

3. Pengujian Sistem

Tahapan pengujian sistem dalam membangun sistem Convolutional Neural Network (CNN) dimulai dengan menyiapkan dataset representatif untuk pelatihan dan validasi model. Kemudian, proses pelatihan model CNN dilakukan dengan data pelatihan untuk meningkatkan

akurasi dan mengoptimalkan parameter. Setelah itu, model dievaluasi dengan data validasi yang tidak pernah dilihat sebelumnya untuk memastikan bahwa performanya dapat diprediksi secara umum, bukan hanya dengan menggunakan data yang tidak pernah dilihat sebelumnya. Sebelum diimplementasikan dalam aplikasi atau sistem yang sebenarnya, langkah terakhir adalah memperbaiki dan menyetel parameter model untuk meningkatkan kinerja dan luasnya.

```
[ ] import numpy as np
from google.colab import files
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
%matplotlib inline

uploaded = files.upload()

for fn in uploaded.keys():

    path = fn
    img = image.load_img(path, target_size=(150, 150))

    imgplot = plt.imshow(img)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

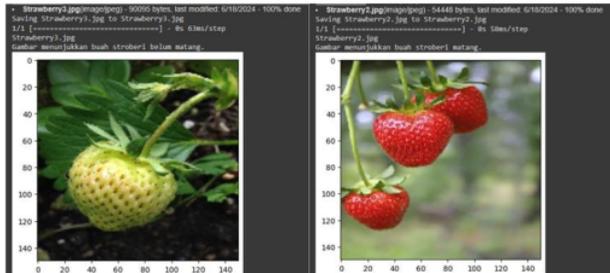
    x /= 255.0

    classes = model.predict(x)

    print(fn)
    if classes[0][0] >= 0.5:
        print('Gambar menunjukkan buah stroberi matang.')
    else:
        print('Gambar menunjukkan buah stroberi belum matang.')
```

Gambar 13 Pengujian Sistem dengan tresholding

Gambar 13 menunjukkan kode untuk menguji sistem deteksi kematangan buah stroberi menggunakan model yang telah dilatih sebelumnya. Pertama, gambar diunggah dan dimuat menggunakan Google Colab. Kemudian, gambar diproses untuk memastikan ukuran yang sesuai (150x150 piksel), ditampilkan menggunakan matplotlib, dan diubah menjadi bentuk array untuk disesuaikan dengan model. Setelah dinormalisasi, model memprediksi kematangan stroberi berdasarkan gambar yang diunggah, dan hasilnya ditampilkan berdasarkan threshold 0.5: jika nilai prediksi lebih besar atau sama dengan 0.5, gambar dianggap menunjukkan buah stroberi matang; jika tidak, dianggap belum matang.



Gambar 14 Prediksi Buah Stroberi Belum Matang dan Sudah Matang

Pada gambar 14 adalah output yang dihasilkan oleh uji coba pada sistem. Sistem berhasil memberikan hasil yang akurat dalam membedakan antara buah stroberi yang belum matang dan matang berdasarkan gambar yang dimasukkan ke dalam sistem. Dengan menggunakan model yang telah dilatih sebelumnya, sistem mampu memproses gambar

tersebut dengan tepat, mengklasifikasikan apakah buah stroberi tersebut matang atau belum matang berdasarkan hasil prediksi yang ditampilkan

KESIMPULAN

Secara keseluruhan, pengembangan model CNN untuk deteksi kematangan buah stroberi menghasilkan performa yang sangat baik. Model mencapai loss terendah sebesar 0.0383 dan akurasi setinggi 98% pada data validasi, menunjukkan kemampuan yang kuat dalam memprediksi dengan tepat antara buah stroberi yang matang dan belum matang. Grafik akurasi menunjukkan bahwa model tidak hanya mempelajari dengan baik dari data pelatihan, tetapi juga mampu menggeneralisasi untuk data validasi. Hasil evaluasi uji coba pada sistem menegaskan bahwa model berhasil dalam membedakan gambar buah stroberi antara matang dan belum matang, memberikan hasil prediksi sesuai dengan yang diharapkan dan dapat diandalkan.

UCAPAN TERIMA KASIH

Bagian ini disediakan bagi penulis untuk menyampaikan ucapan terima kasih, baik kepada pihak penyandang dana penelitian, pendukung fasilitas, atau bantuan ulasan naskah. Bagian ini juga dapat digunakan untuk memberikan pernyataan atau penjelasan, apabila artikel ini merupakan bagian dari skripsi/tesis/disertasi/makalah konferensi/hasil penelitian.

DAFTAR REFERENSI

- Aminullah, M. (2024).¹² Klasifikasi chest X-ray image processing COVID-19 menggunakan metode Convolutional Neural Network dengan arsitektur Visual Geometri Group 16 (VGG 16). *JSR: Jaringan Sistem Informasi Robotik*, 8(1), 68–72.
- Ananda, T. P., Widyasari, S. V., Muttaqin, M. I., & Stefanie, A. (2023).¹ Identifikasi tingkat kematangan buah pepaya menggunakan metode Convolutional Neural Network (CNN). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(3), 2094–2097.
- Areni, I. S., Amirullah, I., & Arifin, N. (2019).¹⁵ Klasifikasi kematangan stroberi berbasis segmentasi warna dengan metode HSV. *Jurnal Penelitian Enjiniring*, 23(2), 113–116.
- Azmi, K., Defit, S., & Sumijan, S. (2023).⁶ Implementasi Convolutional Neural Network (CNN) untuk klasifikasi batik tanah liat Sumatera Barat. *Jurnal Unitek*, 16(1), 28–40.
- Bimanjaya, A., Handayani, H. H., & Darminto, M. R. (2021).⁷ Ekstraksi tapak bangunan dari orthophoto menggunakan model Mask R-CNN (Studi kasus: Kelurahan Darmo, Kota Surabaya). *Jurnal Teknik ITS*, 10(2), C198–C203.

- 19 Herdianto, H. (2022). Klasifikasi objek menggunakan metode Convolutional Neural Network (CNN). *SNASTIKOM*, 1(01), 330–336.
- 11 Hermawan, E. (2021). Klasifikasi pengenalan wajah menggunakan masker atau tidak dengan mengimplementasikan metode CNN (Convolutional Neural Network). *Jurnal Industri Kreatif Dan Informatika Series (JIKIS)*, 1(1), 33–43.
- 10 Jumadi, J., Yupianti, Y., & Sartika, D. (2021). Pengolahan citra digital untuk identifikasi objek menggunakan metode hierarchical agglomerative clustering. *JST (Jurnal Sains Dan Teknologi)*, 10(2), 148–156.
- 14 Lauw, K. O., Santoso, L. W., & Intan, R. (2020). Identifikasi jenis anjing berdasarkan gambar menggunakan Convolutional Neural Network berbasis Android. *Jurnal Infra*, 8(2), 37–43.
- 5 Mahardika, I. K., Bektiarso, S., Santoso, R. A., Novit, A., Saiyendra, R. B., & Dewi, R. K. (2023). Analisis peran suhu pada pertumbuhan dan perkembangan tanaman stroberi. *Phydagogic: Jurnal Fisika Dan Pembelajarannya*, 5(2), 86–91.
- 8 Puspaningtyas, M., Yulia, E., & Farazila, F. R. (2022). Pemanfaatan buah stroberi dalam rangka menunjang pengembangan produk pangan di Desa Pandanrejo. *Jumat Pertanian: Jurnal Pengabdian Masyarakat*, 3(2), 92–95.
- 13 Susanto, P. C., Arini, D. U., Yuntina, L., Soehaditama, J. P., & Nuraeni, N. (2024). Konsep penelitian kuantitatif: Populasi, sampel, dan analisis data (sebuah tinjauan pustaka). *Jurnal Ilmu Multidisiplin*,

Penerapan Teknologi CNN Dalam Proses Pendeteksi Kematangan Buah Stroberi

ORIGINALITY REPORT

22%

SIMILARITY INDEX

21%

INTERNET SOURCES

5%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1	ejournal.itn.ac.id Internet Source	2%
2	cot.unhas.ac.id Internet Source	2%
3	maryamsejahtera.com Internet Source	2%
4	www.bagiteknologi.com Internet Source	1%
5	repo.undiksha.ac.id Internet Source	1%
6	journal.lembagakita.org Internet Source	1%
7	ojs.uma.ac.id Internet Source	1%
8	repository.ukwms.ac.id Internet Source	1%
9	Submitted to University of Mary Student Paper	1%

10	Submitted to University of Muhammadiyah Malang Student Paper	1 %
11	elibrary.nusamandiri.ac.id Internet Source	1 %
12	ojsamik.amikmitragama.ac.id Internet Source	1 %
13	al-afkar.com Internet Source	1 %
14	eprints.umk.ac.id Internet Source	1 %
15	www.researchgate.net Internet Source	1 %
16	ejurnal.unisan.ac.id Internet Source	1 %
17	Submitted to Sriwijaya University Student Paper	<1 %
18	doaj.org Internet Source	<1 %
19	prosiding.snastikom.com Internet Source	<1 %
20	journal.widyakarya.ac.id Internet Source	<1 %
21	eprints.umpo.ac.id Internet Source	<1 %

<1 %

22

www.gamelab.id

Internet Source

<1 %

23

www.tib.eu

Internet Source

<1 %

24

pt.scribd.com

Internet Source

<1 %

25

repositori.uma.ac.id

Internet Source

<1 %

26

repository.radenfatah.ac.id

Internet Source

<1 %

27

search.trdizin.gov.tr

Internet Source

<1 %

28

e-journals.unmul.ac.id

Internet Source

<1 %

29

hostjournals.com

Internet Source

<1 %

30

journal.unisan.ac.id

Internet Source

<1 %

31

www.nu.or.id

Internet Source

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off