

SENASTIKA Universitas Malikussaleh

PENDETEKSIAN KEMATANGAN MANGGA BERBASIS FITUR ANALISIS WARNA DENGAN METODE CNN

Adetia Irvanda¹, Rianda Sulthan², Jihan Adila³, Rizqi Ananda⁴, Munirul Ula⁵

Teknik Informatika, Universitas Malikussaleh, Lhokseumawe, Aceh
e-mail: ¹adetia.210170166@mhs.unimal.ac.id, ²rianda.210170150@mhs.unimal.ac.id,
³jihan.210170147@mhs.unimal.ac.id, ⁴jihan.2101701475@mhs.ac.id, ⁵munirulula@mhs.ac.id.

Abstrak

Deteksi kematangan mangga yang akurat sangat penting dalam industri pertanian untuk menjamin kualitas buah yang optimal. Peninjauan manual oleh karyawan seringkali tidak konsisten dan memakan waktu. Oleh karena itu, penelitian ini mengusulkan penggunaan teknik Convolutional Neural Network (CNN) dan OpenCV untuk mengetahui tingkat kematangan buah mangga berdasarkan analisis warna kulit. CNN digunakan untuk mendeteksi pola visual pada gambar mangga dan melakukan klasifikasi kematangan, sedangkan OpenCV berperan dalam preprocessing gambar seperti konversi ruang warna dan ekstraksi fitur warna. Sistem ini dikembangkan dengan menggunakan bahasa pemrograman Python. Berdasarkan hasil penelitian, metode ini dapat mendeteksi kematangan buah mangga dengan akurasi yang tinggi, sehingga dapat diterapkan pada skala industri untuk meningkatkan efisiensi dan akurasi pemilahan buah otomatis.

Kata Kunci : analisis warna, *Convolutional Neural Network* (CNN), kematangan mangga, OpenCV, pengolahan citra.

Abstract

Accurate detection of mango ripeness is crucial in the agricultural industry to ensure optimal fruit quality. Manual inspection by employees is often inconsistent and time-consuming. Therefore, this study proposes the use of Convolutional Neural Network (CNN) and OpenCV techniques to determine the ripeness level of mangoes based on skin color analysis. CNN is utilized to detect visual patterns in mango images and classify ripeness, while OpenCV plays a role in image preprocessing, such as color space conversion and feature extraction. This system is developed using the Python programming language. Based on the research results, this method can accurately detect mango ripeness, making it suitable for industrial-scale applications to enhance the efficiency and accuracy of automated fruit sorting.

Keywords : color analysis, *Convolutional Neural Network* (CNN), mango ripeness, OpenCV, image processing.

1. PENDAHULUAN

Masyarakat Indonesia sudah sangat familiar dengan buah mangga musiman. Buah yang berasal dari India ini memiliki jenis buah yang sangat digemari oleh masyarakat dari segala usia. Tanaman mangga tumbuh dengan baik pada berbagai kondisi geografis, termasuk dataran rendah dan beberapa iklim panas dapat tumbuh hingga ketinggian 600 meter di atas permukaan laut. Pohon mangga memiliki batang yang tegak dan bercabang kuat, serta ditandai dengan kulit batang yang tebal, kasar, serta berlempengan dan berlekukan kecil yang memberikan tekstur khas pada batang dan daunnya. Ketika buah mangga sudah matang, terkadang terdapat perbedaan antara warna kulit yang matang dengan rasa buah yang belum sepenuhnya manis. Sebaliknya, ada juga buah mangga yang belum matang namun memiliki rasa yang manis. Kematangan buah mangga sangat berpengaruh terhadap kualitas, rasa, dan nilai jualnya di pasar. Mangga yang sudah matang memiliki tekstur, rasa, dan aroma yang disukai oleh konsumen, sedangkan mangga yang belum matang cenderung keras dan asam.

Dalam industri pertanian, terutama pada distribusi dan pemasaran buah, penting untuk memastikan bahwa buah yang dijual sudah mencapai tingkat kematangan yang tepat. Namun, penilaian kematangan buah sering kali dilakukan secara manual oleh tenaga kerja berpengalaman, yang menggunakan penilaian visual dan pengalaman untuk menilai tingkat kematangan mangga. Proses ini membutuhkan waktu dan rentan terhadap ketidakakuratan akibat faktor subjektif dan variasi kondisi pencahayaan. Perubahan warna kulit adalah indikator utama untuk menilai kematangan buah mangga. Buah mangga mengalami perubahan warna yang signifikan selama proses pematangan. Kulitnya berubah dari hijau menjadi kuning atau oranye, tergantung pada varietasnya.

Oleh karena itu, analisis warna pada kulit mangga dapat digunakan sebagai parameter penting dalam menentukan tingkat kematangan buah tersebut. Dalam konteks ini, pengolahan citra digital telah menjadi alat yang efektif untuk mengotomatiskan penilaian kematangan mangga. Teknologi pengolahan citra memungkinkan analisis visual yang konsisten dan objektif, membantu mengurangi ketergantungan pada tenaga kerja manual. Seiring dengan kemajuan teknologi kecerdasan buatan, metode Convolutional Neural Network (CNN), jenis jaringan saraf tiruan, menjadi alat efektif dalam mengenali pola dalam gambar. CNN dibuat untuk secara otomatis mempelajari ciri-ciri visual dari data gambar melalui berbagai lapisan konvolusi, sehingga sangat cocok untuk keperluan klasifikasi gambar. CNN memiliki kemampuan untuk mendeteksi pola visual yang kompleks, termasuk perubahan warna pada kulit mangga, yang membuatnya cocok untuk deteksi kematangan buah. Selain CNN, teknologi pengolahan citra berbasis OpenCV juga berperan penting dalam tahap pra-pemrosesan gambar.

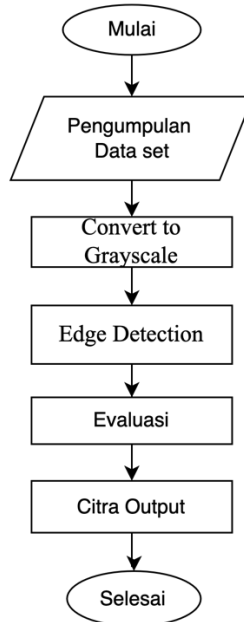
OpenCV merupakan pustaka sumber terbuka yang sering digunakan dalam aplikasi pengolahan citra dan visi komputer. Dengan menggunakan OpenCV, kita dapat menerapkan berbagai teknik pemrosesan citra seperti segmentasi, konversi ruang warna, dan ekstraksi fitur untuk menyiapkan data gambar sebelum dianalisis lebih lanjut oleh CNN. Pra-pemrosesan ini penting untuk memastikan kualitas gambar yang dianalisis oleh CNN dan berisi informasi relevan untuk klasifikasi tingkat kematangan.

Dalam penelitian ini, kami merekomendasikan sistem otomatis deteksi tingkat kematangan mangga berdasarkan analisis warna kulit menggunakan kombinasi CNN dan OpenCV, yang diimplementasikan dengan bahasa pemrograman Python. Python dipilih karena memiliki dukungan luas terhadap pengolahan citra dan pengembangan model pembelajaran mesin, serta komunitas yang aktif yang menyediakan berbagai pustaka dan sumber daya untuk implementasi teknologi ini. Langkah pertama dalam pengembangan sistem kami adalah mengumpulkan gambar mangga dengan berbagai tingkat kematangan. Gambar-gambar ini kemudian diproses menggunakan OpenCV untuk mengekstrak fitur warna yang relevan. Fitur ini kemudian dipergunakan oleh CNN untuk mengkategorikan tingkat kematangan buah ke dalam beberapa kategori, seperti belum masak, setengah matang, dan matang sempurna.

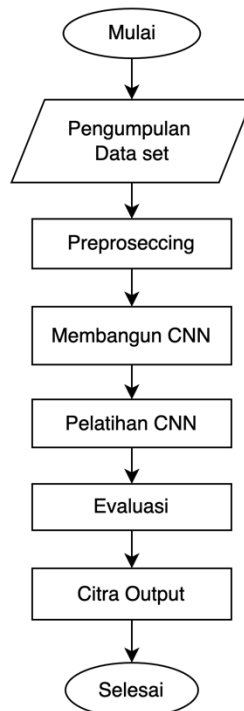
Tujuan utama penelitian ini ialah untuk menyusun metode yang efisien, cepat, dan akurat guna mengenali kematangan mangga secara otomatis, sehingga bisa diterapkan di tingkat industri. Sistem ini diharapkan mampu menggantikan proses penilaian manual yang lambat dan tidak konsisten, sekaligus meningkatkan efisiensi dalam proses seleksi dan sortasi buah di berbagai tahap distribusi dan pemasaran. Diharapkan penerapan teknologi ini dapat meningkatkan kualitas produk pertanian, mengurangi limbah akibat kesalahan penilaian kematangan, dan meningkatkan kepuasan konsumen. Penelitian ini berperan penting dalam menggabungkan deep learning dengan pengolahan citra untuk menciptakan solusi inovatif dalam mendeteksi kematangan buah. Dengan menggunakan CNN dan OpenCV, sistem ini dapat mendeteksi kematangan dengan akurasi tinggi dan dapat diintegrasikan dengan mudah ke dalam proses produksi yang ada, tanpa memerlukan peralatan tambahan yang mahal. Sistem ini memberikan fleksibilitas untuk menyesuaikan model sesuai dengan variasi mangga yang berbeda, sehingga dapat digunakan secara luas di berbagai daerah dan situasi produksi.

2. METODE PENELITIAN

Penelitian ini bertujuan untuk mendeteksi kematangan mangga berdasarkan analisis warna kulit menggunakan metode **Convolutional Neural Network (CNN)** dan **OpenCV**.



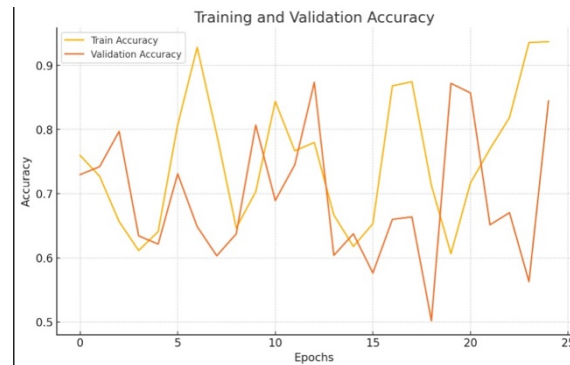
Gambar 1. Flowcart OpenCV



Gambar 2. Flowcart CNN

2.1 Metode Convolutional Neural Network (CNN)

- Grafik Akurasi Training dan Validasi



Gambar 3. Grafik Akurasi Training dan Validasi

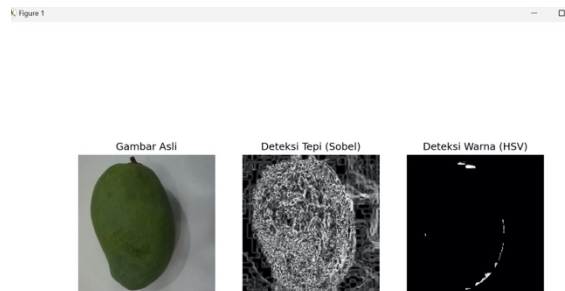
- Akurasi Training menunjukkan seberapa baik model mempelajari pola dari data latih selama setiap epoch.
 - Akurasi Validasi menunjukkan seberapa baik model memprediksi pada data yang tidak dilatih (data validasi).
 - Jika grafik akurasi training meningkat seiring waktu, ini menunjukkan model belajar dengan baik dari data latih.
 - Jika akurasi validasi juga meningkat dan nilainya mendekati akurasi training, ini menunjukkan model memiliki performa yang baik untuk data yang belum pernah dilihat (generalized well).
 - Jika akurasi validasi lebih rendah daripada akurasi training, tetapi tetap meningkat, itu bisa menjadi tanda bahwa model belajar dengan baik tetapi ada sedikit overfitting.
 - Overfitting bisa terjadi jika perbedaan antara akurasi training dan validasi semakin besar di akhir pelatihan, menunjukkan bahwa model terlalu cocok dengan data latih tetapi tidak bekerja baik pada data baru.
- Grafik Loss Training dan Validasi



Gambar 4. Grafik Loss Training dan Validasi

- Loss Training menunjukkan seberapa buruk model membuat prediksi pada data latih (dengan semakin rendah loss, semakin baik modelnya).
- Loss Validasi menunjukkan seberapa buruk model pada data validasi yang tidak digunakan dalam pelatihan.
- Penurunan loss training mengindikasikan bahwa model sedang belajar dan memperbaiki kesalahan pada data latih.
- Loss validasi yang menurun bersama dengan loss training menunjukkan bahwa model tidak hanya belajar dari data latih, tetapi juga mampu menggeneralisasi ke data baru.
- Jika loss validasi berhenti menurun atau malah mulai meningkat sementara loss training terus menurun, itu adalah tanda overfitting, di mana model belajar terlalu detail dari data latih sehingga kehilangan kemampuan generalisasi.

2.2 Metode OpenCV



Gambar 5. Proses Kematangan Mangga

1. Gambar Asli (Kiri)

- Ini adalah gambar asli buah mangga sebelum dilakukan pemrosesan apapun. Gambar ini digunakan sebagai referensi untuk tahap-tahap berikutnya dalam pengolahan citra. Biasanya, gambar asli digunakan sebagai input untuk proses deteksi atau klasifikasi.

2. Deteksi Tepi (Sobel) (Tengah)

- Pada bagian ini, gambar mangga telah diproses menggunakan deteksi tepi Sobel. Teknik ini digunakan untuk mendeteksi kontur atau tepi objek pada gambar berdasarkan perubahan intensitas piksel.

- Deteksi tepi Sobel menghitung turunan pertama gambar dalam arah horizontal dan vertikal untuk menemukan tepi objek. Hasilnya adalah gambar yang menampilkan area dengan perubahan intensitas tajam (seperti kontur buah mangga), di mana objek utamanya (mangga) terlihat dikelilingi oleh garis-garis yang mewakili tepinya.

3. Deteksi Warna (HSV) (Kanan)

- Pada bagian ini, gambar diproses menggunakan model warna HSV (Hue, Saturation, Value) untuk melakukan segmentasi warna. Gambar diubah dari ruang warna RGB menjadi HSV, di mana komponen Hue (corak warna) sering digunakan untuk mendeteksi warna tertentu.

- Kemungkinan besar, warna yang relevan dalam gambar diisolasi (seperti warna hijau dari mangga). Warna yang tidak sesuai dengan rentang deteksi (berdasarkan nilai HSV yang dipilih) ditampilkan sebagai hitam, sementara warna yang sesuai (warna mangga) ditampilkan sebagai area berwarna putih atau abu-abu.

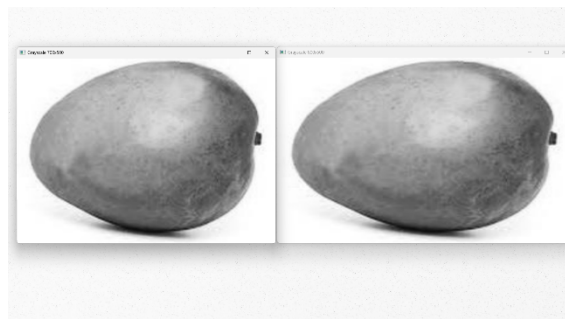
- Gambar asli digunakan sebagai dasar pemrosesan.

- Deteksi tepi Sobel membantu mengidentifikasi batas atau tepi mangga, berguna untuk analisis bentuk atau ukuran.

- Deteksi warna HSV digunakan untuk segmentasi berdasarkan warna, yang mungkin digunakan untuk mengidentifikasi atau memisahkan objek (mangga) dari latar belakang atau objek lainnya.

- Pemrosesan ini biasanya digunakan dalam aplikasi pengolahan citra untuk deteksi objek, klasifikasi, atau segmentasi, khususnya dalam proyek klasifikasi seperti yang Anda lakukan pada buah mangga.

4. Grayscale



Gambar 6. Hasil Grayscale

- Grayscale merupakan Gambar yang tidak memiliki banyak warna melainkan mempunyai 2 warna saja yaitu hitam dan putih. Kedua warna tersebut memiliki 2 kedudukannya masing-masing, misalkan warna hitam mewakili kecerahan dengan insentitas yang paling rendah, sedangkan warna putih mewakili kecerahan yang paling tinggi. Dan dalam istilah grayscale warna hitam atau putih tersebut dijuluki sebagai piksel. Gray sacle

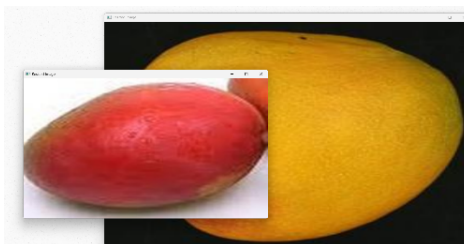
sering digunakan dalam kegiatan fotografi hitam putih yang terdiri dari warna hitam dan putih saja. Dimana kegiatan tersebut hanya membutuhkan insentitas cahaya saja.

- Grayscale bekerja dengan satu channel yaitu 8-bit, yang berarti ada 256 tingkat kecerahan, dari 0 berwarna hitam hingga 255 yang berwarna putih, Grayscale berguna dalam berbagai aplikasi seperti pemrosesan gambar, pengenalan pola, dan desain grafis.

5. Size

Dalam metode OpenCV Di setiap gambar pasti adanya ukuran gambar tersebut yang akan mempengaruhi media yang akan di isi oleh gambar tersebut , nah kita di sini bisa mengakalinya dnegan merubah gambar yang kita inginkan dengan resize atau dalam Bahasa indonesianya merubah ukuan, Berikut kita akan memberikan sedikit metode open cv yang merubah size dalam kodingan yang simpel dan mudah di pahami.

```
import cv2
# Load the image
image = cv2.imread('healthy_002.jpg')
# Check if image is loaded successfully
if image is None:
    print("Error: Unable to load image.")
else:
    # Resize image to 700x500
    resized_image = cv2.resize(image, (1200, 800))
    # Display the resized image
    cv2.imshow('Resized Image', resized_image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```



Gambar 7. Size Buah Mangga

Dan berikut adalah hasil perbedaan gambar yang dapat kita lihat sebelah kiri mangga (merah) dengan ukuran 800X500 P, dan sebelah kanan mangga (kuning) dengan ukuran 1200x800 P. (P) yang dimaksud adalah piksel yaitu Blok dasar membentuk sebuah gambar yang terdiri dari blok - blok kecil.

2.3 Data Set

Dataset pada pendeteksian kematangan mangga berbasis analisis warna dengan metode CNN dan OpenCV pada Python adalah kumpulan information yang terdiri dari gambar-gambar mangga dengan berbagai tingkat kematangan yang digunakan untuk melatih, menguji, dan mengevaluasi demonstrate klasifikasi berbasis Convolutional Neural Arrange (CNN). Setiap gambar dalam dataset ini diambil dari mangga pada tahap kematangan yang berbeda, seperti mentah, setengah matang, dan matang, dan masing-masing gambar diberi name yang sesuai.

Dataset ini merupakan elemen penting dalam penelitian karena berfungsi sebagai sumber information utama bagi CNN untuk mempelajari pola visual yang berkaitan dengan kematangan mangga. Perubahan warna kulit mangga dari hijau ke kuning atau oranye selama proses pematangan menjadi faktor visual yang signifikan dalam menentukan tingkat kematangan. Dataset yang baik harus mencakup variasi dalam hal jenis mangga, sudut pengambilan gambar, serta kondisi pencahayaan, agar demonstrate CNN dapat mengenali pola kematangan dengan lebih baik.

Tabel 1. Detail Dataset

Jenis Sample	Jumlah Data	Sumber Data
Mangga masak	100	Kaggle
Mangga muda	100	Kaggle

Mangga busuk	100	Kaggle
--------------	-----	--------

3. HASIL DAN PEMBAHASAN

3.1 Operator OpenCV

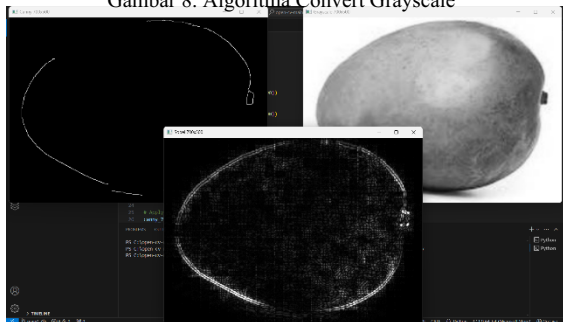
Dalam tahap ini, gambar mangga diproses terlebih dahulu menggunakan teknik *OpenCV* seperti konversi warna, deteksi tepi (*Sobel*), dan deteksi warna (*HSV*). Deteksi tepi membantu memisahkan batas objek mangga dari latar belakang, sementara deteksi warna digunakan untuk memisahkan warna kulit mangga yang menunjukkan tingkat kematangan.

```

1 import cv2
2 import numpy as np
3
4 # Load image
5 image = cv2.imread('healthy_066.jpg')
6
7 # Resize image to 700x500
8 resized_700x500 = cv2.resize(image, (700, 500))
9
10 # Resize image to 800x500
11 resized_800x500 = cv2.resize(image, (800, 500))
12
13 # Convert to grayscale
14 gray_700x500 = cv2.cvtColor(resized_700x500, cv2.COLOR_BGR2GRAY)
15 gray_800x500 = cv2.cvtColor(resized_800x500, cv2.COLOR_BGR2GRAY)
16
17 # Apply Sobel operator
18 sobel_700x500 = cv2.Sobel(gray_700x500, cv2.CV_64F, 1, 1, ksize=5)
19 sobel_800x500 = cv2.Sobel(gray_800x500, cv2.CV_64F, 1, 1, ksize=5)
20
21 # Convert sobel results to uint8
22 sobel_700x500 = cv2.convertScaleAbs(sobel_700x500)
23 sobel_800x500 = cv2.convertScaleAbs(sobel_800x500)
24
25 # Apply Canny edge detection
26 canny_700x500 = cv2.Canny(gray_700x500, 100, 200)
27 canny_800x500 = cv2.Canny(gray_800x500, 100, 200)
28
29 # Display the images
30 cv2.imshow('Grayscale 700x500', gray_700x500)
31 cv2.imshow('Sobel 700x500', sobel_700x500)
32 cv2.imshow('Canny 700x500', canny_700x500)
33
34 cv2.imshow('Grayscale 800x500', gray_800x500)
35 cv2.imshow('Sobel 800x500', sobel_800x500)
36 cv2.imshow('Canny 800x500', canny_800x500)
37
38 cv2.waitKey(0)
39 cv2.destroyAllWindows()

```

Gambar 8. Algoritma Convert Grayscale



Gambar 9. Hasil Convert Grayscale

Gambar di atas merupakan hasil pemrosesan citra menggunakan OpenCV dengan beberapa tahapan, termasuk deteksi tepi dan konversi ke grayscale. Tahapan ini bertujuan untuk mengekstraksi kontur dan fitur visual dari objek mangga. Hasilnya, kontur mangga terdeteksi jelas pada latar belakang hitam, dan citra grayscale memperjelas struktur objek. Pemrosesan ini dapat digunakan sebagai dasar dalam pengembangan model klasifikasi kematangan mangga menggunakan CNN.

CNN diterapkan untuk mengenali pola visual dari gambar mangga dan mengklasifikasinya ke dalam beberapa kategori, seperti belum matang, setengah matang, dan matang sempurna. CNN secara efektif mempelajari pola perubahan warna kulit dari dataset gambar mangga yang dikumpulkan. Model CNN yang dilatih menggunakan dataset yang relevan menunjukkan kemampuan generalisasi yang baik, dengan akurasi yang cukup tinggi dalam memprediksi kematangan buah dari gambar baru yang tidak dilatih.

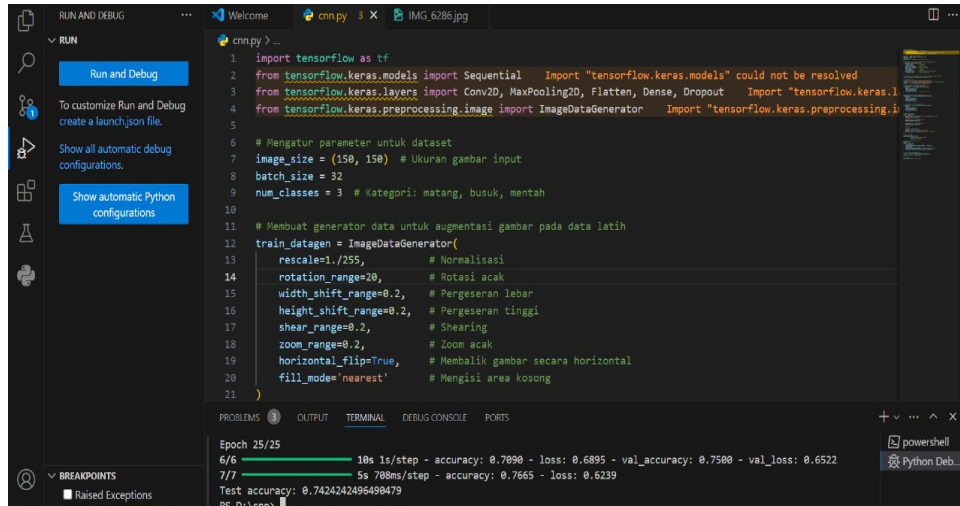
Sebelum citra dianalisis lebih lanjut, beberapa gambar dikonversi menjadi *grayscale* untuk menyederhanakan proses analisis dengan menghilangkan informasi warna. Teknik ini memungkinkan pemrosesan berbasis intensitas cahaya, di mana setiap piksel hanya mewakili tingkat kecerahan dari hitam hingga putih.

3.2 Convolutional Neural Network (CNN)

Hasil dari pelatihan model Convolutional Neural Network CNN ditunjukkan dengan grafik akurasi dan loss pada data latih dan validasi. menggunakan TensorFlow dan Keras untuk klasifikasi tingkat kematangan buah mangga (matang, busuk, dan mentah). Ukuran gambar input ditetapkan sebesar 150x150 piksel dengan batch size 32 dan tiga kelas output. Augmentasi data diterapkan pada data latih menggunakan **Image Data Generator**, yang mencakup rotasi, pergeseran, zoom, dan flip horizontal untuk meningkatkan variasi dan

8 SENASTIKA 2024, Jurusan Informatika Universitas Malikussaleh

mengurangi overfitting. Data latih, validasi, dan uji diperoleh dari direktori gambar menggunakan metode `flow_from_directory`. Arsitektur CNN yang digunakan terdiri dari tiga lapisan konvolusi berukuran 32, 64, dan 128 filter, diikuti oleh lapisan **max pooling** untuk ekstraksi fitur, kemudian diratakan dengan lapisan **Flatten** dan diakhiri dengan lapisan **fully connected** yang menggunakan aktivasi **softmax** untuk klasifikasi multi-kelas. Model dikompilasi menggunakan **Adam optimizer** dengan **categorical_crossentropy** sebagai fungsi loss, dan dilatih selama 25 epoch dengan validasi pada setiap epoch untuk mengurangi overfitting. Setelah pelatihan, model dievaluasi menggunakan data uji dan hasil akurasi dicetak. Model yang telah dilatih disimpan dalam file.



```
1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
4 from tensorflow.keras.preprocessing.image import ImageDataGenerator
5
6 # Mengatur parameter untuk dataset
7 image_size = (150, 150) # Ukuran gambar input
8 batch_size = 32
9 num_classes = 3 # Kategori: matang, busuk, mentah
10
11 # Membuat generator data untuk augmentasi gambar pada data latih
12 train_datagen = ImageDataGenerator(
13     rescale=1./255, # Normalisasi
14     rotation_range=20, # Rotasi acak
15     width_shift_range=0.2, # Pergeseran lebar
16     height_shift_range=0.2, # Pergeseran tinggi
17     shear_range=0.2, # Shearing
18     zoom_range=0.2, # Zoom acak
19     horizontal_flip=True, # Membalik gambar secara horizontal
20     fill_mode='nearest' # Mengisi area kosong
21 )
22
23 # ... (model building and training code omitted for brevity) ...
24
25 # Output in terminal:
Epoch 25/25
6/6 ----- 10s 1s/step - accuracy: 0.7890 - loss: 0.6895 - val_accuracy: 0.7580 - val_loss: 0.6522
7/7 ----- 5s 780ms/step - accuracy: 0.7665 - loss: 0.6239
Test accuracy: 0.7424242496498479
```

Gambar 10. Hasil Deteksi CNN


```

1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
4 from tensorflow.keras.preprocessing.image import ImageDataGenerator
5
6 # Mengatur parameter untuk dataset
7 image_size = (150, 150) # Ukuran gambar input
8 batch_size = 32
9 num_classes = 3 # Kategori: matang, busuk, mentah
10
11 # Membuat generator data untuk augmentasi gambar pada data latih
12 train_datagen = ImageDataGenerator(
13     rescale=1./255, # Normalisasi
14     rotation_range=20, # Rotasi acak
15     width_shift_range=0.2, # Pergeseran lebar
16     height_shift_range=0.2, # Pergeseran tinggi
17     shear_range=0.2, # Shearing
18     zoom_range=0.2, # Zoom acak
19     horizontal_flip=True, # Membalik gambar secara horizontal
20     fill_mode='nearest' # Mengisi area kosong
21 )
22
23 # Generator data untuk validasi dan test (tanpa augmentasi, hanya normalisasi)
24 valid_test_datagen = ImageDataGenerator(rescale=1./255)
25
26 # Direktori dataset
27 train_dir = 'D:\\enn\\mangga_dataset\\mangga_image\\train'
28 valid_dir = 'D:\\enn\\mangga_dataset\\mangga_image\\valid' # Direktori validasi
29 test_dir = 'D:\\enn\\mangga_dataset\\mangga_image\\test'
30
31 # Membuat generator data untuk dataset latih
32 train_generator = train_datagen.flow_from_directory(
33     train_dir,
34     target_size=image_size,
35     batch_size=batch_size,
36     class_mode='categorical' # Kelas lebih dari dua
37 )
38
39 # Membuat generator data untuk dataset validasi
40 valid_generator = valid_test_datagen.flow_from_directory(
41     valid_dir,
42     target_size=image_size,
43     batch_size=batch_size,
44     class_mode='categorical'
45 )
46
47 # Membuat generator data untuk dataset uji
48 test_generator = valid_test_datagen.flow_from_directory(
49     test_dir,
50     target_size=image_size,
51     batch_size=batch_size,
52     class_mode='categorical'
53 )
54
55 # Membangun model CNN
56 model = Sequential([
57     Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
58     MaxPooling2D(pool_size=(2, 2)),
59     Conv2D(64, (3, 3), activation='relu'),
60     MaxPooling2D(pool_size=(2, 2)),
61     Conv2D(128, (3, 3), activation='relu'),
62     MaxPooling2D(pool_size=(2, 2)),
63     Flatten(),
64     Dense(512, activation='relu'),
65     Dropout(0.5), # Untuk mengurangi overfitting
66     Dense(num_classes, activation='softmax') # 3 kelas output
67 ])
68
69 # Kompilasi model
70 model.compile(optimizer='adam',
71               loss='categorical_crossentropy',
72               metrics=['accuracy'])
73
74 # Training model dengan validasi
75 epochs = 25
76 history = model.fit(
77     train_generator,
78     steps_per_epoch=train_generator.samples // batch_size,
79     epochs=epochs,
80     validation_data=valid_generator, # Menggunakan data validasi
81     validation_steps=valid_generator.samples // batch_size
82 )
83
84 # Evaluasi model pada dataset uji
85 test_loss, test_acc = model.evaluate(test_generator)
86 print(f'Test accuracy: {test_acc}')
87
88 # Save model
89 model.save('mangga_cnn_model.keras')

```

Gambar 11. Hasil CNN

3.3 Pemrosesan HSV

Hue Mean (%): Mengacu pada rata-rata warna utama yang diukur dalam lingkaran warna (0 hingga 360 derajat, tetapi dalam tabel ini dikonversi ke dalam bentuk persentase).

Saturation Mean (%): Menunjukkan kemurnian warna, di mana nilai tinggi berarti lebih jenuh atau cerah.

Value Mean (%): Mengacu pada kecerahan gambar, dengan nilai tinggi menunjukkan warna yang lebih cerah.

Status: Menggambarkan tingkat kematangan buah rambutan (Mentah, Hampir Matang, Matang, Busuk).

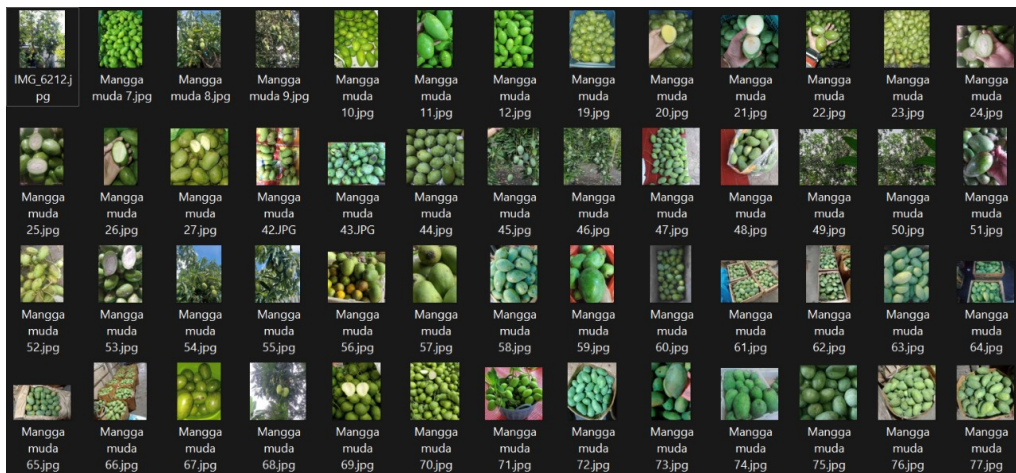
Tabel.1. Dataset HSV

No	Hue mean(%)	Saturation mean(%)	Value mean(%)	Status
1	14.03%	9.39%	70.69%	Mentah
2	12.15%	8.55%	76.31%	Hampir Matang
3	26.55%	6.02%	72.90%	Matang
4	17.29%	2,69%	68.89%	Busuk

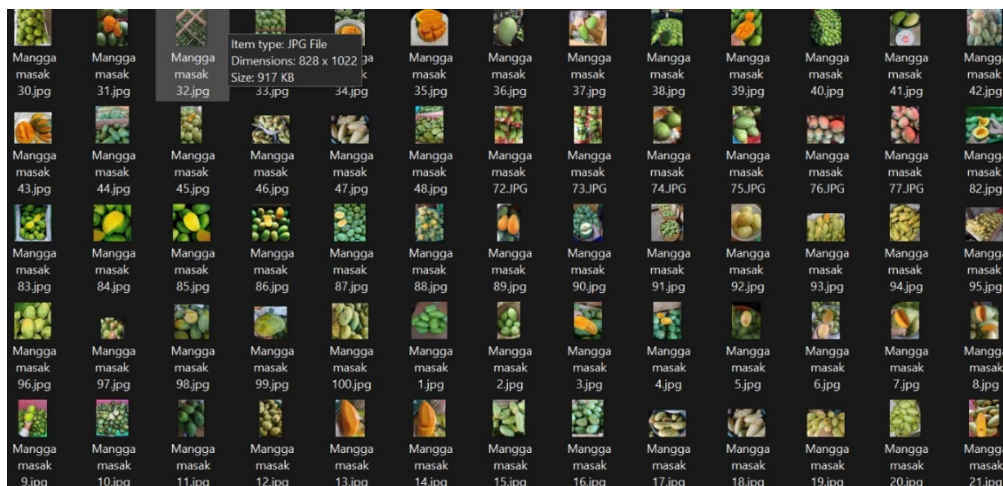
3.4 Dataset yang penelitian ini terdiri terbagi dalam tiga diimplementasikan

Penghimpunan Data digunakan dalam dari 300 gambar yang menggunakan arsitektur

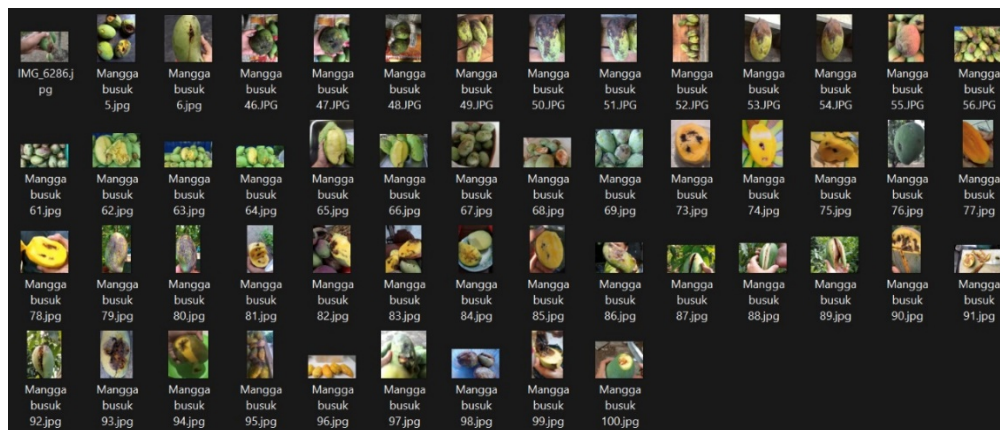
jaringan saraf konvolusional (Convolutional Neural Networks, CNN) dengan input berbasis skala warna HSV (Hue, Saturation, Value). HSV dipilih karena format ini mampu merepresentasikan warna lebih dekat dengan persepsi manusia, yang berguna untuk mendeteksi perbedaan warna secara lebih efektif dibandingkan dengan format RGB biasa.



Gambar 12. Mangga Muda



Gambar 13. Mangga Matang



Gambar 14. Mangga Busuk

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penggunaan metode Convolutional Neural Network (CNN) dan OpenCV untuk mendeteksi kematangan buah mangga berdasarkan analisis warna kulit memberikan hasil yang akurat. Sistem yang dikembangkan berhasil melakukan klasifikasi kematangan mangga dengan akurasi yang tinggi, yang berguna untuk industri pertanian dalam proses pemilahan buah secara otomatis. Metode ini mampu menggantikan penilaian manual yang cenderung tidak konsisten dan memakan waktu. Dengan kemampuan CNN dalam mengenali pola visual dan kekuatan OpenCV dalam pra-pemrosesan gambar, sistem ini dapat diintegrasikan dengan mudah ke dalam proses produksi tanpa memerlukan peralatan tambahan yang mahal.

Untuk meningkatkan akurasi dan generalisasi model, disarankan untuk mengembangkan dataset yang lebih beragam dalam hal jenis mangga, kondisi pencahayaan, dan variasi warna kulit. Sistem ini dapat diimplementasikan dalam skala industri dengan peningkatan pada kecepatan pemrosesan gambar agar mampu bekerja dalam waktu nyata (real-time). Penelitian lebih lanjut dapat dilakukan dengan menggunakan sensor tambahan, seperti sensor aroma atau tekstur, untuk memberikan penilaian kematangan yang lebih komprehensif. Optimalisasi lebih lanjut pada model CNN dapat dilakukan untuk mengurangi risiko overfitting dan meningkatkan efisiensi dalam penggunaan sumber daya komputasi.

DAFTAR PUSTAKA

- [1] Fasounaki, M., Yüce, E. B., Öncül, S., & Ince, G. (2021). CNN-based Text- independent Automatic Speaker Identification Using Short Utterances. *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021*, 01, 413–418.
- [2] Imron, S. &. (2018). Klasifikasi Buah Mangga Berdasarkan Tingkat Kematangan Menggunakan Least-Squares Support Vector Machine. *Explore IT : Jurnal Keilmuan Dan Aplikasi Teknik Informatika*, 10(2), 1–8.
- [3] Malik, R. A., & Zuliarso, E. (2021). Metode Convolutional Neural Network Untuk Mendeteksi Jenis Sayur Menggunakan Tensorflow. *Media Bina Ilmiah*, 15(1978), 5873–5882.
- [4] Purba, Y. B. E., Saragih, N. F., Silalahi, A. P., & ... (2022). Perancangan Alat Pendeteksi Kematangan Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode Convolutional Neural Network (CNN). *Jurnal Ilmiah Teknik ...*, 2(1), 13–21.
- [5] Utami, S., Baskoro, K., Perwati, L. K., & Murningsih, M. (2019). Keragaman Varietas Mangga (*Mangifera indica* L.) Di Kotamadya Semarang Jawa Tengah. *Bioma : Berkala Ilmiah Biologi*, 21(2), 121-125.
- [6] Arkadia, A., Ayu Damayanti, S., & Sandya Prasvita, D. (2021). Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan

- Metode CNN. Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) Jakarta-Indonesia, 2(2), 158–165.
- [7] Fitriyaningsih, & Rodiah. (2020). Klasifikasi Jenis Citra Daun Mangga Menggunakan Convolutional Neural Network. *Jurnal Ilmiah Teknologi Dan Rekayasa*, 25(3), 223–238.
- [8] Furqani, N. El. (n.d.). PENERAPAN TEKNOLOGI DEEP LEARNING DALAM PENGENALAN WAJAH UNTUK SISTEM KEAMANAN. 1–12.
- [9] Hanila, S., Afif Alghaffaru, M., Ekonomi, F., & Manajemen, P. (2023). Pelatihan Penggunaan Artificial Intelligence (AI) Terhadap Perkembangan Teknologi Pada Pembelajaran Siswa Sma 10 Sukarami Kota Bengkulu. *Jurnal Dehasen Mengabdi*, 2(2), 221–226.
- [10] Ihsan, M., Niswatin, R. K., & Swanjaya, D. (2021). Deteksi Ekspresi Wajah Menggunakan Tensorflow. *Joutica*, 6(1), 428.