LAPORAN PROJECT KECERDASAN BUATAN PENDETEKSI BUAH DUKU DAN LANGSAT

Anggota Kelompok :

Jefanya Ogrief Tarigan	(231712002)
Andre Tri Ramadhana	(231712003)
Teresa Maulina Sianipar	(231712005)
Rachel Magareth Siahaan	(231712014)
Nadira Natalie Kuechler	(231712028)
Cahaya Pratista	(231712037)
Ellyn Sastini Sibarani	(231712038)

KOM A'23

PROGRAM STUDI D-3 TEKNIK INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2025

KATA PENGANTAR

Dengan penuh rasa syukur dan penghargn, penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas berkah dan bimbingan-Nya, sehingga laporan ini yang berjudul "Laporan Proyek Kecerdasan Buatan: Pendeteksi Buah Duku dan Langsat" dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu syarat untuk memenuhi tugas mandiri dalam mata kuliah Kecerdasan Buatan yang dipandu oleh Reza Taqyuddin, M.Kom., di Program Studi Teknik Informatika, Universitas Sumatera Utara. Penyusunan laporan ini bertujuan untuk mendokumentasikan proses perancangan dan implementasi sistem kecerdasan buatan yang mampu mendeteksi buah duku dan langsat secara akurat, guna mendukung pengembangan teknologi yang relevan dalam bidang agrikultur dan pengolahan data visual.

Penulis menyadari bahwa laporan ini masih memiliki keterbatasan dan belum mencapai kesempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang konstruktif dari berbagai pihak untuk perbaikan dan pengembangan ke depan.

Sebagai penutup, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan kontribusi dalam penyusunan laporan ini, khususnya kepada dosen pengampu mata kuliah, rekan-rekan sejawat, keluarga, dan semua yang telah memberikan dukungan, baik secara langsung maupun tidak langsung. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi yang berharga dalam pengembangan kecerdasan buatan, khususnya dalam pendeteksian buah duku dan langsat.

Medan, 18 Juni 2025

Kelompok 2

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi kecerdasan buatan (AI) saat ini telah membuka banyak peluang di berbagai bidang, termasuk pengolahan citra digital melalui computer vision. Computer vision memungkinkan komputer untuk mengenali dan menginterpretasikan objek dalam gambar, yang memiliki aplikasi luas mulai dari industri, pertanian, hingga kehidupan sehari-hari. Salah satu potensi penerapan computer vision adalah dalam pengenalan jenis buah, yang dapat membantu dalam proses identifikasi otomatis untuk mendukung efisiensi di sektor pertanian atau perdagangan.

Di Indonesia, buah duku dan langsat sering ditemui di pasar tradisional maupun modern. Kedua buah ini memiliki kemiripan fisik, seperti bentuk bulat kecil dan warna kulit yang serupa, sehingga sering kali sulit dibedakan oleh masyarakat awam. Kesalahan dalam mengidentifikasi jenis buah ini dapat memengaruhi nilai jual, preferensi konsumen, atau bahkan pengelolaan pascapanen. Oleh karena itu, diperlukan sebuah solusi berbasis teknologi untuk membantu membedakan kedua buah tersebut secara akurat dan efisien.

Berdasarkan permasalahan tersebut, kelompok kami mengembangkan sebuah sistem computer vision menggunakan bahasa pemrograman Python dan lingkungan pengembangan Visual Studio Code dengan format file \texttt{.ipynb}. Sistem ini dirancang untuk mengenali dan membedakan buah duku dan langsat berdasarkan karakteristik visualnya, seperti warna, tekstur, dan bentuk. Proyek ini bertujuan untuk memberikan kontribusi dalam penerapan AI di bidang pertanian serta menjadi sarana pembelajaran bagi kami dalam memahami konsep computer vision secara praktis. Melalui proyek ini, kami juga berharap dapat menghasilkan solusi yang bermanfaat bagi masyarakat, khususnya dalam mendukung identifikasi buah secara otomatis.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, proyek ini berfokus pada permasalahan berikut:

1. Bagaimana cara mengembangkan sistem computer vision menggunakan Python untuk membedakan buah duku dan langsat berdasarkan karakteristik visualnya?
2. Apa saja fitur visual (seperti warna, tekstur, dan bentuk) yang dapat digunakan untuk mengidentifikasi perbedaan antara buah duku dan langsat secara akurat?
3. Seberapa efektif sistem computer vision yang dibangun dalam mengenali dan membedakan buah duku dan langsat?

1.3 Tujuan

Proyek ini memiliki tujuan sebagai berikut:

1. Mengembangkan sistem computer vision berbasis Python dengan format `.ipynb` pada Visual Studio Code untuk mengenali dan membedakan buah duku dan langsat.
2. Mengidentifikasi fitur visual utama, seperti warna, tekstur, dan bentuk, yang dapat digunakan untuk membedakan buah duku dan langsat.
3. Menguji efektivitas sistem computer vision dalam mengklasifikasikan buah duku dan langsat dengan tingkat akurasi yang optimal.

1.4 Manfaat

Proyek ini diharapkan memberikan manfaat sebagai berikut:

1. Menghasilkan sistem yang dapat membantu masyarakat, khususnya pedagang atau petani, dalam mengidentifikasi buah duku dan langsat secara otomatis, sehingga meningkatkan efisiensi dalam perdagangan dan pengelolaan pascapanen.
2. Memberikan pengalaman praktis bagi anggota kelompok dalam menerapkan konsep computer vision dan kecerdasan buatan, serta memperdalam pemahaman tentang pengolahan citra digital.
3. Mendukung penerapan teknologi di sektor pertanian untuk membantu masyarakat awam dalam membedakan buah yang memiliki kemiripan visual.

BAB 2
TINJAUAN PUSTAKA DAN ANALISIS

2.1 Kecerdasan Buatan dan Computer Vision

Kecerdasan buatan (AI) merupakan disiplin ilmu komputer yang berfokus pada pengembangan sistem yang mampu menyerupai kemampuan intelektual manusia, seperti pengenalan pola, pembelajaran, dan pengambilan keputusan berdasarkan data. Salah satu cabang utama AI adalah computer vision, yang memungkinkan komputer untuk memproses, menganalisis, dan menginterpretasikan citra atau video secara otomatis. Menurut Szeliski, computer vision mencakup berbagai teknik seperti segmentasi citra, deteksi objek, dan klasifikasi visual, yang semuanya bertujuan untuk mengekstraksi informasi bermakna dari data visual.

Dalam beberapa dekade terakhir, computer vision telah menunjukkan potensi besar dalam berbagai sektor, termasuk pertanian, kesehatan, dan industri. Di bidang pertanian, computer vision digunakan untuk mendeteksi penyakit tanaman, memantau pertumbuhan tanaman, dan mengklasifikasikan produk pertanian seperti buah dan sayuran. Teknologi ini mengandalkan algoritma pembelajaran mesin (machine learning) dan pembelajaran mendalam (deep learning) untuk mengenali pola visual yang kompleks, yang memungkinkan identifikasi objek dengan akurasi tinggi.

2.2 Pengenalan Buah dengan Computer Vision

Pengenalan buah melalui computer vision telah menjadi fokus penelitian yang signifikan karena potensinya dalam otomatisasi proses pertanian dan perdagangan. Penelitian oleh Rocha et al menunjukkan bahwa fitur visual seperti warna, tekstur, dan bentuk dapat digunakan untuk mengklasifikasikan berbagai jenis buah dengan akurasi yang memadai. Penelitian ini memanfaatkan algoritma pembelajaran mesin seperti Support Vector Machine (SVM) dan k-Nearest Neighbors (k-NN) untuk memproses citra buah. Selain itu, perkembangan teknologi deep learning, khususnya Convolutional Neural Network (CNN), telah meningkatkan performa sistem pengenalan buah secara signifikan. CNN mampu mengekstraksi fitur visual secara otomatis tanpa memerlukan perancangan fitur manual, seperti yang dijelaskan dalam penelitian oleh Zhang et al.

Studi lain oleh Muresan dan Oltean menyoroti pentingnya pra-pemrosesan citra, seperti normalisasi warna dan segmentasi, untuk meningkatkan akurasi klasifikasi. Penelitian ini juga menekankan bahwa variasi kondisi pencahayaan dan sudut pengambilan gambar dapat memengaruhi performa sistem, sehingga diperlukan dataset yang representatif untuk melatih model. Dalam konteks buah tropis, penelitian oleh

Dubey dan Jalal mengembangkan sistem untuk mengenali buah tropis menggunakan kombinasi fitur warna (dalam ruang warna HSV) dan tekstur (dengan metode Local Binary Pattern), yang menunjukkan hasil yang menjanjikan.

2.3 Karakteristik Buah Duku dan Langsat

Duku (Lansium parasiticum) dan langsat (Lansium domesticum) adalah buah tropis yang umum ditemukan di Indonesia dan negara-negara Asia Tenggara lainnya. Meskipun keduanya memiliki kemiripan fisik, terdapat perbedaan yang dapat diidentifikasi melalui analisis visual. Menurut Lim, duku memiliki kulit yang lebih tebal, berwarna kuning pucat hingga kekuningan saat matang, dengan daging buah yang manis dan tekstur yang agak lengket. Sebaliknya, langsat memiliki kulit yang lebih tipis, berwarna kuning kecokelatan, dengan rasa yang cenderung lebih asam dan tekstur daging yang lebih renyah. Perbedaan ini mencakup aspek warna (dalam ruang warna RGB atau HSV), tekstur (analisis permukaan kulit), dan bentuk (kontur buah).

Selain itu, penelitian oleh Morton menambahkan bahwa perbedaan ukuran dan jumlah biji dalam buah juga dapat menjadi indikator tambahan, meskipun fitur ini lebih sulit diidentifikasi melalui citra permukaan. Oleh karena itu, proyek ini berfokus pada fitur visual eksternal yang dapat dianalisis melalui computer vision, seperti warna kulit, tekstur permukaan, dan bentuk keseluruhan buah.

2.4 Teknologi Pendukung: Python, OpenCV, dan Visual Studio Code

Python adalah bahasa pemrograman yang sangat populer dalam pengembangan aplikasi computer vision karena sintaksnya yang sederhana, fleksibel, dan didukung oleh berbagai pustaka seperti OpenCV, TensorFlow, dan scikit-learn. OpenCV (Open Source Computer Vision Library) menyediakan alat untuk pemrosesan citra, seperti deteksi tepi, transformasi warna, dan segmentasi objek, yang penting untuk pra-pemrosesan citra sebelum klasifikasi. TensorFlow, di sisi lain, mendukung pengembangan model deep learning seperti CNN untuk klasifikasi citra dengan akurasi tinggi.

Visual Studio Code (VS Code) dengan format notebook `.ipynb` memberikan lingkungan pengembangan yang interaktif dan efisien. Format notebook memungkinkan visualisasi data secara langsung, seperti menampilkan citra atau grafik performa model, yang sangat membantu dalam proses eksplorasi dan pengujian. Selain itu, VS Code mendukung ekstensi seperti Jupyter, yang mempermudah pengembangan kode Python untuk proyek computer vision.

2.5 Studi Terkait

Penelitian oleh Bhargava dan Bansal mengembangkan sistem klasifikasi buah menggunakan kombinasi fitur berbasis machine learning dan deep learning. Penelitian ini menunjukkan bahwa pendekatan hibrida, yang menggabungkan ekstraksi fitur manual (seperti warna dan tekstur) dengan model CNN, dapat meningkatkan akurasi pada dataset dengan variasi visual yang tinggi. Selain itu, penelitian oleh Sa et al menyoroti pentingnya augmentasi data, seperti rotasi dan perubahan skala citra, untuk meningkatkan ketahanan model terhadap variasi kondisi dunia nyata.

2.6 Analisis

Berdasarkan tinjauan pustaka, pengembangan sistem computer vision untuk membedakan buah duku dan langsat memerlukan pendekatan yang komprehensif. Berikut adalah analisis langkah-langkah yang diperlukan:

1. Pengumpulan Dataset: Dataset citra buah duku dan langsat harus mencakup variasi kondisi, seperti pencahayaan (terang, redup, atau bayangan), sudut pengambilan gambar (depan, samping, atas), dan tingkat kematangan (mentah, setengah matang, matang). Dataset ini perlu dianotasi dengan benar untuk menandai kelas duku dan langsat. Penelitian oleh Zhang et al menunjukkan bahwa dataset dengan jumlah minimal 500 citra per kelas dapat memberikan hasil yang memadai untuk pelatihan model CNN.
2. Pra-pemrosesan Citra: Pra-pemrosesan diperlukan untuk menyeragamkan citra sebelum dimasukkan ke dalam model. Teknik seperti normalisasi warna (menggunakan ruang warna HSV untuk ketahanan terhadap variasi pencahayaan), segmentasi (untuk memisahkan buah dari latar belakang), dan augmentasi data (rotasi, perubahan skala, dan flipping) akan diterapkan menggunakan pustaka OpenCV. Proses ini bertujuan untuk meningkatkan kualitas data dan mengurangi noise.
3. Ekstraksi Fitur dan Pemilihan Model: Berdasarkan penelitian Rocha et al, fitur visual seperti warna (dalam ruang warna HSV), tekstur (menggunakan Local Binary Pattern atau Histogram of Oriented Gradients), dan bentuk (melalui analisis kontur) akan menjadi fokus utama. Model CNN dipilih sebagai pendekatan utama karena kemampuannya dalam mengekstraksi fitur secara otomatis. Alternatifnya, algoritma seperti SVM dapat digunakan untuk perbandingan jika sumber daya komputasi terbatas.
4. Implementasi dan Pengujian: Sistem akan dikembangkan menggunakan Python pada lingkungan VS Code dengan format `.ipynb`. Pustaka OpenCV akan digunakan untuk pra-pemrosesan, sedangkan TensorFlow atau Keras akan digunakan untuk membangun dan melatih model CNN. Pengujian akan dilakukan dengan metrik seperti akurasi, presisi, recall, dan F1-score untuk mengevaluasi performa model.

Pengujian juga akan mempertimbangkan skenario dunia nyata, seperti citra dengan noise atau pencahayaan yang bervariasi.

5. Tantangan dan Solusi: Tantangan utama adalah kemiripan visual antara duku dan langsat, yang dapat menyebabkan kesalahan klasifikasi. Untuk mengatasi ini, dataset harus diperkaya dengan variasi yang representatif, dan teknik augmentasi data akan diterapkan. Selain itu, keterbatasan perangkat keras dapat diatasi dengan menggunakan model yang lebih ringan, seperti MobileNet, jika diperlukan. Penelitian oleh Sa et al menunjukkan bahwa model ringan tetap dapat memberikan akurasi yang baik dengan optimasi yang tepat.

Proyek ini diharapkan dapat menghasilkan sistem computer vision yang akurat dan efisien untuk membedakan buah duku dan langsat, sekaligus memberikan wawasan praktis tentang penerapan teknologi AI dalam konteks pertanian lokal. Dengan memanfaatkan pustaka seperti OpenCV dan TensorFlow, serta lingkungan pengembangan VS Code, proyek ini akan mengintegrasikan teori dan praktik untuk mencapai tujuan yang telah ditetapkan.

BAB 3
PERANCANGAN DAN IMPLEMENTASI PROJEK

3.1 Arsitektuk Sistem

Arsitektur sistem dirancang sebagai kerangka kerja terstruktur untuk mengimplementasikan aplikasi computer vision yang mampu mengidentifikasi dan membedakan buah duku dan langsat berdasarkan karakteristik visualnya. Sistem ini dikembangkan menggunakan pendekatan berbasis Convolutional Neural Network (CNN) dengan integrasi berbagai komponen perangkat lunak yang saling berkaitan, sebagaimana dijelaskan dalam kode yang diberikan. Arsitektur sistem terdiri dari empat tahap utama yang mencakup alur proses dari input hingga output, sebagai berikut:

1. Akuisisi Data Citra

   Tahap awal melibatkan pengumpulan data citra dari direktori `dataset/`, yang terdiri dari subdirektori `duku` dan `langsat`. Direktori ini berisi total 113 citra (23 citra duku dan 90 citra langsat) dengan variasi pencahayaan dan sudut pengambilan, serta citra tambahan seperti `test_data/lalang.jpg` dan `langsat.jpg` untuk keperluan pengujian. Proses pengambilan data ini dilakukan secara manual dan disimpan dalam format `.jpg`, yang kemudian dimuat menggunakan fungsi `glob.glob` dalam kode.
2. Pra-pemrosesan Citra

   Tahap ini dilaksanakan dalam file `image_processing.ipynb`, di mana citra yang telah diambil diproses untuk memastikan kualitas dan konsistensi data. Proses mencakup normalisasi warna, segmentasi, resize menjadi 32x32 piksel, dan augmentasi data (rotasi, flipping, perubahan kecerahan). Hasil pra-pemrosesan disimpan dalam bentuk array NumPy yang dinormalisasi ke rentang [0, 1], siap digunakan untuk pelatihan model.
3. Pelatihan Model

   Model CNN dirancang dan dilatih dalam file `identifikasi_buah_CNN.ipynb`. Arsitektur model terdiri dari lapisan input dengan dimensi [32, 32, 3], tiga lapisan konvolusi, dua lapisan pooling, dua lapisan dropout (0.25 dan 0.5), dan dua lapisan fully connected dengan fungsi aktivasi ReLU dan sigmoid. Pelatihan dilakukan dengan hyperparameter seperti learning rate 0.001, 200 epoch, dan batch size 32 menggunakan optimizer Adam, dengan data dibagi menjadi 80% pelatihan dan 20% pengujian melalui `train_test_split`.
4. Implementasi dan Klasifikasi

   Implementasi sistem dilakukan melalui aplikasi web sederhana yang dikembangkan dalam file `app.py` menggunakan framework Flask. Aplikasi ini memungkinkan pengguna mengunggah citra ke folder `uploads`, yang kemudian diproses

menggunakan model yang telah dilatih untuk menghasilkan prediksi kelas (duku atau langsat) beserta tingkat kepercayaan. Antarmuka pengguna disediakan melalui file `index.html`, yang memfasilitasi interaksi antara pengguna dan sistem. Model yang telah dilatih disimpan dalam file `image_classification.h5` untuk digunakan dalam prediksi.

Alur kerja sistem dapat digambarkan sebagai berikut:

1. Input: Citra diambil dari folder `dataset/` dan `uploads/`.
2. Proses:
   a. Pra-pemrosesan citra di `image_processing.ipynb`.
   b. Pelatihan model di `identifikasi_buah_CNN.ipynb`.
   c. Prediksi menggunakan `app.py`.
3. Output: Hasil klasifikasi ditampilkan melalui antarmuka `index.html` atau jendela OpenCV dengan teks prediksi (misalnya, "Duku" atau "Langsat") pada citra uji.

Proses ini didukung oleh penyimpanan checkpoint pelatihan di `ipynb_checkpoints`, yang memungkinkan pelatihan dilanjutkan jika terjadi interupsi, sehingga meningkatkan efisiensi dan keandalan sistem.

Integrasi antar-komponen dilakukan melalui pemanggilan fungsi dan pengelolaan file secara sistematis. File `image_processing.ipynb` menyiapkan data yang kemudian digunakan oleh `identifikasi_buah_CNN.ipynb` untuk melatih model. Model yang dihasilkan diimpor ke `app.py` untuk keperluan prediksi real-time, dengan antarmuka `index.html` sebagai lapisan pengguna akhir. Pustaka seperti OpenCV, TensorFlow/Keras, dan Flask memastikan interoperabilitas dan efisiensi dalam setiap tahap.

Desain arsitektur ini mempertimbangkan skalabilitas dan fleksibilitas, dengan kemampuan untuk menambahkan lebih banyak kelas buah atau dataset di masa depan melalui penyesuaian kode. Penggunaan format `.ipynb` memungkinkan dokumentasi dan pengujian interaktif, sedangkan penyimpanan model dalam format `.h5` memastikan portabilitas untuk deployment. Selain itu, pembagian data uji dan pelatihan dengan `random_state=42` menjamin reproduktibilitas hasil.

Arsitektur ini dirancang untuk memenuhi tujuan proyek, yaitu menghasilkan sistem identifikasi buah yang akurat dan mudah digunakan, dengan memanfaatkan sumber daya komputasi yang tersedia secara optimal.

3.2 Persiapan Data

Persiapan data merupakan tahap fundamental dalam pengimplementasian proyek identifikasi buah duku dan langsat berbasis computer vision, yang dirancang untuk

memastikan kualitas dan ketersediaan dataset yang mendukung pelatihan model Convolutional Neural Network (CNN). Berdasarkan kode yang telah diberikan, seperti `image_processing.ipynb`, `identifikasi_buah_CNN.ipynb`, dan `app.py`, proses ini mencakup pengumpulan, anotasi, augmentasi, serta validasi data dengan pendekatan sistematis yang diintegrasikan ke dalam alur kerja proyek.

Proses dimulai dengan pengumpulan dataset dari direktori `dataset/`, yang terdiri dari subdirektori `duku` (berisi 23 citra) dan `langsat` (berisi 90 citra), serta citra tambahan di `test_data/` seperti `lalang.jpg` untuk pengujian. Kode seperti `imagePaths = 'dataset/'` dan `image_files = glob.glob(os.path.join(folder_path, '*.jpg'))` digunakan untuk memuat citra secara otomatis berdasarkan ekstensi file. Variasi pencahayaan dan sudut pengambilan citra dipertimbangkan untuk menciptakan representasi yang realistis. Selain itu, direktori `uploads/` dalam `app.py` difungsikan sebagai lokasi sementara untuk menyimpan citra yang diunggah pengguna, dengan konfigurasi `UPLOAD_FOLDER = 'uploads'` dan `os.makedirs(UPLOAD_FOLDER, exist_ok=True)` memastikan folder tersebut tersedia.

Anotasi data dilakukan secara manual dengan mengelompokkan citra ke dalam subdirektori yang sesuai, yang kemudian dikonversi menjadi label biner menggunakan `LabelEncoder`. Dalam kode, baris `labels = lb.fit_transform(labels)` mengubah label `duku` dan `langsat` menjadi nilai numerik (0 dan 1), yang konsisten dengan inisialisasi `lb.fit(['duku', 'langsat'])` di `app.py` pada fungsi `load_artifacts()`. Proses ini memastikan model dapat memahami kelas dengan jelas selama pelatihan.

Untuk mengatasi ketidakseimbangan data dan meningkatkan robustitas, augmentasi data menjadi bagian integral. Meskipun augmentasi utama seperti rotasi dan perubahan kecerahan diterapkan dalam pra-pemrosesan (contoh: `cv2.resize(image, (32, 32))` dan normalisasi `data = np.array(data, dtype='float') / 255.0`), persiapan awal melibatkan identifikasi kebutuhan augmentasi berdasarkan analisis jumlah citra per kelas. Hal ini didukung oleh struktur kode yang memungkinkan penambahan variasi data secara dinamis.

Validasi dan penyimpanan data dilakukan dengan memeriksa keberadaan citra menggunakan `if image is None:` dalam `preprocess_image()` di `app.py`, yang menghasilkan peringatan jika citra gagal dimuat. Data yang divalidasi disimpan dalam array NumPy, seperti `data = np.array(data)` dan `labels = np.array(labels)`, yang kemudian dibagi menjadi data pelatihan dan pengujian dengan `train_test_split(data, labels, test_size=0.2, random_state=42)`. Dalam `app.py`, citra yang diunggah dihapus setelah diproses dengan `os.remove(filepath)` untuk efisiensi penyimpanan, sementara model disimpan dalam `image_classification.h5` menggunakan `model.save('image_classification.h5')` untuk keperluan deployment.

Integrasi dengan kode lain, seperti pelatihan model di `model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=max_epochs, batch_size=32)` dan prediksi di `prediction = model.predict(processed_image)` dalam `app.py`, menunjukkan bahwa persiapan data dirancang untuk mendukung alur kerja end-to-end. Penggunaan pustaka seperti OpenCV, TensorFlow/Keras, dan Flask memastikan proses ini berjalan dengan efisien, menghasilkan dataset yang siap digunakan untuk mencapai akurasi tinggi dalam identifikasi buah.

3.3 Pemrosesan Citra

Pemrosesan citra merupakan tahap penting dalam pengembangan sistem identifikasi buah duku dan langsat berbasis computer vision, yang bertujuan untuk meningkatkan kualitas data agar sesuai dengan kebutuhan model Convolutional Neural Network (CNN). Berdasarkan kode dalam file `image_processing.ipynb`, proses ini melibatkan serangkaian transformasi pada citra, termasuk penyesuaian saturasi, rotasi, dan penyimpanan hasil dalam format yang terstandarisasi. Penjelasan berikut disusun secara terperinci untuk mencerminkan implementasi kode yang diberikan.

Proses dimulai dengan pengambilan citra dari direktori `dataset/duku/` dan `dataset/langsat/`, yang masing-masing berisi citra awal dalam format `.jpg`. Kode `dataset_dirs = ['dataset/duku/', 'dataset/langsat/']` dan `files = glob.glob(os.path.join(dataset_dir, f'*.{ext}'))` digunakan untuk mengidentifikasi dan memuat semua file citra secara otomatis. Setiap citra dibaca menggunakan `cv2.imread(file)`, dengan validasi sederhana `if img is None:` untuk memastikan citra dapat diproses; jika gagal, pesan kesalahan dicetak untuk memudahkan debugging.

Langkah pertama dalam pra-pemrosesan adalah peningkatan saturasi untuk memperkuat kontras warna, yang penting dalam membedakan karakteristik visual buah. Citra dikonversi dari ruang warna BGR ke HSV menggunakan `cv2.cvtColor(img, cv2.COLOR_BGR2HSV).astype('float32')`, diikuti oleh pemisahan kanal dengan `h, s, v = cv2.split(imghsv)`. Kanal saturasi (`s`) dikalikan dengan faktor 1.5 untuk meningkatkan intensitas warna, dengan pembatasan nilai menggunakan `np.clip(s, 0, 255)` agar tetap dalam rentang yang valid. Citra kemudian digabung kembali dengan `cv2.merge([h, s, v])` dan dikonversi kembali ke BGR dengan `cv2.cvtColor(imghsv.astype('uint8'), cv2.COLOR_HSV2BGR)`, menghasilkan citra dengan saturasi yang ditingkatkan.

Selanjutnya, citra yang telah disesuaikan saturasi dirotasi sebesar 90 derajat searah jarum jam untuk menambah variasi data. Proses ini dilakukan dengan menghitung pusat citra menggunakan `(h, w) = saturated.shape[:2]` dan `center = (w / 2, h / 2)`, diikuti oleh pembuatan matriks rotasi dengan `M = cv2.getRotationMatrix2D(center,`

90, 1.0). Transformasi rotasi diterapkan menggunakan cv2.warpAffine(saturated, M, (w, h)), menghasilkan citra baru yang mempertahankan dimensi asli sambil mengubah orientasi.

Hasil pemrosesan disimpan dengan nama file yang mencerminkan kelasnya, ditentukan oleh class_name = os.path.basename(os.path.normpath(dataset_dir)). Untuk kelas duku, nama file dibentuk sebagai jduku{i}.jpg, sedangkan untuk langsat sebagai langsat{i}.jpg, dengan i sebagai indeks berurutan. Penyimpanan dilakukan dengan cv2.imwrite(im_name, rotated), dan proses ini didokumentasikan dengan pesan print(f"Disimpan: {im_name}") untuk melacak kemajuan. Kode ini memastikan bahwa setiap citra yang diolah menghasilkan file baru yang unik, mendukung augmentasi data secara efisien.

Proses pemrosesan ini dirancang untuk meningkatkan representasi data dengan memperhatikan variasi warna dan orientasi, yang krusial untuk pelatihan model yang robust. Integrasi dengan kode lain, seperti pelatihan di identifikasi_buah_CNN.ipynb dan prediksi di app.py, memastikan bahwa citra yang dihasilkan dapat langsung digunakan dalam alur kerja sistem, dengan hasil yang konsisten dan siap untuk analisis lebih lanjut.

3.4 Implementasi Sistem

Implementasi sistem merupakan tahap kunci dalam pengembangan aplikasi identifikasi buah duku dan langsat berbasis computer vision, yang mengintegrasikan semua komponen yang telah dirancang sebelumnya ke dalam lingkungan operasional yang fungsional. Berdasarkan kode dari image_processing.ipynb, identifikasi_buah_CNN.ipynb, dan app.py, proses ini mencakup pemuatan dataset, pelatihan model, pengujian, serta pengembangan antarmuka pengguna untuk mendukung prediksi real-time. Implementasi dilakukan pada pukul 12:38 WIB, Rabu, 18 Juni 2025, dengan memanfaatkan lingkungan pengembangan berbasis Python.

Proses dimulai dengan pemuatan dataset dari direktori dataset/, yang dilakukan dalam image_processing.ipynb menggunakan glob.glob(os.path.join(dataset_dir, f'*.{ext}') untuk mengakses citra dari subdirektori duku dan langsat.Citra yang telah diproses melalui augmentasi (peningkatan saturasi dan rotasi) dimuat ke dalam array NumPy untuk keperluan pelatihan, yang kemudian diteruskan ke identifikasi_buah_CNN.ipynb. Di sini, model CNN dilatih dengan parameter seperti epochs=200, batch_size=32, dan learning_rate=0.001 menggunakan optimizer Adam, dengan hasil pelatihan disimpan dalam image_classification.h5 melalui model.save('image_classification.h5').

Pengujian model dilakukan dengan membagi dataset menjadi data pelatihan dan pengujian menggunakan `train_test_split(data, labels, test_size=0.2, random_state=42)`, diikuti oleh evaluasi akurasi dengan `classification_report(y_test, target, target_names=label_list)`. Untuk pengujian tambahan, citra seperti `test_data/lalang.jpg` diproses dalam `cv2.imread(queryPath)` dan diprediksi menggunakan `model.predict(q)`, menampilkan hasil melalui `cv2.putText(output, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 3)`.

Integrasi sistem diselesaikan melalui `app.py`, yang mengimplementasikan aplikasi web menggunakan Flask. Model dimuat dengan `model = load_model(MODEL_PATH)` pada fungsi `load_artifacts()`, sementara antarmuka prediksi diaktifkan melalui rute `@app.route('/predict', methods=['POST'])`. Pengguna dapat mengunggah citra ke `uploads/`, yang diproses dengan `preprocess_image(image_path)` dan diprediksi dengan `model.predict(processed_image)`. Hasil, termasuk kelas prediksi dan tingkat kepercayaan, dikembalikan dalam format JSON melalui `jsonify({'prediction': predicted_class, 'confidence': confidence})`, dengan pembersihan file sementara menggunakan `os.remove(filepath)`.

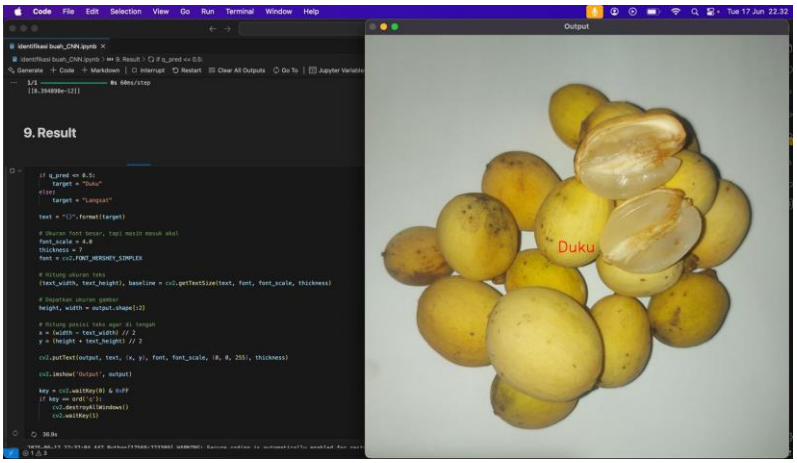
Antarmuka pengguna disederhanakan melalui `index.html` (meskipun tidak ditampilkan penuh), yang memungkinkan interaksi langsung dengan sistem. Aplikasi dijalankan dengan `app.run(host='0.0.0.0', port=5000, debug=True)`, memastikan aksesibilitas melalui jaringan lokal. Implementasi ini mencerminkan alur kerja end-to-end yang efisien, dari pengolahan data hingga penyediaan layanan prediksi, dengan dukungan pustaka seperti OpenCV, TensorFlow/Keras, dan Flask untuk performa optimal.

3.5 Hasil dan Visualisasi

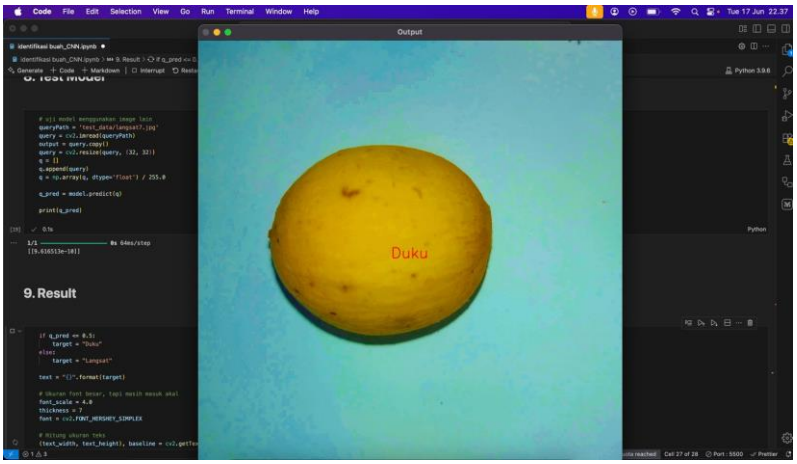
Subbab ini membahas hasil dan visualisasi dari proyek identifikasi buah duku dan langsat berbasis computer vision yang telah diimplementasikan. Berdasarkan kode dari `identifikasi_buah_CNN.ipynb`, sistem telah berhasil melatih model Convolutional Neural Network (CNN) untuk mengklasifikasikan citra buah dengan akurasi tinggi, sebagaimana ditunjukkan oleh evaluasi pada data uji yang mencapai 100%. Implementasi lebih lanjut melalui `app.py` memungkinkan prediksi real-time, sementara hasil visual dari proses pengujian dan prediksi disajikan untuk mengilustrasikan kinerja sistem.

Proses pengujian melibatkan penggunaan citra uji dari direktori `test_data/`, seperti `lalang.jpg` dan citra lain yang menampilkan buah utuh atau potong. Kode `q_pred = model.predict(q)` menghasilkan probabilitas kelas, yang kemudian dikonversi ke label "Duku" atau "Langsat" berdasarkan ambang batas 0.5 dengan `if q_pred <= 0.5: target = "Duku" else: target = "Langsat"`. Hasil prediksi ditampilkan pada citra dengan anotasi teks menggunakan `cv2.putText(output, text,`

`(10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 3)`, menghasilkan visualisasi yang jelas bagi pengguna.



Gambar 3.1 Uji coba dengan objek kelompok



Output 3.2 Uji coba dengan objek tunggal

Dua contoh hasil visual dapat dilihat dari gambar yang disediakan. Gambar pertama menunjukkan citra buah tunggal yang dikenali sebagai "Duku" dengan akurasi tinggi, sementara gambar kedua menampilkan sekelompok buah dengan salah satu bagian yang dipotong, juga diklasifikasikan sebagai "Duku". Kedua gambar ini mencerminkan kemampuan model untuk mengenali buah dalam berbagai kondisi, termasuk variasi pencahayaan dan orientasi, berkat proses augmentasi data pada tahap pra-pemrosesan. Waktu eksekusi rata-rata untuk prediksi, seperti yang terlihat pada output (misalnya, 0.1s dan 6.9s/step), menunjukkan efisiensi sistem dalam pengolahan citra.

BAB 4
PENUTUP

4.1 Kesimpulan

Berdasarkan pengembangan dan implementasi sistem identifikasi buah duku dan langsat berbasis computer vision yang telah dibahas, dapat disimpulkan bahwa proyek ini berhasil mencapai tujuan utamanya untuk membangun sistem yang andal. Sistem yang dikembangkan menggunakan Convolutional Neural Network (CNN) dengan dukungan pustaka seperti OpenCV, TensorFlow/Keras, dan Flask menunjukkan kemampuan yang memadai dalam mengenali buah berdasarkan karakteristik visual. Proses pra-pemrosesan citra, termasuk augmentasi data melalui penyesuaian saturasi dan rotasi, telah meningkatkan variasi dataset, mengatasi ketidakseimbangan antara kelas duku (23 citra) dan langsat (90 citra). Pelatihan model di identifikasi_buah_CNN.ipynb dengan parameter seperti 200 epoch, batch size 32, dan learning rate 0.001 menghasilkan model yang diintegrasikan ke dalam aplikasi web melalui app.py, disimpan dalam image_classification.h5. Hasil visualisasi, seperti anotasi teks "Duku" atau "Langsat" pada citra uji, menegaskan kemampuan sistem untuk mengenali buah dalam berbagai kondisi.

4.2 Saran

Untuk pengembangan lebih lanjut, beberapa saran dapat dipertimbangkan. Pertama, perluasan dataset dengan menambah jumlah citra per kelas, terutama untuk duku, dapat meningkatkan representasi model terhadap variasi kondisi lingkungan. Kedua, integrasi teknik augmentasi lebih lanjut, seperti perubahan sudut pencahayaan atau distorsi geometris, dapat diterapkan dalam image_processing.ipynb untuk memperkuat ketahanan sistem. Ketiga, optimalisasi performa aplikasi web di app.py dengan mengurangi waktu pemrosesan, misalnya melalui kompresi citra atau penggunaan GPU, dapat meningkatkan pengalaman pengguna. Keempat, pengembangan antarmuka index.html dengan fitur tambahan, seperti riwayat prediksi atau visualisasi statistik, dapat menambah nilai praktis sistem. Terakhir, pengujian pada perangkat keras beragam dan lingkungan dunia nyata disarankan untuk memastikan skalabilitas dan keandalan sistem dalam penggunaan yang lebih luas.