



Klasifikasi Jenis Tanaman Alpukat Berdasarkan Bentuk Daun Menggunakan Algoritma CNN

Agum Pratama¹, Tito Sugiharto², Panji Novantara³

Email: ¹pratamaagum15@gmail.com, ²tito@uniku.ac.id, ³panji.novantara@uniku.ac.id

¹²³Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Kuningan

Diterima: 18 Juni 2025 | Direvisi: 01 Agustus 2025 | Disetujui: 04 Agustus 2025

©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Tanaman alpukat merupakan salah satu komoditas hortikultura yang sangat populer di Indonesia, terutama di Pulau Jawa, karena manfaat kesehatannya dan nilai ekonomis yang tinggi. Namun, perbedaan bentuk daun pada berbagai varietas alpukat seringkali menyulitkan proses identifikasi, baik bagi pembeli maupun penjual, sehingga dapat menyebabkan kesalahan transaksi dan kerugian. Identifikasi secara manual memerlukan keterampilan khusus yang tidak selalu tersedia, terutama di daerah seperti Kabupaten Kuningan. Untuk menjawab permasalahan tersebut, penelitian ini bertujuan mengembangkan aplikasi berbasis Android yang mampu mengklasifikasikan varietas alpukat, yakni alligator, kendil, dan mentega, berdasarkan citra daun secara otomatis. Aplikasi ini menggunakan algoritma Convolutional Neural Network (CNN) dengan model pre-trained SSDMobileNetV2 FPN-Lite yang diimplementasikan melalui framework TensorFlow. Dataset yang digunakan terdiri dari 4.800 gambar daun alpukat yang dibagi untuk proses pelatihan, validasi, dan pengujian. Hasil pengujian menunjukkan bahwa model mampu mencapai tingkat akurasi sebesar 99%. Untuk kelas alligator, nilai presisi dan recall masing-masing adalah 1.00 dan 0.98; untuk kelas kendil sebesar 1.00 dan 0.99; dan untuk kelas mentega sebesar 0.99 dan 1.00. Temuan ini membuktikan bahwa algoritma CNN efektif dalam mengklasifikasikan varietas alpukat berdasarkan ciri visual daun. Dengan demikian, aplikasi ini berpotensi menjadi alat bantu yang cepat, akurat, dan praktis dalam proses identifikasi varietas alpukat, baik untuk keperluan komersial maupun edukatif.

Kata kunci: Daun Alpukat, Deep Learning, CNN, MobileNetV2, SSD, TensorFlow.

Classification of Avocado Plant Varieties Based on Leaf Shape Using CNN Algorithm

Abstract

Avocado plants are one of the most popular horticultural commodities in Indonesia, especially in Java, due to their health benefits and high economic value. However, differences in leaf shape across avocado varieties often make identification difficult for both buyers and sellers, which can lead to transaction errors and losses. Manual identification requires specialised skills that are not always available, especially in areas such as Kuningan Regency. To answer these problems, this research aims to develop an Android-based application that is able to classify avocado varieties, namely alligator, kendil, and butter, based on leaf images automatically. This application uses Convolutional Neural Network (CNN) algorithm with SSDMobileNetV2 FPN-Lite pre-trained model implemented through TensorFlow framework. The dataset used consists of 4,800 avocado leaf images divided for training, validation, and testing processes. The test results show that the model is able to achieve an accuracy rate of 99%. For the alligator class, the precision and recall values were 1.00 and 0.98 respectively; for the kendil class, 1.00 and 0.99; and for the butter class, 0.99 and 1.00. These findings prove that the CNN algorithm is effective in classifying avocado varieties based on visual features of the leaves. Thus, this application has the potential to be a fast, accurate, and pre-classified tool.

Keywords: Avocado Leaf, Deep Learning, CNN, MobileNetV2, SSD, TensorFlow

1. PENDAHULUAN

Tanaman alpukat merupakan salah satu tanaman berakar tunggang yang banyak dikembangkan di Indonesia karena nilai ekonomis dan manfaat kesehatannya. Pulau Jawa menjadi wilayah utama dalam produksi alpukat dengan berbagai varietas seperti mentega, kendil, alligator, wina, dan lainnya. Alpukat salah satu tanaman yang populer dan banyak dikonsumsi oleh masyarakat di Indonesia, khususnya di Pulau Jawa [1]. Berdasarkan data dari Badan Pusat Statistik (2024), produksi alpukat di Indonesia tercatat mencapai 669.260 ton pada tahun 2021, meningkat menjadi 865.780 ton pada tahun 2022, dan meningkat lagi menjadi 874.046 ton pada tahun 2023 [2]. Tingginya nilai ekonomi pada tanaman alpukat, mendorong banyak orang untuk membudidayakan tanaman ini [3]. Namun terdapat tantangan yang sering dihadapi oleh pedagang dan konsumen adalah dalam membedakan jenis bibit tanaman alpukat, terutama ketika bibit memiliki ciri-ciri yang mirip. Permasalahan umum ini mengakibatkan kerugian finansial karena bibit yang ditanam tidak sesuai dengan harapan dan tujuan. Permasalahan serupa terjadi di “Toko Bibit dan Bunga Hikmah Tani Kadugede” di Kabupaten Kuningan, di mana identifikasi secara manual membutuhkan ahli botani atau agronom sehingga tidak efektif untuk diterapkan pada populasi dalam jumlah besar.

Untuk mengatasi permasalahan ini, dibutuhkan kemampuan mengenali jenis bibit alpukat secara akurat dan mudah. Identifikasi manual membutuhkan keahlian khusus dan pengalaman, yang tidak dimiliki semua orang. Oleh karena itu, keterlibatan pakar sangat penting. Dalam konteks ini, Bapak Yana sebagai pemilik toko sekaligus pakar bibit lokal berperan penting dalam proses validasi data dan pemetaan jenis varietas yang beredar di wilayah tersebut.

Sebagai solusi yang lebih praktis dan efisien, teknologi *Deep Learning* menjadi alternatif yang menjanjikan. Dengan pendekatan ini, proses klasifikasi citra daun dapat dilakukan secara otomatis dan akurat. Salah satu algoritma yang banyak digunakan dalam klasifikasi visual adalah *Convolutional Neural Network* (CNN)[4]. CNN terbukti efektif dalam mengenali pola kompleks dalam gambar dan telah banyak diterapkan dalam bidang pertanian digital[5]. Penggunaan CNN dalam penelitian ini diharapkan dapat memberikan kontribusi nyata dalam membantu konsumen dan pelaku usaha mengenali jenis alpukat secara cepat, tepat, dan efisien.

Penelitian sebelumnya telah menunjukkan keberhasilan penggunaan *Deep Learning*, khususnya *Convolutional Neural Network* (CNN), untuk klasifikasi citra tanaman. Penelitian pada daun paprika mencapai akurasi 99%. Dataset yang digunakan sebanyak 9892 dalam format jpg, terbagi ke dalam dua kelas yaitu, kelas “*Bacterial_spot*” sebanyak 3988 citra, dan kelas “*healthy*” sebanyak 5904 citra. Model penelitian ini menggunakan 2 lapisan convolution layer, 2 lapisan pooling layer dengan ukuran (2x2), 1 lapisan dropout, 2 lapisan dense (fully connected layer), dan 1 lapisan flatten. Penelitian ini menggunakan Python dengan framework TensorFlow dan Keras. Model yang dilatih menunjukkan akurasi 97%. Pengujian dengan data testing menunjukkan performa model yang sangat baik, dengan nilai precision, recall, dan f1 score sebesar 99%, serta akurasi pengujian juga mencapai 99%. Ini membuktikan bahwa model mampu melakukan klasifikasi dengan tepat[6]. Identifikasi 10 jenis daun tanaman mencapai 92%. Dataset yang digunakan sebanyak 10.000 gambar dalam format jpg, masing-masing 400 gambar untuk 10 jenis daun tanaman yang berbeda, diantaranya daun nangka, jambu biji, sirih, jeruk nipis, belimbing uluh, kemangi, seledri, pepaya, pandan, dan lidah buaya. Dalam penelitian ini ukuran datasetnya 224 x 224 pixel. Hasil dari penelitian ini menghasilkan akurasi sebesar 92% dan nilai f1-Score sebesar 88,51. Nilai recall sebesar 87% dan nilai precision sebesar 89%. Dengan hasil tersebut maka dapat dikatakan bahwa penggunaan metode Convolutional Neural Network (CNN) akurat digunakan untuk identifikasi daun tanaman berdasarkan visualisasi citra dari sepuluh jenis daun tumbuhan. [7]. dan perbandingan model pada tanaman herbal mencapai 98% [8].

Hipotesis dalam penelitian ini adalah bahwa algoritma Convolutional Neural Network (CNN) dapat digunakan secara efektif untuk mengklasifikasikan jenis tanaman alpukat berdasarkan citra daun, khususnya untuk tiga varietas utama yaitu alligator, kendil, dan mentega. Dengan kemampuan CNN dalam mengenali pola visual yang kompleks, sistem klasifikasi ini diharapkan mampu membedakan karakteristik morfologi daun dari masing-masing varietas secara akurat. Selain itu, implementasi model CNN dalam bentuk aplikasi berbasis Android diyakini dapat memberikan solusi praktis dan efisien dalam proses identifikasi, sehingga mempermudah pengguna, baik petani, penjual, maupun pembeli bibit alpukat, dalam mengenali jenis bibit secara langsung melalui pengambilan gambar daun.

2. METODE PENELITIAN

Untuk memperoleh data penelitian, ditentukan metode dan teknik penelitian. Dalam penelitian ini terdiri dari metode pengumpulan data, metode pengembangan sistem, dan metode penyelesaian masalah.

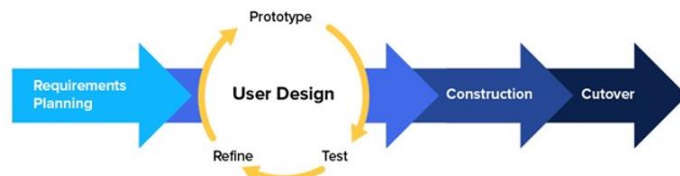
2.1. Metode Pengumpulan Data

Metode pengumpulan data dilakukan untuk mendapatkan informasi dari data penelitian. Metode pengumpulan data dengan observasi sekaligus melakukan wawancara dengan bapak Yana selaku pemilik Toko Bibit dan Bunga Hikmah Tani Kadugede dan Studi Pustaka dalam mengumpulkan informasi-informasi mengenai Computer Vision, Deep Learning, Algoritma CNN, TensorFlow, MobileNetV2, SSD, Image Classification, Object Detection, Android, dan literatur terkait.

2.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Rapid Application Development* (RAD), yang menekankan pengembangan iteratif dan pembuatan prototipe cepat Model RAD terdiri dari empat tahapan utama: Tahap *requirements planning*, peneliti meneliti masalah yang sedang berlangsung, memutuskan apa yang diperlukan dalam membangun suatu sistem aplikasi, karena langkah ini merupakan tahap pertama untuk mencapai keberhasilan dalam

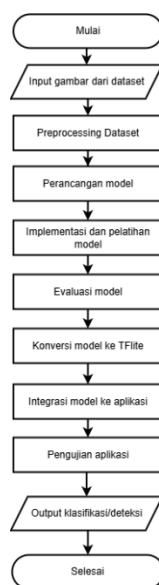
pembangunan sebuah sistem; Tahap *user design*, peneliti membuat prototype sebagai representasi awal untuk memahami fitur-fitur yang akan ada di aplikasi, dilanjutkan dengan memberikan umpan balik berdasarkan pengalaman menggunakan prototype, peneliti memperbaiki bug dan fitur yang tidak diinginkan hingga *prototype* memenuhi harapan; Tahap *construction*, diawali dari tahap desain hingga model kerja. peneliti memastikan bahwa sistem atau aplikasi yang dibangun berfungsi dengan baik dan sesuai dengan yang diharapkan user; Tahap *cutover*. peneliti melakukan pengujian setelah *construction*, baik dari segi front-end dan back-end telah sesuai. Model Rapid Application Development (RAD) di tunjukkan pada Gambar 1.



Gambar 1. Model RAD

2.3. Metode Penyelesaian Masalah

Metode penyelesaian masalah menggunakan pendekatan *Deep Learning* dengan algoritma CNN dan Framework TensorFlow untuk klasifikasi gambar dan deteksi objek.

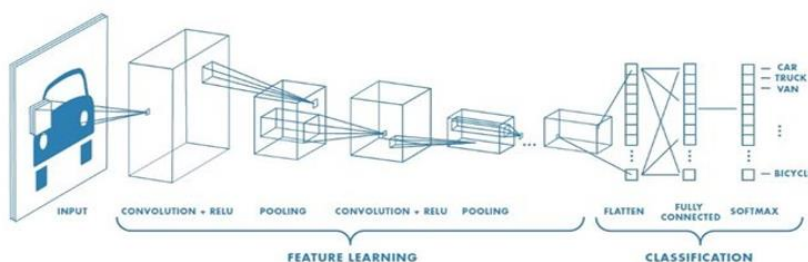


Gambar 2. Alur Penyelesaian Masalah

Gambar 2 memperlihatkan alur penyelesaian masalah yang mencakup beberapa tahapan diantaranya, pengumpulan dataset, augmentasi dataset, pelabelan dataset, split dataset, konversi XML ke CSV, konversi CSV ke TFRecord, perancangan model CNN, implementasi dan pelatihan model, evaluasi model, konversi ke TFLite, integrasi model ke aplikasi, dan pengujian aplikasi.

2.4. Alur Arsitektur Algoritma CNN

Arsitektur algoritma Convolutional Neural Network (CNN) terdiri dari tiga lapisan syaraf utama yaitu convolutional layer, pooling layer, dan fully-connected layer[5]. Ketiga lapisan itu memiliki fungsinya masing-masing, yaitu untuk mengekstraksi ciri dari citra masukan dan untuk mengklasifikasikan citra. Ini merupakan suatu kelebihan dari metode CNN karena bisa sekaligus melakukan pengekstraksian ciri dan pengklasifikasian citra hanya dengan menggunakan 1 metode.

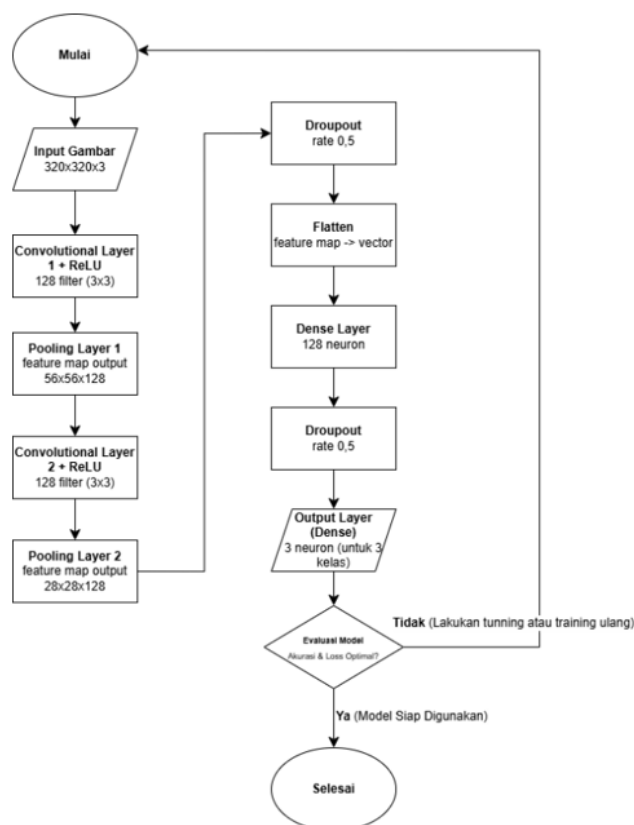


Gambar 3. Arsitektur Algoritma CNN

Gambar 3 memperlihatkan alur arsitektur algoritma CNN yang terdiri dari beberapa lapisan. *Input Layer* sebagai data masukan untuk selanjutnya akan dilakukan pemrosesan. *Convolutional Layer* sebagai lapisan yang digunakan untuk melakukan operasi konvolusi pada *output layer* sebelumnya. Aktivasi ReLU sebagai lapisan aktivasi pada CNN yang berada diantara *Convolutional Layer* dan *Pooling Layer*. *Pooling Layer* sebagai lapisan yang berfungsi untuk mengurangi ukuran spasial dari fitur konvolusi sehingga dapat mengurangi sumber daya komputasi yang dibutuhkan, sehingga mempercepat komputasi[9]. *Flatten Layer* sebagai langkah untuk mengubah array 2 dimensi menjadi 1 dimensi atau vektor tunggal, *Fully Connected Layer* sebagai lapisan yang digunakan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Aktivasi Softmax sebagai fungsi yang digunakan dalam perhitungan probabilitas untuk menentukan klasifikasi multi kelas dengan output kelas yang mempunyai nilai probabilitas yang paling tinggi[10], [11], [12], [13], [14].

2.5. Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program [15]. Flowchart dibuat agar mudah dipahami dan dijelaskan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.



Gambar 4. Flowchart Algoritma CNN

3. HASIL DAN PEMBAHASAN

Pada tahap ini dilakukan tahap implementasi dan pengujian untuk memastikan bahwa sistem sudah berjalan sesuai yang di rencanakan.

3.1. Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari citra gambar daun alpukat yang diambil secara manual menggunakan kamera smartphone. Dataset ini mencakup tiga jenis daun alpukat, yaitu Aligator, Kendil, dan Mentega dengan total dataset mencapai 4.800 gambar.

Tabel 1. Tabel Informasi Dataset

Dataset Asli	Jumlah Dataset Augmentasi	Jumlah Training	Jumlah Validation	Jumlah Testing

105	4.800	1.280	480	480
-----	-------	-------	-----	-----

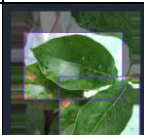
3.2. Augmentasi

Ukuran Proses ini melibatkan pembuatan gambar baru dari gambar asli dengan menerapkan transformasi tertentu seperti resize gambar ke 320x320, rotasi, flipping (membalik), zoom, pergeseran posisi, perubahan tingkat kecerahan, distorsi bentuk dan grayscale. Dalam penelitian ini, augmentasi dilakukan menggunakan tools jupyter anaconda dengan pustaka ImageDataGenerator dari Keras, yang dijalankan di penyimpanan local Setiap kelas daun (misalnya Kendil, Aligator, dan Mentega) ditargetkan memiliki total 1600 gambar.

3.3. Pelabelan Dataset

Pelabelan dataset adalah pemberian anotasi atau label yang akurat pada setiap gambar dalam dataset untuk memberitahu model objek apa saja yang ada di dalam gambar tersebut dan di mana lokasinya. Pada penelitian ini, dataset di unggah ke platform Roboflow untuk dilakukan proses pelabelan. Dataset yang telah dilakukan pelabelan di ekspor ke dalam bentuk formal Pascal VOC XML untuk menyimpan informasi anotasi gambar.

Tabel 2. Tabel Proses Pelabelan di Roboflow

No	Jenis Daun Alpukat	Pelabelan
1.	Daun alpukat aligator	
2.	Daun alpukat kendil	
3.	Daun alpukat mentega	

3.4. Split Dataset

Setelah ekspor ke Pascal VOC XML adalah mengekstrak file arsipnya. Setelah terekstrak, dataset gambar dan folder anotasi pasangannya kemudian dibagi menjadi tiga bagian: data training (80%), data validation (10%), dan data testing (10%). Setelah proses pemisahan (splitting) dataset selesai, file anotasi yang masih dalam format Pascal VOC XML perlu dikonversi menjadi format CSV (Comma-Separated Values) untuk setiap set data (training, validation, dan testing) dan create label_map.pbtxt.

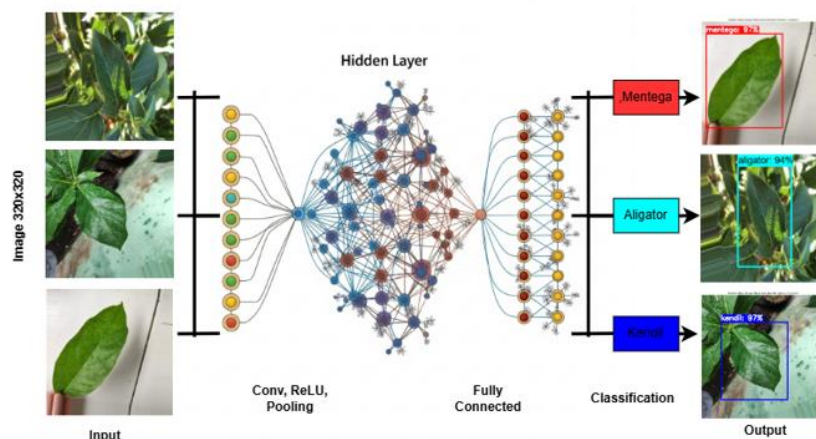
3.5. Konversi CSV ke TFRecord

Setelah file anotasi dalam format CSV (“train_labels.csv”, “val_labels.csv”, dan “test_labels.csv”) berhasil dibuat dari data XML, langkah selanjutnya adalah mengonversi CSV ke format TFRecord, yang biasanya memiliki ekstensi record. Dengan mengonversi ke format TFRecord, proses pelatihan model deteksi objek menggunakan TensorFlow dapat berjalan lebih lancar, cepat, dan efisien, memanfaatkan sepenuhnya kapabilitas yang ditawarkan oleh ekosistem TensorFlow.

3.6. Konfigurasi pipeline.config

File pipeline.config adalah inti dari proses pelatihan dan valuasi model menggunakan TensorFlow Object Detection API. File ini mendefinisikan seluruh arsitektur model, parameter pelatihan, serta konfigurasi dataset input untuk tahap pelatihan dan evaluasi. Pada file pipeline.config menggunakan pre-trained model ssd_mobilenet_v2_fpn-lite_320x320_coco17_tpu-8. Konfigurasi pipeline.config ini secara keseluruhan disesuaikan untuk melatih model deteksi objek SSD MobileNetV2 FPN Keras pada dataset kustom yang terdiri dari 3 kelas daun alpukat (aligator, kendil, dan mentega) dengan gambar input berukuran 320x320 piksel. Pelatihan dilakukan selama 50.000 steps dengan memanfaatkan fine-tuning dari checkpoint deteksi pre-trained untuk mempercepat konvergensi dan meningkatkan kinerja model.

3.7. Pemodelan Convolutional Neural Network



Gambar 5. Pemodelan Jaringan CNN

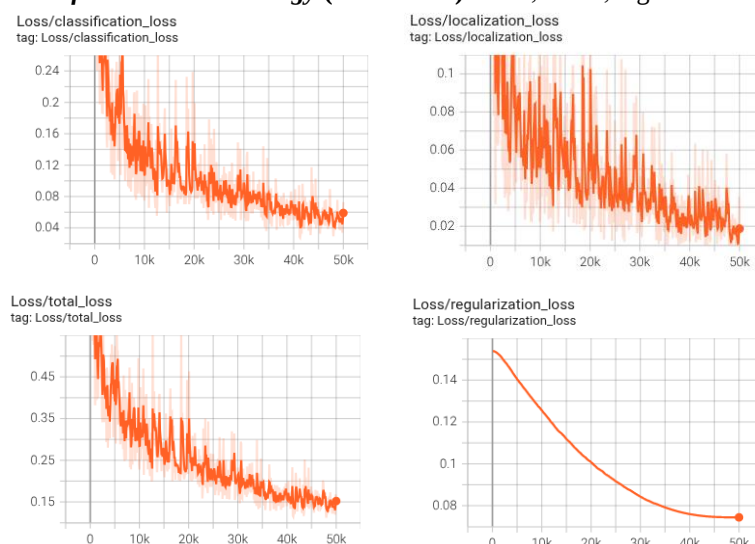
gambar 5 menampilkan alur kerja atau arsitektur dari sebuah jaringan Convolutional Neural Network (CNN) yang digunakan untuk deteksi objek pada 3 jenis daun alpukat (alligator, kendil, dan mentega). Komponen-komponen utama dalam arsitektur ini meliputi: (1) Input untuk deteksi merupakan atau dataset daun yang akan diproses oleh model untuk dilakukan deteksi. Pada sisi kiri ditampilkan tiga contoh gambar daun sebagai input yang mengindikasikan bahwa inputan ini telah diubah ukurannya menjadi 320x320 piksel, yang kemungkinan merupakan citra RGB. (2) Representasi Neuron Input & Proses Awal (Conv, ReLU, Pooling), setelah gambar input, terdapat visualisasi serangkaian node atau neuron input (lingkaran berwarna). Informasi dari gambar input kemudian diproses melalui tahapan awal CNN yang terdiri dari lapisan Konvolusi (Conv) untuk ekstraksi fitur, fungsi aktivasi ReLU untuk memperkenalkan non-linearitas, dan lapisan Pooling untuk mereduksi dimensi spasial dari *feature map*. (3) Fully Connected Layer, setelah fitur-fitur diekstraksi dan diproses melalui Hidden Layer, data kemudian diteruskan ke lapisan Fully Connected. Pada lapisan ini, setiap neuron terhubung ke semua neuron dari lapisan sebelumnya. Lapisan ini biasanya berfungsi untuk melakukan klasifikasi berdasarkan fitur-fitur yang telah dipelajari. (4) Classification, tahap ini menggambarkan bagaimana model mengklasifikasikan objek yang terdeteksi ke dalam kelas-kelas yang telah ditentukan. Pada gambar, ini diilustrasikan dengan blok-blok berwarna yang masing-masing merepresentasikan kelas objek yang berbeda, contoh yaitu "Aligator" (biru muda), "Mentega" (biru tua), dan "Mentega" (hijau muda). (5) Output menampilkan hasil dari proses deteksi objek pada gambar-gambar input. Setiap objek daun yang terdeteksi akan diberikan kotak pembatas (bounding box) di sekelilingnya, beserta label kelas yang diprediksi oleh model dan seringkali disertai dengan skor kepercayaan.

3.8. Training Dataset

Tahapan training dataset adalah tahapan penting dalam pembuatan model object detection, untuk identifikasi jenis daun alpukat alligator, kendil, dan mentega. Proses ini bertujuan agar *neural network* dapat mempelajari pola-pola spesifik dari setiap jenis daun sehingga mampu menghasilkan model deteksi objek dengan tingkat akurasi yang tinggi. Dengan menjalankan `model_main_tf2.py`, TensorFlow Object Detection API akan mulai melatih model berdasarkan konfigurasi `pipeline.config` yang telah ditentukan, menggunakan dataset daun alpukat untuk mempelajari cara mengidentifikasi jenis alligator, kendil, dan mentega. Proses training model akan menghasilkan checkpoint yang berupa grafik tensor. Grafik ini menyimpan seluruh riwayat dan informasi terkait sesi training yang sudah dijalankan. Setelah semua tahap pelatihan selesai, Selanjutnya adalah mengubah grafik tensor tersebut menjadi sebuah grafik inferensi (inference graph). Setelah membuat `save_model.pb`, langkah selanjutnya adalah mengubah *TensorFlow SavedModel* menjadi TFLite Model. Proses ini menggunakan *TensorFlowLite Converter* karena berfungsi untuk mengonversi model ke format TFLite. Setelah model TensorFlow TFLite (`detect.tflite`) berhasil dibuat, selanjutnya mengubahnya menjadi model metadata TFLite. Sebelum proses konversi ini dilakukan, buat terlebih dahulu file bernama "labels.txt". Penting untuk dicatat bahwa file "labels.txt" ini berbeda dengan file "label_map.pbtxt". Perbedaannya terletak pada isinya, di mana "labels.txt" secara spesifik hanya memuat daftar nama-nama kelas.

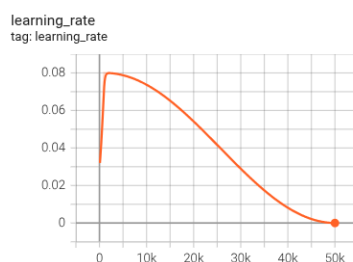
3.9. Hasil Training

Proses training model dilakukan sebanyak 50.000 steps dan `batch_size` yang digunakan sebesar 8 `batch_size` menggunakan dataset pelatihan yang berisi 1.280 gambar tersimpan dalam folder training. Proses training model menggunakan GPU dan berjalan selama 2 jam.



Gambar 6. Grafik Loss pada Training Model

Gambar 6 grafik loss tersebut secara kolektif menunjukkan bahwa proses pelatihan model machine learning sebanyak 50.000 steps berjalan dengan baik. Pada grafik Loss/classification_loss, kurva ini menunjukkan penurunan nilai kerugian dalam mengklasifikasikan objek. Nilai loss turun secara signifikan dari puncaknya di sekitar 0.24 pada awal pelatihan menjadi stabil di angka ~ 0.05 pada akhir 50.000 steps. Penurunan paling drastis memang terjadi dalam 10.000 steps pertama, yang menunjukkan model cepat belajar fitur-fitur dasar untuk klasifikasi. Penurunan paling signifikan terjadi dalam 5.000 steps pertama. Grafik Loss/localization_loss merepresentasikan kesalahan model dalam menentukan lokasi (bounding box) objek. Terjadi penurunan yang sangat baik, dari nilai awal di atas 0.10 menjadi sangat rendah di sekitar ~ 0.02 menjelang akhir pelatihan. Ini menandakan akurasi model dalam melokalisasi objek semakin tinggi. Grafik Loss/regularization_loss menunjukkan bagaimana model mengelola kompleksitasnya untuk mencegah *overfitting*. Nilai loss menurun secara konsisten dari ~ 0.15 di awal hingga mencapai ~ 0.075 pada 50.000 steps. Penurunan yang stabil ini menandakan bahwa teknik regularisasi bekerja efektif selama proses pelatihan. Grafik Loss/total_loss, menunjukkan tren menurun yang kuat dan menjadi indikator utama keberhasilan pelatihan. Nilai total loss turun dari puncaknya di atas 0.50 pada awal pelatihan menjadi sekitar ~ 0.15 pada akhir 50.000 steps. Penurunan ini mengindikasikan bahwa model berhasil belajar dari data, meningkatkan kemampuannya dalam melakukan klasifikasi, dan menentukan lokasi objek dengan baik, serta mengelola kompleksitasnya untuk mencegah *overfitting*.

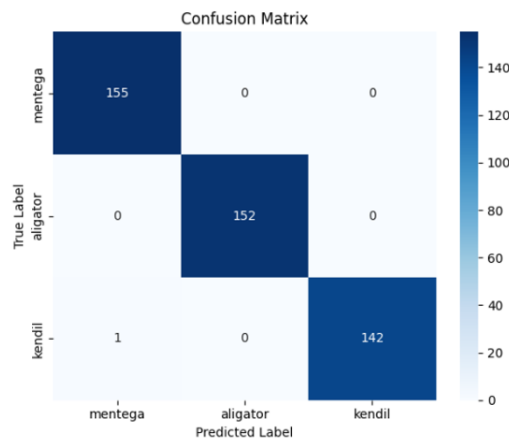


Gambar 7. Grafik Learning Rate pada Training Model

Gambar 7 merupakan grafik learning rate pada proses training model yang berlangsung selama 50.000 steps. Pada permulaan training (sekitar 0 hingga 5.000 steps), learning rate dimulai dari nilai yang relatif rendah (sekitar 0.03) dan kemudian meningkat secara signifikan hingga mencapai puncaknya di sekitar 0.08 dengan tujuan untuk mencegah model melakukan perubahan parameter yang terlalu drastis di awal training ketika bobot model masih diinisialisasi secara acak. Setelah mencapai puncaknya, learning rate mulai menurun secara bertahap dan konsisten seiring bertambahnya jumlah steps (dari 5.000 hingga 50.000 steps) bertujuan untuk memungkinkan model melakukan penyesuaian yang lebih halus terhadap parameter seiring dengan semakin dekatnya model ke solusi optimal. Learning rate yang semakin kecil di akhir training membantu model untuk konvergen secara stabil ke titik minimum yang lebih baik.

3.10. Confusion Matrix

Confusion matrix adalah tabel evaluasi untuk model klasifikasi. Berfungsinya untuk menampilkan secara terperinci perbandingan hasil prediksi model dengan kondisi sebenarnya, sehingga mudah untuk melihat perbandingan data yang diklasifikasi secara benar dan salah oleh model tersebut. Dalam proses pengujian ini, sebanyak 480 gambar dialokasikan untuk set data pengujian. Hasil dari pengujian tersebut kemudian divisualisasikan menggunakan gambar confusion matrix yang terlampir di bawah ini.



Gambar 8. Confusion Matrix

Berikut adalah interpretasi detail dari matriks tersebut:

Total Data Uji:

- Baris "kendil": 140 TP + 1 FP + 0 FN = 141
- Baris "mentega": 0 FP + 168 TP + 1 FN = 169
- Baris "aligator": 0 FN + 0 FP + 167 TP = 167

Analisis Per Kelas:

- Kelas "kendil":
 TP = 140, FP = 1 (dari mentega), FN = 0
 Precision: $\frac{TP}{TP+FP} = \frac{140}{140+1} = \frac{140}{141} = 0,9928 = 0,99$
 Recall: $\frac{TP}{TP+FN} = \frac{140}{140+0} = \frac{140}{140} = 1.0$
 F1-Score: $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{0,99 \times 1.00}{0,99 + 1.00} = 0,9949 = 1.0$
- Kelas "mentega":
 TP = 168, FP = 0, FN = 0
 Precision: $\frac{TP}{TP+FP} = \frac{168}{168+0} = \frac{168}{168} = 1.0$
 Recall: $\frac{TP}{TP+FN} = \frac{168}{168+1} = \frac{168}{169} = 0,994 = 0,99$
 F1-Score: $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{1.00 \times 0,99}{1.00 + 0,99} = 0,9949 = 1.0$
- Kelas "aligator":
 TP = 167, FP = 0, FN = 0
 Precision: $\frac{TP}{TP+FP} = \frac{167}{167+0} = \frac{167}{167} = 1.0$
 Recall: $\frac{TP}{TP+FN} = \frac{167}{167+4} = \frac{167}{171} = 0,976 = 0,98$
 F1-Score: $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{1.00 \times 0,98}{1.00 + 0,98} = 0,989 = 0,99$

4. KESIMPULAN

Algoritma Convolutional Neural Network (CNN) telah berhasil diterapkan untuk mengklasifikasikan jenis tanaman alpukat berdasarkan bentuk daun. Model CNN dilatih menggunakan dataset yang terdiri dari 4.800 gambar daun alpukat (1.600 gambar per kelas untuk jenis aligator, kendil, dan mentega), yang dibagi menjadi data training (80%), data validasi (10%), dan data testing (10%). Pengujian model deteksi objek pada 480 gambar data testing menunjukkan kinerja yang sangat tinggi dengan akurasi keseluruhan mencapai 99%. Secara spesifik, untuk kelas "aligator" diperoleh presisi 1.00 dan recall 0.98; untuk kelas "kendil" diperoleh presisi 1.00 dan recall 0.99; dan untuk kelas "mentega" diperoleh presisi 0.99 dan recall 1.00.

DAFTAR PUSTAKA

- [1] P. W. K. Waluyo Sri, "Identifikasi Jenis Buah Alpukat Melalui Sistem Pakar Berbasis Dempster Shafer," *Jurnal BATIRSI*, vol. 8, no. 1, pp. 12–16, 2024.
- [2] bps.go.id, "Produksi Tanaman Buah-buahan, 2021-2023," bps.go.id.
- [3] F. Ayu, F. Sari, R. Wulanningrum, and L. S. Wahyuniar, "Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi) 1275 Penggunaan Metode CNN (Convolutional Neural Network) untuk Klasifikasi Jenis Tanaman Alpukat Berdasarkan Pola Daun," Online, 2023.
- [4] T. Sugiharto, S. Kom, and M. Eng, *PENGOLAHAN CITRA DIGITAL MENGGUNAKAN PROSES KONVOLUSI*. 2021. [Online]. Available: www.penerbitlitmus.co.id
- [5] T. Sugiharto *et al.*, *Pengolahan Citra Digital dan Deteksi Objek*. Malang: PT Literasi Nusantara Abadi Grup, 2025. [Online]. Available: www.penerbitlitmus.co.id
- [6] S. Nurhasanah Nugraha, R. Pebrianto, and E. Fitri, "Penerapan Deep Learning Pada Klasifikasi Tanaman Paprika Berdasarkan Citra Daun Menggunakan Metode CNN," *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS*, vol. 8, no. 2, pp. 133–142, 2023.
- [7] A. H. Nasrullah and H. Annur, "Implementasi Metode Convolutional Neural Network Untuk Identifikasi Citra Digital Daun," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 2, p. 726, Apr. 2023, doi: 10.30865/mib.v7i2.5962.
- [8] S. Arnandito and T. B. Sasongko, "Comparison of EfficientNetB7 and MobileNetV2 in Herbal Plant Species Classification Using Convolutional Neural Networks," 2024. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [9] T. Sugiharto, Saparudin, and W. Fawwaz Al Maki, "Indonesian Cued Speech Transliterate System Using Convolutional Neural Network MobileNet," in *2024 Ninth International Conference on Informatics and Computing (ICIC)*, Medan, Indonesia: IEEE, 2024, pp. 1–7. doi: 10.1109/ICIC64337.2024.10957117.
- [10] D. Gunawan and H. Setiawan, "Convolutional Neural Network dalam Analisis Citra Medis," 2022.
- [11] A. A. Santosa, R. Y. N. Fu'adah, and S. Rizal, "Deteksi Penyakit pada Tanaman Padi Menggunakan Pengolahan Citra Digital dengan Metode Convolutional Neural Network," *JOURNAL OF ELECTRICAL AND SYSTEM CONTROL ENGINEERING*, vol. 6, no. 2, pp. 98–108, Feb. 2023, doi: 10.31289/jesce.v6i2.7930.
- [12] M. Rafly Alwanda, R. Putra, K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," 2020.
- [13] J. Khatib Sulaiman, S. Gunawan Ramdhani, E. Itje Sela, and U. Teknologi Yogyakarta, "Implementasi Face Recognition Untuk Sistem Presensi Universitas Menggunakan Convolutional Neural Network," *Indonesian Journal of Computer Science Attribution*, vol. 12, no. 6, pp. 2023–4098.
- [14] N. A. Purwitasari and M. Soleh, "Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Processing (Implementation Of Artificial Neural Network Algorithm In Chatbot Development Using Natural Language Processing Approach)," Mar. 2022.
- [15] A. Zalukhu *et al.*, "PERANGKAT LUNAK APLIKASI PEMBELAJARAN FLOWCHART," *Jurnal Teknologi Informasi dan Industri*, vol. 4, no. 1, 2023.