

PENGEMBANGAN APLIKASI ANDROID UNTUK KLASIFIKASI TINGKAT KEMATANGAN BUAH ALPUKAT MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) BERDASARKAN PENGOLAHAN CITRA

Bagaskara Dwi Rizky¹⁾, Arnes Sembiring²⁾, I Gusti Prahmana³⁾, Saifuddin Muhammad Jalil⁴⁾

^{1,2,3}Program Studi Teknik Informatika, ⁴Program Studi Teknik Industri

^{1,3}STMIK Kaputama, Binjai, Indonesia, ²Universitas Medan Area, Medan, Indonesia,

⁴Universitas Malikussaleh, Aceh Utara, Indonesia

^{1,3}Jl. Veteran No.4A, Tangsi, Kec. Binjai Kota, Kota Binjai, Sumatera Utara 20714, ²Jalan Sei Serayu Nomor 70 A / Jalan Setia Budi Nomor 79 B, Medan, ⁴Cot Tengku Nie Reuleut, Muara Batu, Aceh Utara, Provinsi Aceh, Indonesia

email: bagaskaradwirizky7@gmail.com¹⁾, arnessembiring@staff.uma.ac.id²⁾,
igustiprahmana4@gmail.com³⁾, saifuddin@unimal.ac.id⁴⁾

Abstrak

Penelitian ini bertujuan untuk mengembangkan aplikasi Android berbasis pengolahan citra yang mampu mengklasifikasi tingkat kematangan buah alpukat secara otomatis. Permasalahan utama dalam proses klasifikasi manual adalah ketidakakuratan dan ketidakkonsistenan hasil, yang dapat berdampak pada kualitas dan nilai jual buah. Untuk mengatasi hal tersebut, diterapkan metode Convolutional Neural Network (CNN) dengan arsitektur MobileNetV2 yang ringan dan efisien untuk perangkat mobile. Dataset citra buah alpukat dibagi ke dalam lima kelas, yaitu Mentah, Setengah Matang, Matang, Terlalu Matang, dan Bukan Alpukat. Proses pelatihan model mencakup augmentasi data, pengoptimalan fungsi aktivasi, serta konversi model ke format TensorFlow Lite agar dapat diintegrasikan ke dalam aplikasi Android. Hasil pengujian menunjukkan bahwa model CNN mampu mencapai tingkat akurasi validasi sebesar 94,61% dan akurasi pengujian sebesar 90,73%. Aplikasi ini memungkinkan pengguna untuk mengambil atau mengunggah gambar buah alpukat, lalu memprosesnya secara real-time dan menampilkan prediksi tingkat kematangannya. Penelitian ini memberikan kontribusi dalam pengembangan teknologi pertanian presisi berbasis kecerdasan buatan untuk membantu proses klasifikasi buah secara cepat, akurat, dan efisien.

Kata Kunci: CNN, Klasifikasi Kematangan Buah Alpukat, Fungsi Optimasi, Mobilenetv2, aplikasi Android, TensorFlow Lite

1. Pendahuluan

Buah alpukat (*Persea americana*) merupakan salah satu komoditas hortikultura yang memiliki nilai ekonomi tinggi dan semakin banyak dikonsumsi masyarakat, khususnya di wilayah tropis seperti Indonesia. Kandungan gizinya yang melimpah, seperti lemak tak jenuh, vitamin E, dan berbagai mineral menjadikan alpukat digemari baik oleh pasar domestik maupun mancanegara [1]. Peningkatan permintaan ini menuntut para petani untuk dapat menyediakan buah alpukat dengan kualitas terbaik, salah satunya ditentukan oleh tingkat kematangan buah sebelum dipasarkan.

Tingkat kematangan alpukat merupakan aspek kritis dalam menentukan mutu buah. Alpukat yang belum matang akan memiliki tekstur keras dan rasa hambar, sedangkan yang terlalu matang cenderung mudah rusak dan tidak tahan lama [2]. Umumnya, indikator visual seperti warna dan tekstur kulit buah digunakan oleh petani atau pedagang untuk menilai kematangan, namun metode ini sangat bergantung pada pengalaman masing-masing individu, sehingga bersifat subjektif dan tidak selalu konsisten [1].

Seiring dengan perkembangan teknologi, khususnya di bidang pengolahan citra digital dan kecerdasan buatan (Artificial Intelligence), kini tersedia pendekatan yang lebih objektif dan efisien. Salah satu metode yang terbukti efektif dalam analisis visual adalah Convolutional Neural Network (CNN). CNN mampu mengenali pola visual kompleks dan mengekstraksi fitur penting dari gambar, seperti warna dan tekstur permukaan kulit buah alpukat [3].

Sejumlah penelitian telah menunjukkan bahwa CNN sangat efektif dalam tugas klasifikasi berbasis gambar di bidang pertanian, termasuk pengklasifikasian kematangan buah. Dengan memanfaatkan citra digital buah alpukat dari berbagai tingkat kematangan, CNN dapat dilatih untuk mengenali perbedaan visual yang signifikan antar kategori. Proses klasifikasi ini dapat dilakukan secara otomatis dan konsisten, sehingga lebih akurat dibandingkan metode konvensional [4].

Penerapan teknologi CNN dalam klasifikasi tingkat kematangan buah tidak hanya memberikan efisiensi bagi pelaku pertanian, tetapi juga membantu mengurangi kerugian akibat kesalahan panen dan memperkecil potensi

limbah buah yang tidak layak konsumsi. Hasil klasifikasi dapat digunakan untuk mendukung pengambilan keputusan dalam proses panen, distribusi, hingga penanganan pascapanen yang lebih optimal [2].

Penelitian yang berjudul “Mendeteksi Kematangan Buah Kelapa Sawit Menggunakan Convolutional Neural Network Deep Learning”. Penelitian ini menggunakan metode Convolutional Neural Network karena CNN memiliki kemampuan yang luar biasa untuk mengekstraksi fitur dari gambar secara otomatis tanpa menggunakan metode ekstraksi fitur manual. CNN digunakan untuk mendeteksi kematangan buah kelapa sawit dengan arsitektur MobileNet, yang dikenal sebagai arsitektur ringan dengan efisiensi tinggi dalam klasifikasi gambar. Hasil penelitian tersebut menunjukkan bahwa sistem yang dirancang dapat mengklasifikasikan buah kelapa sawit menjadi tiga kelas kematangan: mentah, matang, dan busuk, dengan akurasi pelatihan dan pengujian mencapai 100%. Penelitian ini menunjukkan bahwa CNN dapat digunakan secara efektif untuk mendeteksi kematangan buah kelapa sawit dengan hasil yang sangat akurat dan stabil, khususnya dengan arsitektur MobileNet [5].

Penelitian selanjutnya yang berjudul “Pendeteksi Kematangan Mangga Berbasis Fitur Analisis Warna Dengan Metode CNN”. Penelitian ini menggunakan metode Convolutional Neural Network (CNN) dan OpenCV untuk mengidentifikasi kematangan buah mangga dengan menggunakan analisis warna kulit. Hasil penelitian menunjukkan bahwa teknik CNN dan OpenCV dapat dengan akurat mengklasifikasikan kematangan mangga. Sistem yang dibuat dapat menggantikan penilaian manual, yang seringkali tidak konsisten dan memakan waktu. Dengan kemampuan CNN dalam mengenali pola visual dan keunggulan OpenCV dalam pra-pemrosesan gambar, sistem ini dapat diintegrasikan dengan mudah ke dalam proses produksi tanpa memerlukan investasi pada peralatan tambahan yang mahal. Hal ini membuatnya pilihan yang bagus untuk industri pertanian untuk pemilahan buah otomatis [6].

Berdasarkan uraian pada bagian latar belakang, penulis mengajukan sebuah penelitian yang berjudul “Klasifikasi Kematangan Buah Alpukat Menggunakan Convolutional Neural Network berbasis Android”. Penelitian ini dilandasi oleh pemanfaatan teknologi deep learning dan pengolahan citra digital, dengan fokus pada penerapan metode CNN untuk mengklasifikasikan tingkat kematangan alpukat berdasarkan ciri visual kulit buah. Penggunaan metode CNN, khususnya arsitektur MobileNetV2 yang telah dikonversi ke dalam format TensorFlow Lite (.tflite), memungkinkan proses inferensi dapat dijalankan langsung di perangkat Android secara efisien dan cepat, tanpa memerlukan koneksi internet [7].

Tujuan dari pengembangan sistem ini adalah untuk memberikan kemudahan bagi petani, pedagang, serta pelaku usaha di bidang industri dalam mengetahui kondisi kematangan buah alpukat secara lebih cepat dan akurat. Dengan sistem klasifikasi ini, mereka dapat menentukan waktu terbaik untuk mengolah, menyimpan, atau memasarkan buah, sehingga kerugian akibat pembusukan atau limbah buah dapat diminimalkan. Selain itu, sistem ini juga berpotensi meningkatkan efisiensi dalam manajemen pascapanen dan distribusi produk.

2. Landasan Teori

MobileNetV2

MobileNetV2 merupakan salah satu arsitektur dari jaringan saraf konvolusional (CNN) yang dirancang untuk menghasilkan model ringan dan efisien, sehingga sangat sesuai diterapkan pada perangkat bergerak seperti smartphone. Keunggulan dari arsitektur ini terletak pada penggunaan blok inverted residual dan linear bottleneck, yang memungkinkan pengolahan data dilakukan secara lebih cepat dan hemat memori tanpa mengorbankan akurasi. Arsitektur ini telah banyak digunakan dalam berbagai studi klasifikasi citra karena kemampuannya mengenali pola visual yang kompleks dengan sumber daya komputasi yang minimal [8].

Dalam penelitian ini, MobileNetV2 diterapkan menggunakan pendekatan transfer learning, yaitu teknik yang memanfaatkan model yang telah dilatih sebelumnya pada dataset besar seperti ImageNet untuk menyelesaikan tugas klasifikasi baru [9]. Dengan strategi ini, peneliti hanya perlu menyesuaikan lapisan akhir dari model agar sesuai dengan jumlah kelas pada dataset, yaitu mentah, setengah matang, dan matang. Proses fine-tuning dilakukan pada lapisan tertentu agar model dapat beradaptasi terhadap karakteristik unik dari gambar kulit alpukat. Pendekatan ini sangat berguna ketika jumlah data pelatihan terbatas namun tetap menginginkan performa yang tinggi.

Penggunaan MobileNetV2 dengan metode transfer learning terbukti efektif dalam berbagai penelitian di Indonesia, khususnya untuk klasifikasi di bidang pertanian. Sebagai contoh, dalam penelitian tersebut [10], MobileNetV2 berhasil digunakan untuk mengklasifikasi penyakit daun kopi menjadi tiga kategori sehat, miner, dan rust yang menunjukkan akurasi sempurna 100 % pada skema pembagian data latihan dan uji 90:10. Keberhasilan ini membuktikan bahwa model ringan seperti MobileNetV2 dengan transfer learning dapat beroperasi secara optimal di perangkat dengan sumber daya terbatas, termasuk smartphone Android. Oleh karena itu, metode ini sangat sesuai dijadikan dasar dalam pengembangan sistem klasifikasi kematangan buah alpukat yang diusulkan dalam penelitian ini.

Fungsi Optimasi

Adam (Adaptive Moment Estimation) merupakan salah satu algoritma optimasi paling populer dalam deep learning karena kemampuannya untuk mempercepat proses pelatihan dan menghasilkan akurasi tinggi. Algoritma

ini menggabungkan dua teknik penting, yaitu momentum dan RMSProp. Momentum digunakan untuk mempercepat gradien ke arah yang tepat, sedangkan RMSProp membantu dalam menyesuaikan kecepatan pembelajaran berdasarkan kuadrat rata-rata gradien sebelumnya. Dalam sebuah penelitian [11] Adam terbukti memberikan hasil akurasi terbaik pada model CNN untuk klasifikasi ekspresi wajah dengan akurasi mencapai 68,6%, jauh mengungguli metode SGDM dan RMSProp. Selain itu, dalam studi klasifikasi sel darah putih, optimasi Adam menghasilkan akurasi sebesar 97,37% dan recall sebesar 97,43% yang menunjukkan konsistensi performanya yang tinggi. Hal ini menunjukkan bahwa Adam sangat efektif dalam menangani masalah dengan data kompleks dan fitur yang bervariasi.

Metrik Performa

Dalam mengukur performa model klasifikasi (seperti CNN), kita menggunakan beberapa metrik evaluasi. Setiap metrik memberikan sudut pandang berbeda tentang kualitas prediksi model. Berikut adalah penjelasannya:

a. Confusion Matrix

Confusion matrix adalah tabel yang menggambarkan performa model klasifikasi dengan membandingkan label aktual dengan prediksi model [12] seperti ditunjukkan di tabel 1. Tabel ini terdiri dari:

- TP (True Positive): Model benar memprediksi kelas positif.
- TN (True Negative): Model benar memprediksi kelas negatif.
- FP (False Positive): Model salah memprediksi positif padahal negatif.
- FN (False Negative): Model salah memprediksi negatif padahal positif.

Tabel 1. Rumus Confusion matrix

	Prediksi Positif	Prediksi Negatif
Aktual Positif	TP	FN
Aktual Negatif	FP	TN

b. Akurasi (Accuracy)

Akurasi menunjukkan seberapa sering model membuat prediksi yang benar. Cocok digunakan jika data seimbang [13].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

c. Presisi (Precision)

Presisi menunjukkan proporsi prediksi positif yang benar-benar tepat. Penting saat kita ingin meminimalkan kesalahan positif (False Positive) [12].

$$Precision = \frac{TP}{TP + FP}$$

d. Recall (Sensitivity)

Recall mengukur seberapa banyak data positif yang berhasil dikenali oleh model. Penting jika kita ingin meminimalkan kesalahan negatif (False Negative), misalnya untuk diagnosa penyakit [14].

$$Recall = \frac{TP}{TP + FN}$$

e. F1-Score

F1-score adalah rata-rata harmonis antara presisi dan recall. Cocok untuk kasus data tidak seimbang karena mempertimbangkan keseimbangan antara FP dan FN [15].

$$F1 - Score = \frac{2 \times Precision \times recall}{Precision + recall}$$

f. Loss

loss merupakan parameter dari tingkat keyakinan suatu sistem dalam memprediksi hasil yang salah. Fungsi loss melakukan perbandingan [16].

Pengolahan Citra

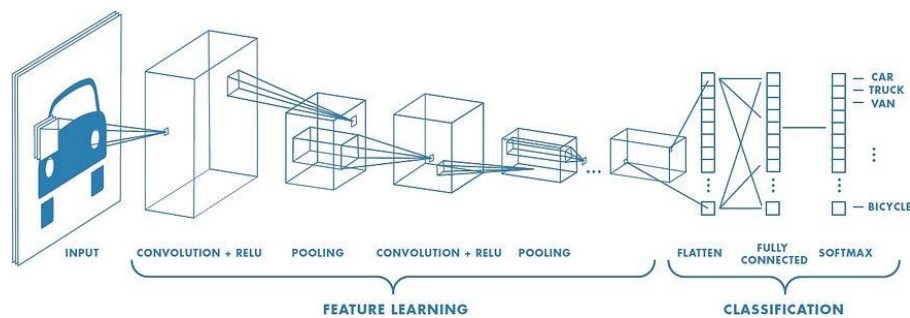
Pengolahan Citra (Digital Image Processing) adalah suatu proses yang menggunakan algoritma komputer untuk mengubah, menginterpretasikan, dan menganalisis gambar digital sehingga menghasilkan informasi yang bermanfaat. Citra digital sendiri adalah representasi visual dari objek nyata dalam bentuk matriks piksel dua dimensi dengan nilai intensitas yang berbeda untuk setiap piksel [17], [18].

Dalam penelitian ini, pengolahan gambar digunakan untuk membantu sistem komputer mengenali dan mengklasifikasikan tingkat kematangan buah alpukat. Secara umum, tujuan utama pengolahan gambar adalah untuk meningkatkan kualitas visual gambar dan mengekstrak informasi penting dari gambar tersebut. Teknologi ini memungkinkan sistem berbasis komputer untuk melakukan inspeksi yang lebih cepat, objektif, dan konsisten daripada metode yang biasanya dilakukan secara manual oleh manusia.

Menurut Rahmawati & Riana [19] Pengolahan citra sangat berguna untuk menemukan pola atau objek dalam gambar yang mungkin sulit diidentifikasi secara langsung, seperti perbedaan warna atau tekstur yang sangat halus. Oleh karena itu, teknologi ini menjadi sangat penting dalam banyak sektor, seperti pertanian, layanan kesehatan, dan keamanan.

Convolutional neural network

Convolutional neural network (CNN) adalah salah satu jenis neural network yang dibuat untuk menangani masalah pengolahan gambar. CNN dapat dengan mudah mengklasifikasikan, mengidentifikasi, dan mengenali pola dalam gambar. Karena arsitekturnya sesuai dengan cara otak manusia memproses gambar, CNN dapat memahami detail gambar dengan lebih baik. Data yang digunakan pada CNN adalah data dua dimensi, seperti citra atau suara, dan menggunakan operasi konvolusi dalam matriks dan sekumpulan bobot konvolusi empat dimensi. Karena sifat proses konvolusi ini CNN hanya dapat digunakan pada data yang memiliki struktur data dua dimensi [20].



Gambar 1. Arsitektur Dasar CNN

Pada Gambar 1 menunjukkan bahwa operasi CNN terdiri dari dua bagian utama: pembelajaran fitur dan klasifikasi. Tiga proses dilakukan dalam bagian pembelajaran fitur, yaitu [20].

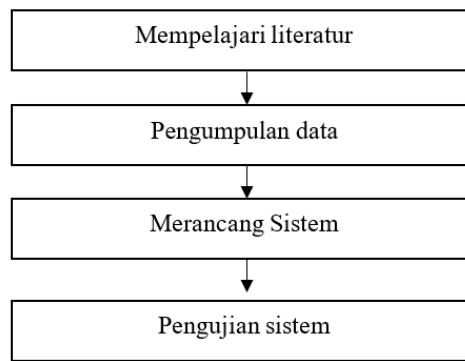
1. Operasi konvolusi (CNN) adalah proses utama yang bertujuan untuk mengekstrak fitur dari gambar masukan.
2. *ReLU (Rectified Linear Unit)* adalah fungsi aktivasi yang memiliki *output* 0 jika input kurang dari 0, Artinya, jika $x \leq 0$ maka $x = 0$ dan jika $x > 0$ maka $x = x$.
3. *Pooling*, yang digunakan untuk mengumpulkan informasi yang paling penting melalui operasi *down-sampling* dan mengurangi jumlah parameter yang ada pada peta fitur. *Max* dan *average pooling* adalah metode *pooling* yang paling umum digunakan.

Pada bagian klasifikasi, terdapat tiga proses yaitu [20] :

1. Flatten, layer ini mengubah output hasil dari lapisan konvolusi diubah menjadi satu kolom vector fitur untuk digunakan pada lapisan terhubung secara penuh.
2. Fully connected layer, Untuk melakukan klasifikasi, lapisan ini mirip dengan Artificial Neural Network (ANN).
3. Softmax, Untuk mendapatkan nilai probabilitas untuk setiap kelas, nilai probabilitas tertinggi adalah hasil prediksi kelas yang dihasilkan. Proses-proses tersebut merupakan prosedur yang sering dilakukan oleh berbagai model CNN.

3. Metode Penelitian

Adam Dalam melakukan penelitian ini, penulis mengikuti beberapa tahapan metodologi penelitian, agar tujuan penelitian dapat tercapai dengan baik. Adapun tahapan yang dilakukan dalam penelitian ini yaitu sebagai berikut :



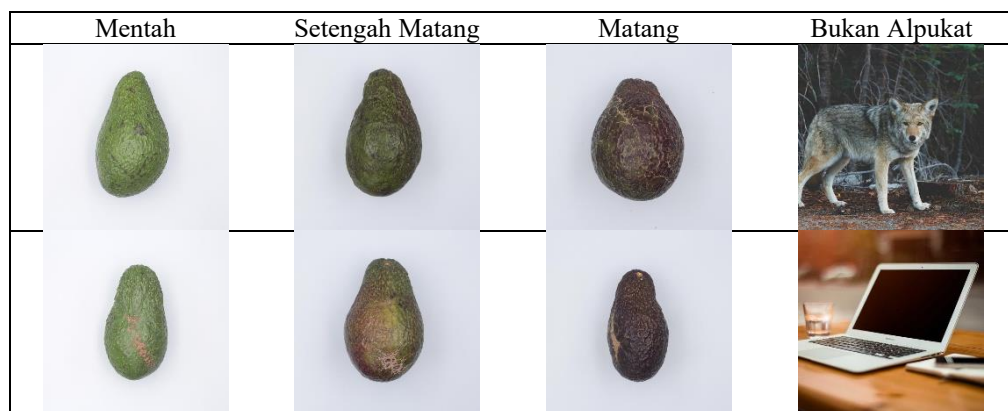
Gambar 2. Metodologi Penelitian

Untuk memperjelas struktur metodologi penelitian di atas, maka penulis membuat keterangannya sebagai berikut :

1. **Mempelajari literatur**
Tahap ini dilakukan mengumpulkan berbagai landasan-landasan teori yang berkaitan dengan metode *Convolutional Neural Network* (CNN), pengolahan citra digital, klasifikasi tingkat kematangan buah, serta pengembangan aplikasi Android. Dalam tahap ini, teori dikumpulkan dari beberapa sumber seperti buku-buku, jurnal, artikel dan referensi lainnya.
2. **Pengumpulan Data**
Pengumpulan data berupa *dataset* gambar buah alpukat yang dikategorikan berdasarkan tingkat kematangan, yaitu mentah, setengah matang, matang, dan terlalu matang. *Dataset* citra alpukat ini diperoleh dari *dataset* mendeley. *Dataset* tersebut berisi kumpulan gambar buah alpukat yang telah di kategorikan sesuai tingkat kematangannya.
3. **Merancang Sistem**
Pada tahap ini dilakukan perancangan sistem untuk masalah yang sedang diteliti. Ini dapat mencakup tahap merancang alur kerja sistem dan juga merancang desain *interface* atau tampilan tatap muka yang akan dibuat.
4. **Pengujian Sistem**
Pada tahap akhir, dilakukan serangkaian pengujian terhadap sistem yang telah dibuat, pengujian-pengujian dilakukan agar dapat menemukan kesalahan-kesalahan (*error*) pada sistem dan melakukan perbaikan-perbaikan yang diperlukan.

Dataset

Penelitian ini menggunakan dataset Mendeley [18] berupa citra buah alpukat yang telah diklasifikasikan ke dalam tiga kategori tingkat kematangan, yaitu mentah, setengah matang, matang dan bukan alpukat. Setiap kategori memiliki jumlah data yang seimbang, yaitu masing-masing sebanyak 2.756 gambar, sehingga total dataset berjumlah 11.024 gambar. Dataset tersebut selanjutnya dibagi ke dalam tiga kelompok utama, yaitu data pelatihan (train), data validasi (validation), dan data pengujian (test). Data pelatihan terdiri dari 7.201 gambar, masing-masing kelas berisi 1.800 gambar. Data validasi terdiri dari 1.543 gambar, masing-masing kelas 385 gambar. Begitu pula dengan data pengujian, sebanyak 1.543 gambar yang juga terbagi rata untuk empat kelas tersebut. Gambar 3 menunjukkan contoh citra yang mewakili setiap kelas pada dataset tersebut.





Gambar 3. Contoh Citra pada Dataset

Untuk meningkatkan akurasi dan kemampuan generalisasi model CNN, dilakukan tahapan praproses data sebelum pelatihan. Praproses ini mencakup teknik augmentasi data pada citra pelatihan, yang bertujuan untuk memperbanyak variasi data secara artifisial [21]. Beberapa bentuk augmentasi yang diterapkan antara lain rotasi acak hingga 30 derajat, pergeseran horizontal dan vertikal hingga 20% dari ukuran gambar, pergeseran bentuk (shear), perbesaran dan pengecilan gambar (zoom), serta pencerminan horizontal (horizontal flip). Augmentasi ini membantu model mengenali pola citra dari berbagai sudut pandang dan variasi posisi.

Selain augmentasi, seluruh citra baik untuk pelatihan, validasi, maupun pengujian juga melalui proses normalisasi, yaitu skala nilai piksel gambar dikonversi dari 0–255 menjadi 0–1. Proses ini penting untuk menyesuaikan data dengan format input model CNN serta mempercepat proses pelatihan. Dengan kombinasi augmentasi dan normalisasi ini, diharapkan model dapat dilatih secara lebih efektif untuk mengklasifikasikan tingkat kematangan buah alpukat berdasarkan citra kulitnya.

4. Kesimpulan

Penelitian ini bertujuan untuk mengembangkan sebuah sistem berbasis aplikasi Android yang mampu mengklasifikasikan tingkat kematangan buah alpukat secara otomatis menggunakan metode Convolutional Neural Network (CNN), khususnya dengan memanfaatkan arsitektur MobileNetV2. Dalam implementasinya, sistem ini menggabungkan proses akuisisi citra, klasifikasi gambar, serta antarmuka pengguna yang ramah untuk memudahkan pengguna akhir, seperti petani, pedagang umum dalam menentukan kualitas buah alpukat berdasarkan tingkat kematangannya. Hasil yang diperoleh dari pelatihan dan pengujian model, serta pengembangan aplikasi Android, dipaparkan secara mendalam dalam bab ini

Alur Modelling Pelatihan Model

Tahap pertama dalam proses ini adalah persiapan dan pembagian dataset. Dataset yang digunakan terdiri dari gambar buah alpukat yang telah diklasifikasikan ke dalam empat kategori tingkat kematangan, yaitu: mentah, setengah matang, matang, dan terlalu matang. Gambar-gambar ini disusun dalam struktur direktori berdasarkan kelas masing-masing. Selanjutnya, dataset dibagi menjadi tiga subset, yaitu data pelatihan (training) sebesar 70%, data validasi sebesar 15%, dan data pengujian (testing) sebesar 15%. Proses pembagian dan penyalinan data dilakukan secara otomatis dengan bantuan teknik multiprocessing agar efisien dalam penanganan file berjumlah besar.

Selanjutnya, dilakukan proses augmentasi data menggunakan ImageDataGenerator untuk memperluas dan memperkaya variasi data pelatihan. Augmentasi ini meliputi rotasi gambar, translasi horizontal dan vertikal, zoom, perubahan pencahayaan, dan flipping horizontal. Semua gambar juga dinormalisasi dengan rescale 1./255 dan diubah ukurannya menjadi 224×224 piksel agar sesuai dengan format input MobileNetV2. Untuk data validasi dan pengujian, hanya dilakukan proses rescaling tanpa augmentasi, agar evaluasi tetap obyektif.

Model yang digunakan adalah MobileNetV2 dengan pre-trained weights dari ImageNet. Model di-load tanpa bagian atasnya (include_top=False) agar dapat dikustomisasi untuk klasifikasi alpukat. Sebagian besar lapisan bawah model dibekukan (tidak dilatih), sementara hanya sekitar 40 lapisan terakhir yang difine-tune agar model dapat belajar pola baru dari dataset spesifik. Di atas lapisan backbone, ditambahkan beberapa lapisan: GlobalAveragePooling2D, Dropout, Dense dengan 256 neuron beraktivasi ReLU, serta Dense akhir dengan fungsi aktivasi softmax untuk mengklasifikasikan keempat kelas.

Model dikompilasi menggunakan optimizer Adam dengan learning rate kecil (1e-5) dan fungsi loss categorical_crossentropy karena digunakan untuk klasifikasi multikelas. Selama pelatihan, digunakan beberapa callback seperti ModelCheckpoint untuk menyimpan model terbaik, EarlyStopping untuk menghentikan pelatihan jika validasi tidak membaik, dan ReduceLROnPlateau untuk menurunkan laju pembelajaran secara adaptif. Masalah ketidakseimbangan kelas diatasi dengan menghitung class_weight secara otomatis berdasarkan distribusi data pelatihan, agar model tidak bias terhadap kelas mayoritas.

Model kemudian dilatih selama maksimum 100 epoch (atau lebih sedikit jika terjadi early stopping), menggunakan generator data yang sudah di-augment. Proses pelatihan dapat dilihat pada gambar dibawah 4.

```

[14] # Pelatihan model
history = model.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator,
    callbacks=[checkpoint, early_stopping, reduce_lr],
    class_weight=class_weights
)

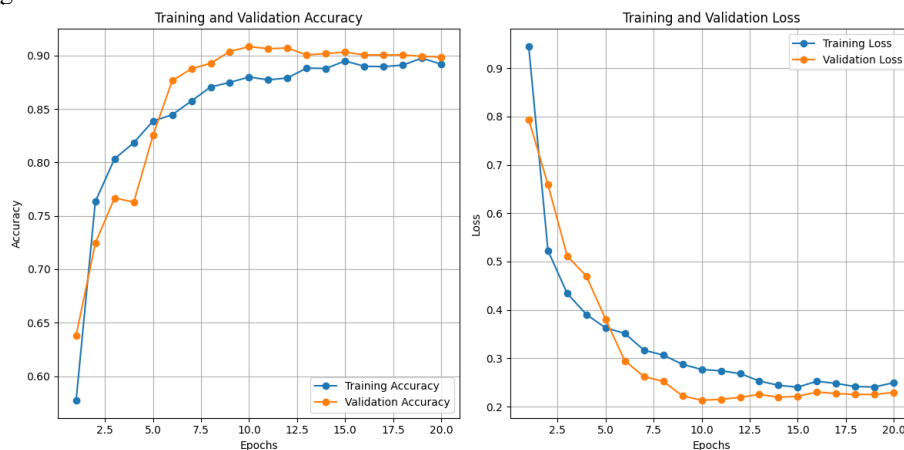
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your 'PyDataset' class should call 'super().__init__'
self._warn_if_super_not_called()

Epoch 1/100
225/225 - 0s 571ms/step - accuracy: 0.4462 - loss: 1.2490WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.
225/225 - 165s 629ms/step - accuracy: 0.4467 - loss: 1.2483 - val_accuracy: 0.6381 - val_loss: 0.7938 - learning_rate: 1.0000e-05
Epoch 2/100
225/225 - 0s 528ms/step - accuracy: 0.7496 - loss: 0.5624WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.
225/225 - 125s 555ms/step - accuracy: 0.7497 - loss: 0.5622 - val_accuracy: 0.7245 - val_loss: 0.6591 - learning_rate: 1.0000e-05
Epoch 3/100
225/225 - 0s 526ms/step - accuracy: 0.7995 - loss: 0.4475WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.
225/225 - 124s 553ms/step - accuracy: 0.7995 - loss: 0.4474 - val_accuracy: 0.7667 - val_loss: 0.5118 - learning_rate: 1.0000e-05
Epoch 4/100
225/225 - 0s 544ms/step - accuracy: 0.8131 - loss: 0.4010 - val_accuracy: 0.7628 - val_loss: 0.4695 - learning_rate: 1.0000e-05
Epoch 5/100
225/225 - 0s 521ms/step - accuracy: 0.8373 - loss: 0.3693WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.
225/225 - 123s 548ms/step - accuracy: 0.8373 - loss: 0.3692 - val_accuracy: 0.8252 - val_loss: 0.3807 - learning_rate: 1.0000e-05
Epoch 6/100
225/225 - 0s 512ms/step - accuracy: 0.8477 - loss: 0.3557WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.
225/225 - 122s 542ms/step - accuracy: 0.8477 - loss: 0.3557 - val_accuracy: 0.8765 - val_loss: 0.2947 - learning_rate: 1.0000e-05
Epoch 7/100
225/225 - 0s 514ms/step - accuracy: 0.8563 - loss: 0.3209WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.
225/225 - 122s 543ms/step - accuracy: 0.8563 - loss: 0.3209 - val_accuracy: 0.8876 - val_loss: 0.2619 - learning_rate: 1.0000e-05
Epoch 8/100
225/225 - 0s 521ms/step - accuracy: 0.8667 - loss: 0.3060WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.
225/225 - 128s 569ms/step - accuracy: 0.8667 - loss: 0.3068 - val_accuracy: 0.8928 - val_loss: 0.2524 - learning_rate: 1.0000e-05
Epoch 9/100

```

Gambar 4. Proses Pelatihan

Setelah proses pelatihan selesai, model dievaluasi menggunakan data testing untuk mengukur kemampuan generalisasi terhadap data yang belum pernah dilihat sebelumnya. Hasil evaluasi menunjukkan bahwa model memperoleh akurasi training sebesar 94.61% dan loss sebesar 0.1490. Sedangkan pada data testing, model mencapai akurasi sebesar 90.73%, yang menunjukkan bahwa model mampu mengklasifikasikan gambar alpukat ke dalam kelas Mentah, Setengah Matang, Matang, dan Bukan Alpukat dengan performa yang sangat baik. Hasil pelatihan divisualisasikan menggunakan grafik akurasi dan loss antara data pelatihan dan validasi yang ditunjukkan di gambar 5.



Gambar 5. Grafik akurasi training dan validasi

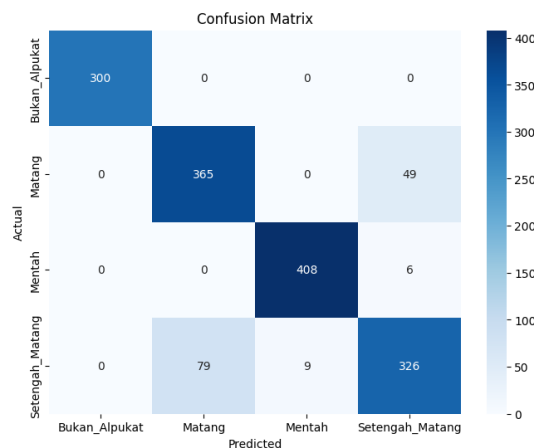
Berdasarkan grafik hasil pelatihan model, terlihat bahwa nilai akurasi dan loss pada data pelatihan (training) dan data validasi menunjukkan performa yang sangat baik. Grafik akurasi menunjukkan peningkatan yang stabil dari epoch ke epoch, di mana akurasi pelatihan meningkat dari sekitar 0,58 hingga mencapai lebih dari 0,90 pada akhir pelatihan. Hal yang sama juga terjadi pada akurasi validasi, yang mengalami lonjakan cepat hingga mencapai nilai di atas 0,91 sekitar epoch ke-9 dan kemudian stabil hingga akhir pelatihan. Kestabilan akurasi validasi ini menunjukkan bahwa model mampu mengenali pola baru dengan sangat baik, yang berarti model memiliki kemampuan generalisasi yang tinggi.

Sementara itu, grafik loss juga menunjukkan tren penurunan baik pada data pelatihan maupun validasi. Nilai loss pelatihan menurun dari sekitar 0,93 menjadi 0,24, sedangkan loss validasi juga menurun drastis dan stabil di angka sekitar 0,22 sejak epoch ke-9 hingga akhir. Tidak terlihat adanya lonjakan drastis atau ketidaksesuaian antara loss pelatihan dan validasi yang dapat mengindikasikan overfitting. Justru, nilai loss yang rendah dan konsisten ini mengindikasikan bahwa model berhasil mencapai konvergensi, yaitu titik optimal di mana model dapat belajar dengan efektif tanpa kehilangan kemampuan generalisasi.

Secara keseluruhan, model menunjukkan performa pelatihan yang sangat baik dengan akurasi validasi yang tinggi dan nilai loss yang rendah serta stabil. Hal ini mengindikasikan bahwa model yang dibangun mampu mengklasifikasikan tingkat kematangan alpukat dengan akurasi tinggi tanpa mengalami permasalahan overfitting atau underfitting. Dengan demikian, model ini sangat layak untuk digunakan pada implementasi nyata, seperti pada aplikasi klasifikasi gambar berbasis Android.

Untuk mengevaluasi performa klasifikasi secara lebih mendalam, dilakukan analisis confusion matrix dan classification report. Dari hasil prediksi pada data pengujian, diperoleh metrik seperti precision, recall, dan F1-

score untuk masing-masing kelas. Confusion matrix divisualisasikan dalam bentuk heatmap menggunakan Seaborn seperti dapat dilihat pada gambar 6.



Gambar 6. Grafik Confusion Matrix

Confusion matrix pada gambar menunjukkan performa klasifikasi model dalam mendeteksi tingkat kematangan buah alpukat berdasarkan citra. Baris pada matriks menunjukkan label sebenarnya (actual), sementara kolom menunjukkan hasil prediksi model (predicted). Dari hasil tersebut, terlihat bahwa kelas Bukan_Alpukat diklasifikasikan dengan sangat baik, dengan seluruh 300 sampel berhasil diprediksi secara tepat tanpa kesalahan. Untuk kelas Matang, terdapat 365 sampel yang diklasifikasikan dengan benar, namun masih terdapat 49 sampel yang justru diprediksi sebagai Setengah_Matang, sehingga menunjukkan adanya kebingungan model antara dua kelas yang memiliki tingkat kematangan yang berdekatan. Kelas Mentah memiliki performa klasifikasi yang sangat tinggi, yaitu 408 dari total sampel diklasifikasikan dengan benar, dan hanya 6 sampel yang salah diklasifikasikan sebagai Setengah_Matang.

Sedangkan pada kelas Setengah_Matang, dari total 414 sampel, 326 diklasifikasikan dengan benar, namun terdapat kesalahan klasifikasi yang cukup besar, yaitu 79 sampel diklasifikasikan sebagai Matang dan 9 sebagai Mentah. Hal ini menunjukkan bahwa kelas Setengah_Matang memiliki tantangan tersendiri karena cenderung tumpang tindih dengan karakteristik visual kelas Matang dan Mentah. Secara keseluruhan, model memiliki kemampuan klasifikasi yang sangat baik dengan akurasi pengujian mencapai 90,73%, namun tetap perlu dilakukan peningkatan akurasi khususnya untuk membedakan kelas yang berdekatan secara visual seperti Setengah_Matang dan Matang. Strategi seperti augmentasi data yang lebih spesifik atau pelatihan ulang dengan penekanan pada kelas yang lebih membingungkan bisa menjadi pendekatan lanjutan untuk memperbaiki performa model.

Langkah terakhir dalam proses ini adalah mengonversi model dari format Keras ke format TensorFlow Lite menggunakan TFLiteConverter. Proses ini juga dilengkapi dengan optimizations untuk memperkecil ukuran model sehingga lebih efisien saat digunakan di perangkat Android. Model hasil konversi disimpan sebagai file .tflite, yang kemudian dapat diintegrasikan ke dalam aplikasi mobile untuk melakukan klasifikasi tingkat kematangan alpukat.

Implementasi

Implementasi sistem merupakan tahap penerapan rancangan ke dalam bentuk nyata berupa aplikasi Android yang dapat digunakan oleh pengguna. Dalam pengembangan aplikasi “Alpukat Detection”, proses implementasi mencakup tampilan antarmuka pengguna (user interface) yang dirancang secara sederhana, intuitif, dan fungsional. Berikut adalah rincian implementasi berdasarkan tampilan-tampilan utama dalam aplikasi:

1. Tampilan Splash Screen

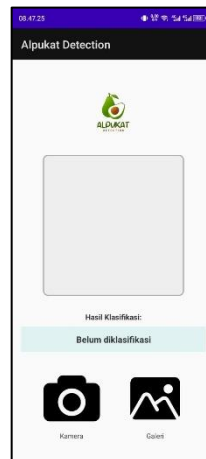
Pada Tampilan ini muncul pertama kali saat aplikasi dijalankan seperti ditunjukkan di gambar 7. Splash screen menampilkan logo aplikasi “Alpukat Detection” dengan latar belakang polos. Tujuan dari tampilan ini adalah sebagai identitas visual aplikasi sekaligus memberikan waktu bagi sistem untuk mempersiapkan proses awal. Dengan tampilan yang sederhana namun profesional, pengguna dapat langsung mengenali identitas aplikasi yang sedang dijalankan.



Gambar 7. Tampilan Splash Screen Awal

2. Tampilan Utama

Setelah proses splash screen selesai, pengguna akan diarahkan ke tampilan utama aplikasi seperti di gambar 8. Tampilan ini menjadi pusat kontrol bagi pengguna untuk memulai proses klasifikasi gambar alpukat. Pada bagian atas tampilan terdapat logo dan nama aplikasi sebagai pengenalan dan pengingat bahwa pengguna sedang berada di dalam aplikasi “Alpukat Detection”.

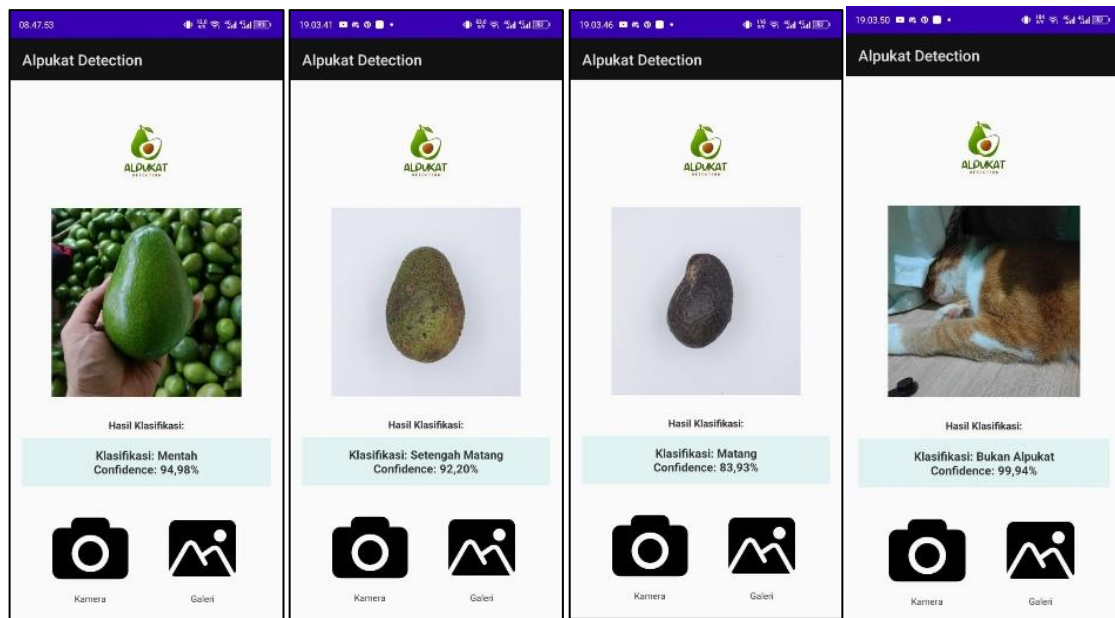


Gambar 8. Tampilan Utama Aplikasi

Di bagian tengah layar terdapat area kosong yang disiapkan untuk menampilkan gambar alpukat yang akan diklasifikasikan. Gambar ini bisa berasal dari kamera secara langsung ataupun dari galeri perangkat pengguna. Setelah gambar ditampilkan, proses klasifikasi akan dimulai secara otomatis dan hasilnya akan ditampilkan pada bagian bawah gambar. Tersedia dua tombol utama di bagian bawah, yaitu Camera digunakan untuk mengakses kamera perangkat dan mengambil gambar alpukat secara langsung. Gallery digunakan untuk memilih gambar yang sudah ada di penyimpanan perangkat.

3. Tampilan Hasil Input Gambar

Di bagian tengah layar terdapat area kosong yang disiapkan untuk menampilkan gambar alpukat yang akan diklasifikasikan. Gambar ini bisa berasal dari kamera secara langsung ataupun dari galeri perangkat pengguna. Setelah gambar ditampilkan, proses klasifikasi akan dimulai secara otomatis dan hasilnya akan ditampilkan pada bagian bawah gambar seperti ditunjukkan pada hasil pengujian di gambar 9. Tersedia dua tombol utama di bagian bawah, yaitu Camera digunakan untuk mengakses kamera perangkat dan mengambil gambar alpukat secara langsung. Gallery digunakan untuk memilih gambar yang sudah ada di penyimpanan perangkat.



Gambar 9. Tampilan Hasil Deteksi Gambar Alpukat dan Bukan Alpukat

Hasil klasifikasi ditampilkan dalam bentuk label seperti: “Mentah”, “Setengah Matang”, “Matang”, “Bukan Alpukat”. Setiap hasil juga disertai dengan nilai persentase probabilitas dari masing-masing kategori, sehingga pengguna dapat memahami tingkat keyakinan sistem terhadap hasil deteksi. Tombol Camera dan Gallery tetap tersedia di tampilan ini untuk memungkinkan pengguna mengganti gambar dan melakukan klasifikasi ulang dengan cepat.

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai pengembangan aplikasi Android untuk klasifikasi tingkat kematangan buah alpukat menggunakan metode Convolutional Neural Network (CNN) berbasis pengolahan citra, dapat disimpulkan bahwa penerapan teknologi CNN mampu mengklasifikasikan kematangan buah alpukat dengan tingkat akurasi yang cukup tinggi. Aplikasi ini mampu mengelompokkan citra buah alpukat ke dalam lima kategori yaitu Mentah, Setengah Matang, Matang, dan Bukan Alpukat. Penggunaan metode CNN, khususnya arsitektur MobileNetV2 yang telah dikonversi ke dalam format TensorFlow Lite (.tflite), sehingga memungkinkan proses inferensi dapat dijalankan langsung di perangkat Android secara efisien dan cepat, tanpa memerlukan koneksi internet.

Akurasi model yang tinggi menunjukkan bahwa citra kulit alpukat merupakan fitur visual yang cukup representatif untuk membedakan tingkat kematangannya. Selain itu, hasil pengujian juga memperlihatkan bahwa sistem dapat melakukan klasifikasi dengan konsistensi yang baik terhadap berbagai jenis gambar alpukat, baik yang diambil dari kamera maupun galeri. Aplikasi yang dikembangkan tidak hanya memberikan klasifikasi dalam bentuk label, tetapi juga dilengkapi dengan probabilitas atau tingkat kepercayaan dari model terhadap hasil prediksi tersebut, sehingga pengguna dapat lebih yakin terhadap hasil deteksi.

Dari sisi implementasi aplikasi, antarmuka yang dirancang cukup sederhana dan ramah pengguna, mulai dari splash screen, halaman utama, hingga tampilan hasil klasifikasi, seluruh elemen dikembangkan untuk memudahkan pengguna dalam mengakses fitur klasifikasi dengan cepat. Dengan adanya aplikasi ini, diharapkan dapat membantu petani, distributor, sistem penyortiran buah otomatis maupun konsumen dalam menentukan tingkat kematangan buah alpukat secara akurat dan efisien, serta meminimalisir kesalahan dalam proses seleksi buah.

6. References

- [1] M. Mukhofifah and E. Nurraharjo, “SISTEM DETEKSI KEMATANGAN BUAH ALPUKAT MENGGUNAKAN METODE PENGOLAHAN CITRA,” *Dinamika Informatika : Jurnal Ilmiah Teknologi Informasi*, vol. 11, no. 1, pp. 12–23, Mar. 2019, doi: 10.35315/INFORMATIKA.V11I1.8144.
- [2] J. Saputra, Y. Sa, V. Yoga Pudya Ardhana, and M. Afriansyah, “RESOLUSI : Rekayasa Teknik Informatika dan Informasi Klasifikasi Kematangan Buah Alpukat Mentega Menggunakan Metode K-Nearest Neighbor Berdasarkan Warna Kulit Buah,” *Media Online*, vol. 3, no. 5, pp. 214–221, 2023, [Online]. Available: <https://djournals.com/resolusi>

- [3] G. W. Wibowo and E. Mulyanto, "Classification of Avocado Ripeness Levels Using CNN Method," *sinkron*, vol. 8, no. 3, pp. 1828–1837, Jul. 2024, doi: 10.33395/sinkron.v8i3.13854.
- [4] Y. A. Pratama, "Membangun Sistem Identifikasi Kematangan Buah Alpukat menggunakan teknologi Pengolahan Citra Digital," *Kalijaga : Jurnal Penelitian Multidisiplin Mahasiswa*, vol. 1, no. 3, pp. 102–108, Jul. 2024, doi: 10.62523/kalijaga.v1i3.18.
- [5] R. Triyogi, R. Magdalena, and B. Hidayat, "Mendeteksi Kematangan Buah Kelapa Sawit Menggunakan Convolution Neural Network Deep Learning," *Jurnal Nasional SAINS dan TEKNIK*, vol. 1, pp. 22–27, 2023, doi: 10.25124/logic.v1i1.6732.
- [6] A. Irvanda, R. Sulthan, J. Adila, R. Ananda, and M. Ula, "PENDETEKSIAN KEMATANGAN MANGGA BERBASIS FITUR ANALISIS WARNA DENGAN METODE CNN," *Prosiding Seminar Nasional Teknologi dan Teknik Informatika (SENASTIKA)*, vol. 1, no. 1, Nov. 2024, Accessed: Feb. 20, 2025. [Online]. Available: <https://proceedings.unimal.ac.id/senastika/article/view/848>
- [7] U. Khaira *et al.*, "Rancang Bangun Aplikasi Deteksi Penyakit Tanaman Jagung Melalui Citra Daun Berbasis Android Menggunakan Algoritma Convolutional Neural Network," 2024.
- [8] E. L. Puji Ristanti, "ANALISIS DAN PERBANDINGAN ARSITEKTUR VGG16 DAN MOBILENETV2 UNTUK KLASIFIKASI DAN IDENTIFIKASI PENYAKIT DAUN PADA TANAMAN CABAI MENGGUNAKAN CNN," Jul. 2024.
- [9] E. Ariawan, "Perbandingan Performa Arsitektur Convolutional Neural Network Menggunakan Transfer Learning untuk Model Deteksi Kesehatan Daun Comparison of Convolutional Neural Network Architecture Performance Using Transfer Learning for Leaf Healthiness Detection Models," vol. 8, no. 1, pp. 1–12, Mar. 2025, doi: 10.30813/jbase.v8i1.8116.
- [1] M. Mukhofifah and E. Nurraharjo, "SISTEM DETEKSI KEMATANGAN BUAH ALPUKAT MENGGUNAKAN METODE PENGOLAHAN CITRA," *Dinamika Informatika : Jurnal Ilmiah Teknologi Informasi*, vol. 11, no. 1, pp. 12–23, Mar. 2019, doi: 10.35315/INFORMATIKA.V11I1.8144.
- [2] J. Saputra, Y. Sa, V. Yoga Pudya Ardhana, and M. Afriansyah, "RESOLUSI : Rekayasa Teknik Informatika dan Informasi Klasifikasi Kematangan Buah Alpukat Mentega Menggunakan Metode K-Nearest Neighbor Berdasarkan Warna Kulit Buah," *Media Online*, vol. 3, no. 5, pp. 214–221, 2023, [Online]. Available: <https://djournals.com/resolusi>
- [3] G. W. Wibowo and E. Mulyanto, "Classification of Avocado Ripeness Levels Using CNN Method," *sinkron*, vol. 8, no. 3, pp. 1828–1837, Jul. 2024, doi: 10.33395/sinkron.v8i3.13854.
- [4] Y. A. Pratama, "Membangun Sistem Identifikasi Kematangan Buah Alpukat menggunakan teknologi Pengolahan Citra Digital," *Kalijaga : Jurnal Penelitian Multidisiplin Mahasiswa*, vol. 1, no. 3, pp. 102–108, Jul. 2024, doi: 10.62523/kalijaga.v1i3.18.
- [5] R. Triyogi, R. Magdalena, and B. Hidayat, "Mendeteksi Kematangan Buah Kelapa Sawit Menggunakan Convolution Neural Network Deep Learning," *Jurnal Nasional SAINS dan TEKNIK*, vol. 1, pp. 22–27, 2023, doi: 10.25124/logic.v1i1.6732.
- [6] A. Irvanda, R. Sulthan, J. Adila, R. Ananda, and M. Ula, "PENDETEKSIAN KEMATANGAN MANGGA BERBASIS FITUR ANALISIS WARNA DENGAN METODE CNN," *Prosiding Seminar Nasional Teknologi dan Teknik Informatika (SENASTIKA)*, vol. 1, no. 1, Nov. 2024, Accessed: Feb. 20, 2025. [Online]. Available: <https://proceedings.unimal.ac.id/senastika/article/view/848>
- [7] U. Khaira *et al.*, "Rancang Bangun Aplikasi Deteksi Penyakit Tanaman Jagung Melalui Citra Daun Berbasis Android Menggunakan Algoritma Convolutional Neural Network," 2024.
- [8] E. L. Puji Ristanti, "ANALISIS DAN PERBANDINGAN ARSITEKTUR VGG16 DAN MOBILENETV2 UNTUK KLASIFIKASI DAN IDENTIFIKASI PENYAKIT DAUN PADA TANAMAN CABAI MENGGUNAKAN CNN," Jul. 2024.
- [9] E. Ariawan, "Perbandingan Performa Arsitektur Convolutional Neural Network Menggunakan Transfer Learning untuk Model Deteksi Kesehatan Daun Comparison of Convolutional Neural Network Architecture Performance Using Transfer Learning for Leaf Healthiness Detection Models," vol. 8, no. 1, pp. 1–12, Mar. 2025, doi: 10.30813/jbase.v8i1.8116.
- [10] A. S. Ardiansyah and A. Nugroho, "Klasifikasi Penyakit Daun Kopi Dengan Arsitektur MobileNetV2," *Jurnal Ilmu Komputer dan Bisnis*, vol. 14, no. 1, pp. 66–73, May 2023, doi: 10.47927/jikb.v14i1.622.
- [11] D. Anggara, N. Suarna, and Y. Arie Wijaya, "PERFORMANCE COMPARISON ANALYSIS OF OPTIMIZER ADAM, SGD, AND RMSPROP ON THE H5 MODEL." [Online]. Available: <https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset>
- [12] P. Kinanti, R. Gustriansyah, and Z. R. Mair, "PENGUNAAN CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK KLASIFIKASI JENIS REMPAH – REMPAH," 2024.

- [13] A. Carolina Wibowo, S. Ardi Lestari, S. Informasi, F. Ilmu Komputer, and U. Duta Bangsa Surakarta, "ANALISIS PENGGUNAAN MACHINE LEARNING DALAM KLASIFIKASI PENENTUAN PENYAKIT JANTUNG," vol. 9, no. 2, 2024.
- [14] A. R. Hanum *et al.*, "ANALISIS KINERJA ALGORITMA KLASIFIKASI TEKS BERT DALAM MENDETEKSI BERITA HOAKS," vol. 11, no. 3, pp. 537–546, 2024, doi: 10.25126/jtiik2024118093.
- [15] Dede Kurniadi, Rifky Muhammad Shidiq, and Asri Mulyani, "Perbandingan Penggunaan Optimizer dalam Klasifikasi Sel Darah Putih Menggunakan Convolutional Neural Network," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 14, no. 1, pp. 77–86, Mar. 2025, doi: 10.22146/jnteti.v14i1.17162.
- [16] A. Anindyo Abhinowo, R. Rizal Isnanto, and D. Eridani, "PEMILIHAN MODEL TERBAIK ALGORITMA CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI JENIS BENCANA ALAM The Best Model Selection Of Convolutional Neural Network Algorithm For Natural Disaster Classification," *Jurnal Teknik Komputer*, vol. 1, no. 4, pp. 199–208, 2023, doi: 10.14710/jtk.v1i4.37656.
- [17] S. N. Ria, M. Walid, and B. A. Umam, "Pengolahan Citra Digital Untuk Identifikasi Jenis Penyakit Kulit Menggunakan Metode Convolutional Neural Network (CNN)," *Energy - Jurnal Ilmiah Ilmu-Ilmu Teknik*, vol. 12, no. 2, pp. 9–16, Dec. 2022, doi: 10.51747/energy.v12i2.1118.
- [18] P. Xavier, P. Rodrigues, and C. L. M. Silva, "'Hass' Avocado Ripening Photographic Dataset," vol. 1, 2024, doi: 10.17632/3XD9N945V8.1.
- [19] E. Rahmawati and D. Riana, "Pengolahan Citra Kutu Kebul untuk Mendeteksi Jumlah Kutu Kebul pada Citra Daun," 2019. [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [20] A. Anhar and R. A. Putra, "Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN)," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 2, p. 466, Apr. 2023, doi: 10.26760/elkomika.v11i2.466.
- [21] M. Nauval Azmi, "ANALISIS PENGARUH DIMENSI GAMBAR PADA KLASIFIKASI MOTIF BATIK DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK," *Jurnal Sistem Informasi dan Informatika (Simika)*, vol. 6, no. 2, 2023.