

Klasifikasi Penyakit Daun Jagung Menggunakan MobileNetV2 dan TensorFlow Lite untuk Deployment Mobile di Distrik Semangga Merauke

Corn Leaf Disease Classification Using MobileNetV2 and TensorFlow Lite for Mobile Deployment in Semangga District Merauke

Jarot Budiasto^{*1}, Hasanudin Jayawardana¹, Tri Kustanti Rahayu¹, Nasra Pratama Putra¹

¹Sistem Informasi, Fakultas Teknik, Universitas Musamus, Merauke, Indonesia

Info Artikel

Diserahkan:
25 April 2026
Direvisi:
29 April 2026
Diterima:
30 April 2026
Diterbitkan:
30 April 2026

ABSTRAK

Studi deteksi penyakit daun jagung berbasis deep learning pada dataset PlantVillage umumnya melaporkan akurasi tinggi namun jarang mengevaluasi efisiensi model untuk deployment mobile secara komprehensif lintas format konversi. Penelitian ini mengimplementasikan transfer learning dengan arsitektur MobileNetV2 untuk klasifikasi empat kelas kondisi daun jagung menggunakan dataset PlantVillage (4.188 citra), dilanjutkan konversi ke TensorFlow Lite sebagai persiapan deployment pada aplikasi Android. Model dilatih dengan pendekatan dua-fase (feature extraction dan fine-tuning). Hasil evaluasi menunjukkan akurasi 91,69% dengan F1-score macro average 90,18%. Analisis confusion matrix mengungkap tiga temuan: (1) precision sempurna 100% pada kelas Healthy tanpa false negative, menjamin model tidak salah mengklasifikasi daun sakit sebagai sehat; (2) kelas Common Rust menjadi confuser utama akibat ketidakseimbangan data pelatihan; dan (3) pertukaran klasifikasi antara Northern Leaf Blight dan Gray Leaf Spot sesuai kemiripan gejala visual. Konversi TFLite terkuantisasi menghasilkan reduksi ukuran 89,93% (25,26 MB ke 2,55 MB) tanpa kehilangan akurasi yang signifikan. Studi ini berfungsi sebagai baseline metodologi sebelum pengumpulan dataset lokal di Distrik Semangga, Merauke.

Kata Kunci : Deep learning, MobileNetV2, transfer learning, penyakit daun jagung, TensorFlow Lite

ABSTRACT

Deep learning-based corn leaf disease detection studies on the PlantVillage dataset commonly report high accuracy but rarely evaluate comprehensive model efficiency for mobile deployment across conversion formats. This study implements transfer learning with MobileNetV2 architecture for classifying four corn leaf conditions using the PlantVillage dataset (4,188 images), followed by TensorFlow Lite conversion as preparation for Android application deployment. The model was trained using a two-phase approach (feature extraction and fine-tuning). Evaluation results show 91.69% accuracy with a macro F1-score of 90.18%. Confusion matrix analysis reveals three findings: (1) perfect 100% precision on the Healthy class with zero false negatives, guaranteeing the model does not misclassify diseased leaves as healthy; (2) Common Rust as the main confuser due to training data imbalance; and (3) classification confusion between Northern Leaf Blight and Gray Leaf Spot, consistent with their visual symptom similarity. Quantized TFLite conversion yields 89.93% size reduction (25.26 MB to 2.55 MB) without significant accuracy loss. This study serves as a methodological baseline before local dataset collection in Semangga District, Merauke.

Keywords : Deep learning, MobileNetV2, transfer learning, corn leaf disease, TensorFlow Lite

Corresponding author email: jarot@unmus.ac.id



This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

1. PENDAHULUAN

Jagung (*Zea mays* L.) merupakan komoditas pangan strategis nasional yang berperan penting dalam ketahanan pangan Indonesia, menjadi sumber karbohidrat utama setelah padi dan bahan baku industri pakan ternak, pangan olahan, serta bioenergi [1]. Di wilayah timur Indonesia, khususnya Kabupaten Merauke, Provinsi Papua Selatan, jagung berpotensi menjadi komoditas andalan baru dengan dukungan lahan luas di sentra pertanian

seperti Distrik Semangga. Namun, produktivitas masih menghadapi tantangan serius akibat tiga penyakit daun utama: hawar daun utara (*Northern Leaf Blight*), karat daun (*Common Rust*), dan bercak kelabu (*Gray Leaf Spot*), yang dapat menurunkan kualitas dan kuantitas hasil panen signifikan [2]. Pengamatan visual manual oleh petani bersifat subjektif, rentan kesalahan, dan tidak efektif untuk pemantauan lahan yang luas, terutama bagi petani skala kecil dengan akses teknologi terbatas [3].

Perkembangan *deep learning* berbasis *Convolutional Neural Network* (CNN) menawarkan solusi otomatis untuk identifikasi penyakit tanaman dengan akurasi tinggi [4], [5]. Berbagai studi telah mendemonstrasikan efektivitas CNN untuk klasifikasi penyakit daun jagung dengan akurasi tinggi: VGG16 mencapai 94,67% [6], ResNet34 dengan F1-score 0,93 [7], MobileNetV2 dengan fitur augmentasi 96,6% [8], hybrid CNN-Vision Transformer 98% [9], dan MobileNetV2 pada citra lapangan 97% [10]. Meskipun demikian, arsitektur CNN berskala besar seperti VGG16 (138 juta parameter) memiliki kebutuhan komputasi yang tinggi sehingga kurang sesuai untuk *deployment mobile*, sementara penelitian yang melaporkan *pipeline* lengkap dari pelatihan hingga evaluasi efisiensi konversi model untuk *deployment* Android masih relatif terbatas [11].

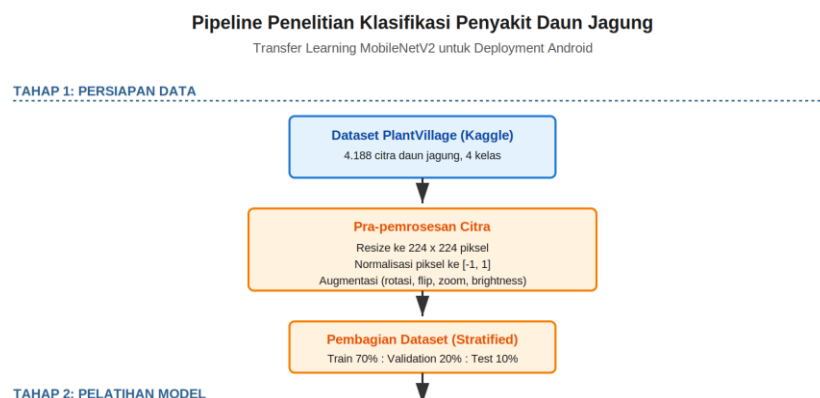
Arsitektur MobileNetV2 dirancang khusus untuk perangkat mobile dengan teknik *inverted residuals* dan *linear bottlenecks* yang memberikan keseimbangan antara akurasi dan efisiensi komputasi [12]. Kombinasi *transfer learning* dengan MobileNetV2 memungkinkan pengembangan model akurat dengan dataset terbatas sekaligus cukup ringan untuk dijalankan pada Android [13]. Beberapa studi terkini bahkan melakukan modifikasi MobileNetV2 untuk meningkatkan performa, seperti integrasi *Squeeze-and-Excitation module* [14] dan *fine-tuning* arsitektur dengan konfigurasi yang dioptimalkan [15]. Untuk *deployment* praktis, TensorFlow Lite (TFLite) menyediakan *framework on-device inference* dengan optimasi seperti *post-training quantization* untuk mereduksi ukuran model [16]. Aspek ini krusial untuk penerapan di wilayah 3T (tertinggal, terdepan, terluar) seperti Distrik Semangga dengan konektivitas internet terbatas. Pendekatan *deployment* TFLite untuk aplikasi Android pada klasifikasi penyakit tanaman telah dieksplorasi pada beberapa studi [17], namun pelaporan komparatif antar format konversi masih terbatas.

Dari kajian literatur, dapat diidentifikasi tiga celah riset yang ingin diisi penelitian ini. Pertama, sebagian besar studi terdahulu fokus pada pencapaian akurasi tanpa pelaporan komprehensif tentang efisiensi model hasil konversi lintas format (Keras, TFLite standar, TFLite terkuantisasi). Kedua, analisis mendalam pola kesalahan klasifikasi per kelas yang dapat memberikan wawasan praktis untuk pengembangan lanjutan masih terbatas. Ketiga, studi baseline yang jujur tentang performa model pada dataset publik sebagai fondasi pengembangan dataset lokal untuk wilayah spesifik Indonesia, khususnya Papua, belum tersedia.

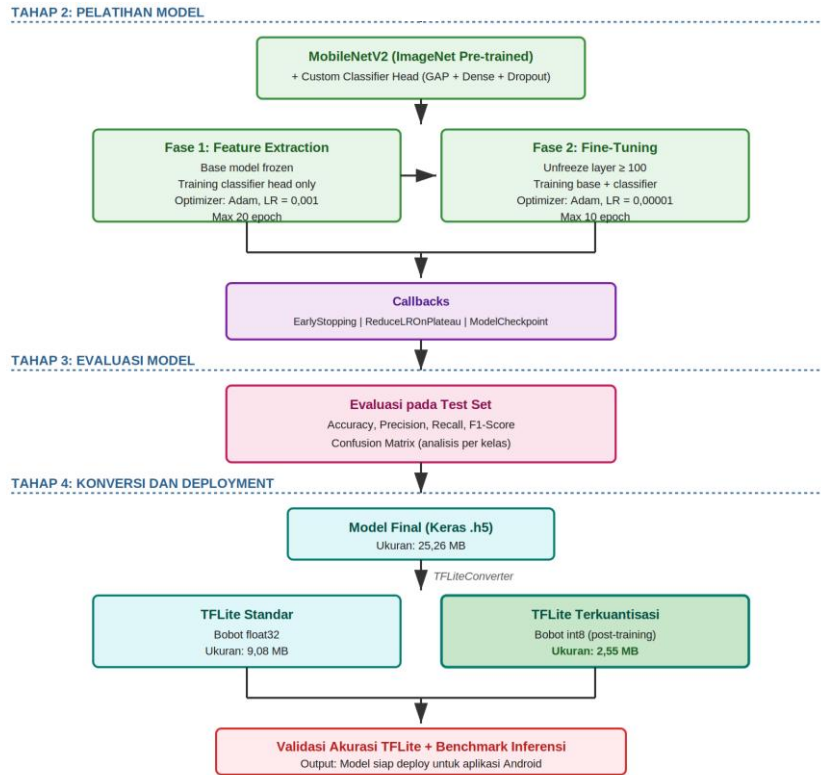
Penelitian ini mengimplementasikan *transfer learning* dengan MobileNetV2 untuk klasifikasi penyakit daun jagung yang dikonversi ke TensorFlow Lite sebagai persiapan *deployment* Android, menggunakan dataset PlantVillage subset Corn/Maize. Kontribusi penelitian mencakup tiga aspek. Pertama, implementasi lengkap *pipeline transfer learning* dua-fase pada MobileNetV2 dengan pelaporan *hyperparameter* lengkap untuk *reproducibility*. Kedua, evaluasi efisiensi model lintas format (Keras, TFLite standar, TFLite terkuantisasi) dengan fokus pada ukuran file dan konsistensi akurasi. Ketiga, analisis mendalam pola kesalahan klasifikasi per kelas yang menjadi landasan pengembangan dataset lokal Distrik Semangga pada penelitian tahap lanjutan. Penelitian ini secara eksplisit diposisikan sebagai studi *baseline* metodologis, bukan klaim *state-of-the-art* pada PlantVillage.

2. METODE PENELITIAN

Penelitian menggunakan pendekatan eksperimental dengan *transfer learning* untuk membangun model klasifikasi penyakit daun jagung. Seluruh proses pelatihan dan evaluasi dilakukan pada *platform* Google Colab dengan akselerasi GPU NVIDIA T4. Alur *pipeline* penelitian disajikan pada Gbr. 1.



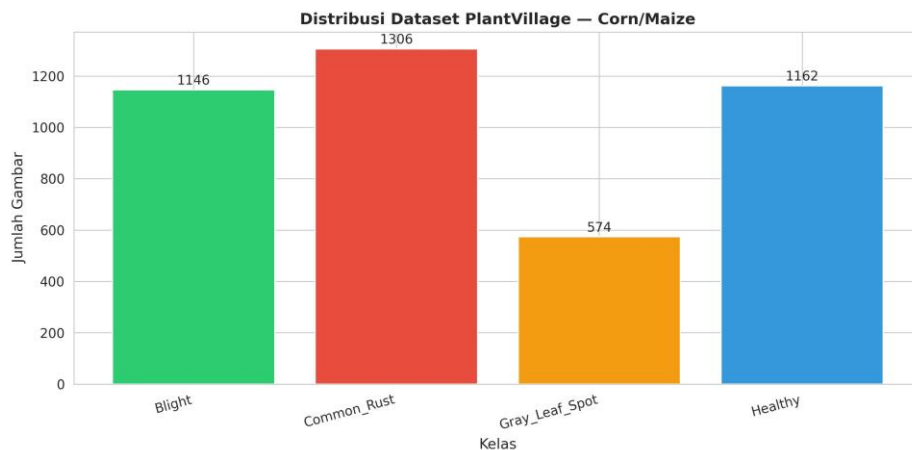
Gbr. 1. Pipeline penelitian klasifikasi penyakit daun jagung dengan transfer learning MobileNetV2



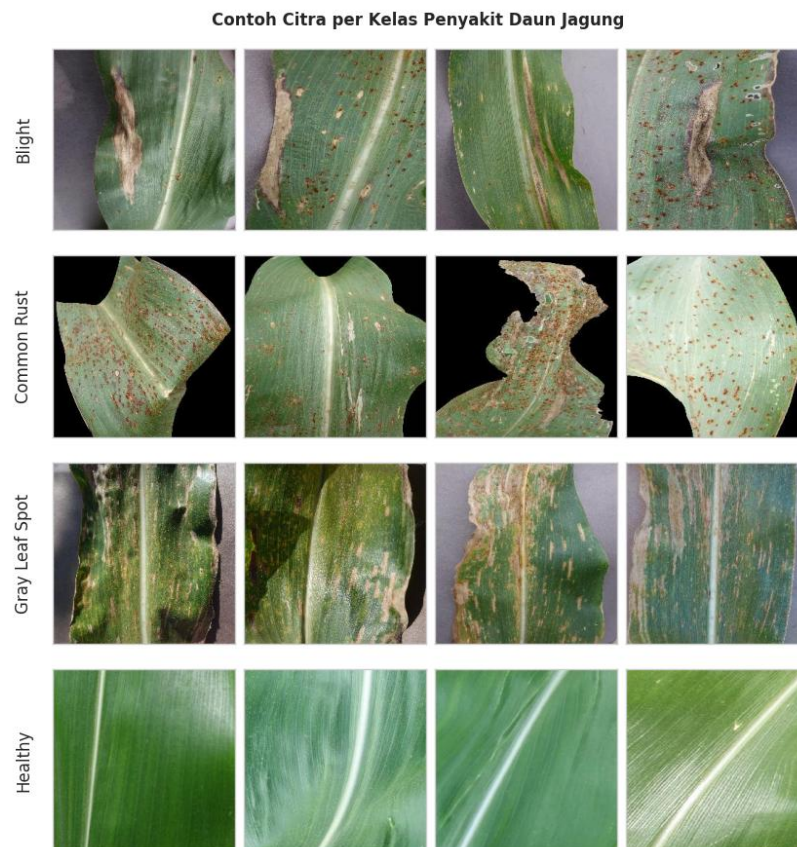
Gbr. 1. Lanjutan

2.1. Dataset

Dataset yang digunakan adalah subset citra daun jagung dari PlantVillage Dataset [18], sebuah dataset *benchmark* yang banyak digunakan pada studi klasifikasi penyakit tanaman berbasis *deep learning* [19], [20]. Dataset terdiri dari 4.188 citra dalam empat kelas: *Healthy*, *Common Rust*, *Gray Leaf Spot*, dan *Northern Leaf Blight*. Gbr. 2 menyajikan distribusi jumlah citra per kelas, di mana *Gray Leaf Spot* memiliki sampel paling sedikit (574 citra atau 13,71%). Contoh citra per kelas ditampilkan pada Gbr. 3, dengan urutan baris pertama *Northern Leaf Blight*, kedua *Common Rust*, ketiga *Gray Leaf Spot*, dan keempat *Healthy*.



Gbr. 2. Distribusi jumlah citra per kelas pada dataset PlantVillage (Corn/Maize)



Gbr. 3. Contoh citra per kelas (dari atas ke bawah): Northern Leaf Blight, Common Rust, Gray Leaf Spot, Healthy

Dataset dibagi menjadi tiga subset dengan proporsi 70% *training*, 20% *validation*, dan 10% *testing* secara stratifikasi per kelas. Pembagian menggunakan *random seed* tetap (42) untuk *reproducibility*. Rincian pembagian disajikan pada Tabel 1.

Tabel 1. Pembagian dataset pada setiap kelas

Kelas	Training	Validation	Testing	Total
Northern Leaf Blight	802	229	115	1.146
Common Rust	914	261	131	1.306
Gray Leaf Spot	401	115	58	574
Healthy	813	232	117	1.162
Total	2.930	837	421	4.188

2.2. Pra-pemrosesan dan Augmentasi Data

Seluruh citra melalui pra-pemrosesan: *resize* ke 224×224 piksel, normalisasi nilai piksel ke rentang [-1, 1] menggunakan fungsi *preprocess_input* MobileNetV2, dan augmentasi khusus data pelatihan. Teknik augmentasi meliputi rotasi acak $\pm 20^\circ$, pergeseran horizontal dan vertikal 15%, *flip* horizontal, *zoom* $\pm 15\%$, dan penyesuaian kecerahan [0,85, 1,15]. Augmentasi hanya pada data *training*.

2.3. Arsitektur Model

Arsitektur menggunakan MobileNetV2 [12] *pre-trained* pada ImageNet tanpa *top layer*, ditambah *classifier head* kustom: GlobalAveragePooling2D, *Dropout rate* 0,3, *Dense layer* 128 unit dengan aktivasi ReLU, *Dropout* kedua *rate* 0,3, dan *Dense output* 4 unit dengan aktivasi *softmax*. Pemilihan 128 unit pada *hidden Dense layer* didasarkan pada rekomendasi lapisan *bottleneck* kompak untuk *transfer learning* MobileNet. Nilai *dropout rate* 0,3 dipilih sebagai kompromi antara regularisasi dan kapasitas model, konsisten dengan praktik umum pada studi klasifikasi citra tanaman [8]. Total parameter model adalah 2.422.468 (*base* MobileNetV2 2.257.984 parameter dan *classifier head* 164.484 parameter).

2.4. Strategi Pelatihan dan Hyperparameter

Pelatihan dilakukan dalam dua fase. Fase pertama (*feature extraction*) membekukan seluruh lapisan *base model* dan hanya melatih *classifier head* dengan *optimizer* Adam ($\beta_1=0,9$, $\beta_2=0,999$, $\epsilon=10^{-7}$), *learning rate* 0,001, selama maksimal 20 *epoch*. Fase kedua (*fine-tuning*) melepas *freeze* pada lapisan ke-100 ke atas dari *base model*

dan melatihnya bersama *classifier head* dengan *learning rate* 1×10^{-5} selama maksimal 10 *epoch*. Penurunan *learning rate* signifikan pada fase kedua menghindari degradasi bobot *pre-trained*.

Tiga *callback* diterapkan: *EarlyStopping* dengan *patience* 5 *epoch* berdasarkan *validation loss*, *ReduceLRonPlateau* dengan faktor 0,5 pada *patience* 3 *epoch*, dan *ModelCheckpoint* yang menyimpan bobot terbaik berdasarkan *validation accuracy*. *Loss function* menggunakan *categorical crossentropy* dengan *batch size* 32. *Random seed* ditetapkan konstan (SEED=42) pada NumPy dan TensorFlow untuk *reproducibility*. Lingkungan pelatihan: Python 3.10, TensorFlow 2.15, GPU NVIDIA Tesla T4 16GB.

2.5. Metrik Evaluasi

Evaluasi menggunakan metrik standar untuk klasifikasi multi-kelas: *accuracy*, *precision*, *recall*, dan *F1-score*. *Accuracy* mengukur proporsi prediksi benar terhadap seluruh sampel (1). *Precision* mengukur proporsi prediksi positif yang benar-benar positif (2), sedangkan *recall* mengukur proporsi sampel positif yang berhasil terdeteksi (3). *F1-score* merupakan *harmonic mean* dari *precision* dan *recall* (4), memberikan ukuran seimbang terutama pada kondisi data tidak seimbang.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{F1 - Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

TP (*True Positive*) adalah jumlah sampel positif yang diprediksi positif; TN (*True Negative*) adalah jumlah sampel negatif yang diprediksi negatif; FP (*False Positive*) adalah jumlah sampel negatif yang salah diprediksi positif; dan FN (*False Negative*) adalah jumlah sampel positif yang salah diprediksi negatif. Ketiga metrik *precision*, *recall*, dan *F1-score* dihitung per kelas, kemudian diagregasi menjadi nilai keseluruhan melalui dua pendekatan: *macro average* dan *weighted average*. *Macro average* (5) adalah rata-rata aritmetik sederhana dari nilai metrik setiap kelas, memberikan bobot yang sama untuk setiap kelas terlepas dari jumlah sampelnya. Sebaliknya, *weighted average* (6) memberikan bobot proporsional berdasarkan jumlah sampel (*support*) pada setiap kelas, sehingga kelas dengan sampel lebih banyak memberikan kontribusi yang lebih besar terhadap nilai agregat.

$$\text{Macro Avg} = \frac{1}{N} \sum \text{Metric}_i \quad (5)$$

$$\text{Weighted Avg} = \frac{\sum (\text{Support}_i \times \text{Metric}_i)}{\sum \text{Support}_i} \quad (6)$$

Pada persamaan (5) dan (6), N adalah jumlah kelas (dalam penelitian ini $N=4$), indeks i merujuk pada setiap kelas ($i = 1, 2, 3, 4$ untuk *Northern Leaf Blight*, *Common Rust*, *Gray Leaf Spot*, dan *Healthy*), Metric_i adalah nilai *precision*, *recall*, atau *F1-score* kelas ke- i , dan Support_i adalah jumlah sampel aktual kelas ke- i pada data uji. Dalam konteks dataset yang tidak seimbang seperti pada penelitian ini, perbedaan antara *macro average* dan *weighted average* memberikan informasi penting: *macro average* menyoroti performa pada kelas minoritas, sedangkan *weighted average* mencerminkan performa keseluruhan yang didominasi kelas mayoritas. Selain metrik di atas, *confusion matrix* digunakan untuk menganalisis pola kesalahan klasifikasi antar kelas secara visual.

2.6. Konversi Model ke TensorFlow Lite

Model Keras dikonversi ke TensorFlow Lite dalam dua varian menggunakan TFLiteConverter: TFLite standar mempertahankan bobot float32, sedangkan TFLite terkuantisasi mengubah bobot ke int8 melalui *post-training quantization default* (tf.lite.Optimize.DEFAULT) [16]. Kedua varian dievaluasi dari segi ukuran file dan konsistensi akurasi klasifikasi pada data uji menggunakan *interpreter* TFLite.

3. HASIL DAN PEMBAHASAN

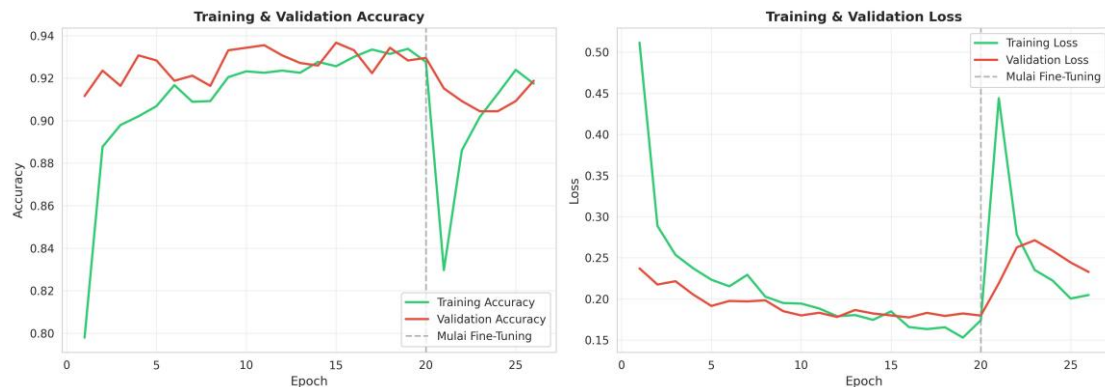
3.1. Hasil Pelatihan Model

Pelatihan berlangsung dalam dua fase dengan total waktu 22,3 menit pada GPU NVIDIA T4 (fase pertama 16,7 menit, fase kedua 5,5 menit). Pada fase pertama, *validation accuracy* puncak 93,67% tercapai di *epoch* ke-15, dengan *validation loss* terbaik 0,1775 di *epoch* ke-16 yang di-*restore* oleh mekanisme *restore_best_weights*. Fase kedua berjalan 6 *epoch* sebelum dihentikan *EarlyStopping*, mencapai *validation accuracy* tertinggi 91,88% pada *epoch* ke-26.

Pada transisi ke fase *fine-tuning*, *training accuracy* sempat menurun tajam dari 92,8% menjadi 83,0% pada *epoch* pertama *fine-tuning* (*epoch* ke-21). Fenomena ini terjadi karena saat sebagian lapisan *base model* di-*unfreeze*, gradien mulai berpropagasi melalui lapisan yang sebelumnya beku, menggeser nilai bobot yang sebelumnya teroptimasi pada tugas ImageNet. Meskipun *learning rate* sudah kecil (1×10^{-5}), pergeseran bobot

pada sejumlah besar parameter (~75% layer MobileNetV2) berinteraksi dengan augmentasi data yang diterapkan, menyebabkan *loss* sementara meningkat. *Training accuracy* cepat pulih dalam beberapa *epoch* berikutnya seiring model beradaptasi dengan konfigurasi baru. Pola ini konsisten dengan dinamika *fine-tuning* yang umum dilaporkan pada literatur *transfer learning* [13].

Fine-tuning tidak memberikan peningkatan signifikan pada *validation accuracy* puncak, mengindikasikan bahwa representasi fitur dari *feature extraction* sudah cukup optimal untuk domain citra daun jagung pada dataset PlantVillage. Gbr. 4 menyajikan kurva *training* dan *validation accuracy* serta *loss* untuk kedua fase, menunjukkan konvergensi stabil tanpa tanda *overfitting* signifikan.



Gbr. 4. Kurva training dan *validation accuracy* (kiri) dan *loss* (kanan) untuk kedua fase pelatihan

3.2. Evaluasi pada Data Uji

Evaluasi pada *test set* menunjukkan *test accuracy* sebesar 91,69% dengan *test loss* 0,2075. Konsistensi antara *validation accuracy* (91,88%) dan *test accuracy* (91,69%) dengan selisih yang sangat kecil (0,19%) mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik tanpa indikasi *overfitting* maupun *data leakage*. Ringkasan metrik evaluasi disajikan pada Tabel 2.

Tabel 2. Ringkasan metrik evaluasi model pada data uji

Metrik	Nilai
Accuracy	91,69%
Precision (macro avg)	90,70%
Recall (macro avg)	90,07%
F1-Score (macro avg)	90,18%
Precision (weighted avg)	91,95%
Recall (weighted avg)	91,69%
F1-Score (weighted avg)	91,57%

Analisis lebih lanjut pada tingkat kelas (Tabel 3) menunjukkan variasi performa model. Kelas *Healthy* memiliki kinerja tertinggi (F1-score 99,57%), diikuti *Common Rust* (93,19%). Kelas *Northern Leaf Blight* menunjukkan *precision* tinggi (93,94%) namun *recall* lebih rendah (80,87%), mengindikasikan masih adanya data yang tidak terdeteksi (*false negative*). Sementara itu, *Gray Leaf Spot* memiliki performa terendah dan seimbang (F1-score 81,03%), yang menunjukkan tantangan dalam pemisahan fitur antar kelas.

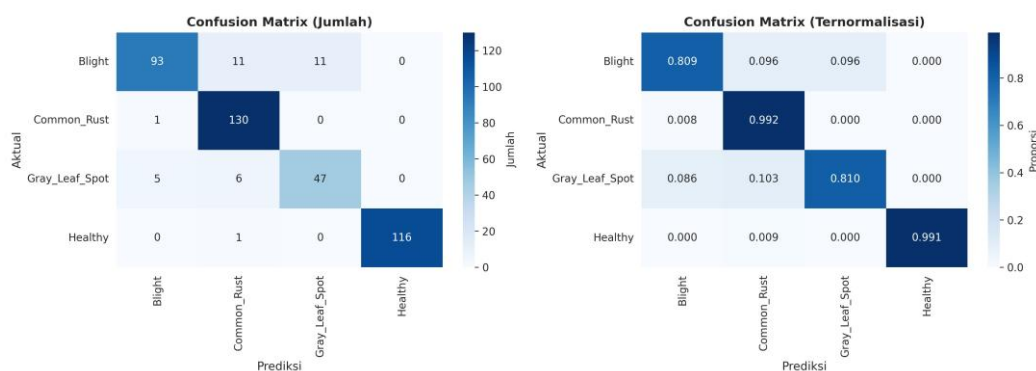
Tabel 3. Metrik evaluasi per kelas pada data uji

Kelas	Precision	Recall	F1-Score	Support
Northern Leaf Blight	93,94%	80,87%	86,92%	115
Common Rust	87,84%	99,24%	93,19%	131
Gray Leaf Spot	81,03%	81,03%	81,03%	58
Healthy	100,00%	99,15%	99,57%	117

Secara keseluruhan, meskipun performa global model tinggi, terdapat perbedaan kemampuan klasifikasi antar kelas yang dipengaruhi oleh kompleksitas pola dan distribusi data.

3.3. Analisis Confusion Matrix

Gbr. 5 menyajikan *confusion matrix* yang mengungkap tiga temuan penting terhadap karakteristik model.



Gbr. 5. Confusion matrix klasifikasi penyakit daun jagung pada data uji (kiri: jumlah, kanan: ternormalisasi)

3.3.1. Temuan 1: Pembedaan Daun Sakit dan Sehat Hampir Sempurna

Dari 304 sampel daun sakit pada data uji (gabungan *Northern Leaf Blight*, *Common Rust*, dan *Gray Leaf Spot*), tidak ada satu pun (0 dari 304) yang salah diklasifikasikan sebagai *Healthy*. Sebaliknya, dari 117 sampel daun sehat, hanya satu (0,85%) salah diprediksi sebagai *Common Rust*. Hasilnya adalah precision sempurna 100% dan *F1-score* 99,57% pada kelas *Healthy*. Implikasi praktisnya, dalam konteks pemanfaatan oleh petani melalui sistem *mobile-based* untuk deteksi penyakit tanaman [19], kesalahan mengklasifikasikan daun sakit sebagai sehat (*false negative* pada *Healthy*) merupakan jenis kesalahan paling merugikan karena menyebabkan keterlambatan intervensi. Model tidak pernah melakukan kesalahan jenis ini pada data uji.

3.3.2. Temuan 2: Common Rust sebagai Confuser Utama

Meskipun kelas *Common Rust* mencatat *recall* sebesar 99,24% (130 dari 131 sampel terklasifikasi dengan benar), kelas ini juga menerima jumlah kesalahan klasifikasi terbanyak. Sebanyak 11 sampel *Northern Leaf Blight*, 6 sampel *Gray Leaf Spot*, dan 1 sampel *Healthy* salah diprediksi sebagai *Common Rust*, sehingga *precision* menurun menjadi 87,84%. Pola ini mengindikasikan adanya kecenderungan model untuk memprediksi *Common Rust* pada kasus yang ambigu. Salah satu penyebab utama adalah ketidakseimbangan data pelatihan, di mana *Common Rust* memiliki jumlah sampel terbanyak (914 citra), sehingga model memperoleh eksposur yang lebih besar terhadap variasi visual kelas tersebut. Fenomena *class imbalance* pada dataset PlantVillage juga dilaporkan pada studi hybrid CNN-ViT [9] sebagai faktor yang memengaruhi bias klasifikasi.

3.3.3. Temuan 3: Pertukaran Blight dan Gray Leaf Spot

Pertukaran klasifikasi antara *Northern Leaf Blight* dan *Gray Leaf Spot* yang dilaporkan dalam literatur patologi tanaman [20] juga teramati: 11 sampel *Blight* diprediksi sebagai *Gray Leaf Spot* dan 5 sampel *Gray Leaf Spot* diprediksi sebagai *Blight*. Kedua penyakit memiliki gejala visual serupa berupa bercak memanjang berwarna kecoklatan hingga keabu-abuan, sehingga kemiripan ini menjadi tantangan umum dalam studi *transfer learning* untuk penyakit jagung [21]. Kelas *Gray Leaf Spot* menjadi kelas dengan performa terendah (*F1-score* 81,03%) karena jumlah sampel pelatihan terbatas (401 citra) dan kemiripan visual dengan *Blight*. Temuan ini memberikan arah jelas untuk pengembangan penelitian lanjutan di Distrik Semangga: pengumpulan data lokal yang seimbang antar kelas, terutama untuk *Gray Leaf Spot*.

3.4. Efisiensi Konversi Model

Model Keras terlatih dikonversi ke TensorFlow Lite dalam dua varian. Perbandingan ukuran ketiga format disajikan pada Tabel 4.

Tabel 4. Perbandingan ukuran model pada berbagai format

Format Model	Ukuran (MB)	Reduksi (%)
Keras (.h5)	25,26	0,00%
TFLite Standar	9,08	64,07%
TFLite Terkuantisasi	2,55	89,93%

Reduksi ukuran sebesar 89,93% (dari 25,26 MB ke 2,55 MB) memungkinkan *deployment* ringan pada aplikasi Android dengan beban penyimpanan minimal. Ukuran 2,55 MB ini berimplikasi praktis penting untuk wilayah dengan infrastruktur terbatas seperti Distrik Semangga, memungkinkan aplikasi berjalan sepenuhnya offline tanpa ketergantungan pada server eksternal.

3.4.1. Validasi Akurasi Model TFLite

Untuk memastikan konversi TFLite tidak merusak performa klasifikasi, model dievaluasi ulang pada data uji dengan *interpreter* TFLite. Hasilnya: akurasi TFLite standar 92,64% dan TFLite terkuantisasi 93,11%, dibanding Keras asli 91,69%. Menariknya, kedua model TFLite mencatat akurasi sedikit lebih tinggi dari Keras asli, yakni dengan selisih 0,95%–1,43% atau setara 4–6 sampel dari total 421 sampel uji.

Fenomena ini memerlukan penjelasan yang cermat. Meskipun *quantization* secara teori umumnya menurunkan akurasi [16], selisih kecil dalam dua arah (naik atau turun) dapat terjadi karena tiga faktor. Pertama, perbedaan *engine inference: interpreter* Keras dan TFLite menggunakan implementasi operasi aritmetika yang berbeda, terutama pada tahap akumulasi bobot dan aktivasi; perbedaan urutan operasi *floating-point* dapat menghasilkan akumulasi error numerik yang sedikit berbeda, yang dapat mengubah prediksi pada sampel dengan probabilitas mendekati ambang batas kelas. Kedua, prosedur *inference: evaluasi* Keras menggunakan *batch-wise inference* (32 sampel sekaligus) sedangkan evaluasi TFLite pada penelitian ini menggunakan *sample-wise inference* (satu per satu), yang dapat memengaruhi operasi normalisasi *batch*. Ketiga, pada model dengan akurasi 91-93%, selisih 4-6 sampel berada dalam margin variasi yang kecil dan dapat dianggap bukan perbedaan signifikan secara statistik. Dalam konteks penelitian ini, temuan utama yang relevan adalah bahwa *quantization* tidak menurunkan akurasi secara signifikan, sehingga model TFLite terkuantisasi dapat direkomendasikan sebagai kandidat *deployment* dari perspektif ukuran.

3.4.2. Benchmark Waktu Inferensi

Pengukuran waktu inferensi dilakukan pada CPU Google Colab (Intel Xeon, arsitektur x86-64) dengan hasil: TFLite standar $9,67 \pm 5,03$ ms dan TFLite terkuantisasi $25,61 \pm 1,84$ ms. Hasil ini tidak dapat diekstrapolasi langsung ke perangkat Android yang menggunakan prosesor ARM karena perbedaan arsitektur fundamental. x86-64 memiliki optimasi *instruction set AVX2* untuk operasi float32, sementara ARM memiliki NEON untuk operasi int8 [16]. Oleh karena itu, hasil *benchmark* di atas tidak dapat digunakan untuk menyimpulkan performa inferensi pada perangkat Android target. *Benchmark* representatif pada perangkat *smartphone* nyata menjadi bagian dari rencana penelitian lanjutan.

3.5. Pembahasan dan Implikasi

Akurasi 91,69% pada penelitian ini lebih rendah dibanding beberapa studi serupa pada dataset PlantVillage yang melaporkan akurasi 94–99% [6], [8], [9]. Perlu dipahami bahwa penelitian ini menggunakan dataset dan *setup* eksperimen yang serupa dengan studi-studi tersebut, sehingga perbedaan akurasi tidak dapat sepenuhnya dijustifikasi dengan argumen kondisi pengujian. Kemungkinan penyebab akurasi relatif lebih rendah mencakup: (1) struktur *classifier head* yang lebih sederhana dibanding modifikasi pada beberapa studi, dimana [8] menambahkan fitur augmentasi dan [9] menggunakan *hybrid architecture*; (2) tidak dilakukan *hyperparameter tuning* ekstensif; dan (3) penggunaan konfigurasi MobileNetV2 standar tanpa modifikasi arsitektur seperti yang dilakukan [14], [22]. Dengan demikian, akurasi 91,69% merupakan *baseline* yang jujur dari *pipeline transfer learning* dua-fase konvensional pada MobileNetV2, bukan *state-of-the-art* pada PlantVillage. Arsitektur alternatif seperti EfficientNet [23] atau pendekatan komparatif lintas arsitektur [21] dapat menjadi bahan perbandingan untuk penelitian lanjutan.

Meskipun akurasinya relatif lebih rendah, penelitian ini memberikan kontribusi yang tidak dibahas eksplisit pada studi-studi terdahulu: pelaporan komprehensif efisiensi konversi model lintas format dengan validasi akurasi pada setiap format. Kebanyakan studi hanya melaporkan akurasi pada model asli; perbandingan akurasi antara Keras, TFLite standar, dan TFLite terkuantisasi pada set uji yang sama memberikan informasi praktis untuk praktisi yang merancang *deployment mobile*. Sebagai contoh, studi-studi terkini pada deteksi penyakit jagung berbasis CNN seperti [6], [9], [24] berfokus pada pencapaian akurasi tanpa analisis komparatif antar format hasil konversi. Kesenjangan inilah yang berupaya diisi oleh penelitian ini.

Keterbatasan utama dataset PlantVillage adalah kondisi pengambilan citra yang terkontrol, seperti latar belakang yang seragam dan pencahayaan yang optimal, sehingga berbeda dari kondisi lapangan nyata. Performa model pada kondisi lapangan berpotensi lebih rendah. Fenomena ini dikonfirmasi oleh studi terdahulu [10] yang menguji MobileNetV2 pada citra lapangan dari empat negara di Afrika. Oleh karena itu, penelitian ini diposisikan sebagai studi *baseline* metodologis sebelum pengumpulan dataset lokal dari Distrik Semangga, yang bertujuan untuk memvalidasi model pada kondisi nyata.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan *transfer learning* dengan arsitektur MobileNetV2 untuk klasifikasi empat kelas kondisi daun jagung menggunakan dataset PlantVillage, sebagai studi *baseline* menuju *deployment* Android. Model mencapai akurasi 91,69% dengan F1-score macro 90,18% pada data uji. Analisis per kelas menunjukkan kelas *Healthy* mencapai *precision* sempurna 100% tanpa *false negative*, sementara *Gray Leaf Spot* menjadi kelas dengan performa terendah akibat ketidakseimbangan sampel dan kemiripan gejala visual. Konversi ke TensorFlow Lite terkuantisasi menghasilkan reduksi ukuran 89,93% (2,55 MB) tanpa menurunkan akurasi klasifikasi.

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan. Pertama, tidak dilakukan perbandingan internal dengan arsitektur alternatif seperti MobileNetV3 dan EfficientNet-Lite, sehingga klaim relatif terkait pemilihan arsitektur belum tervalidasi secara empiris dalam *setup* eksperimen yang sama. Kedua, akurasi yang dicapai lebih rendah dibandingkan beberapa studi terdahulu pada dataset serupa, yang kemungkinan disebabkan oleh tidak dilakukannya *hyperparameter tuning* secara ekstensif maupun modifikasi arsitektur. Ketiga, *benchmark* waktu inferensi dilakukan pada CPU dengan arsitektur x86-64, sehingga tidak merepresentasikan

performa pada prosesor ARM yang umum digunakan pada perangkat Android. Keempat, aplikasi Android terintegrasi belum dikembangkan pada tahap penelitian ini karena fokus utama masih pada persiapan dan evaluasi model.

Arah pengembangan untuk penelitian lanjutan meliputi: (1) pengumpulan dataset lokal dari lahan pertanian Distrik Semangga, Kabupaten Merauke, dengan perhatian khusus pada keseimbangan antar kelas, terutama *Gray Leaf Spot*; (2) perbandingan MobileNetV2 dengan arsitektur *mobile-friendly* alternatif dalam *setup* eksperimen yang sama untuk justifikasi pemilihan arsitektur; (3) *hyperparameter tuning* sistematis dan eksplorasi modifikasi *classifier head*; (4) pengembangan aplikasi Android terintegrasi dengan antarmuka ramah petani, dilanjutkan *benchmark* waktu inferensi pada perangkat smartphone nyata lintas spesifikasi; (5) uji penerimaan (*user acceptance testing*) bersama petani di Distrik Semangga sebagai validasi akhir sistem.

UCAPAN TERIMA KASIH

Penelitian ini didanai oleh DIPA Universitas Musamus Tahun Anggaran 2025 melalui skema Penelitian Dosen Pemula. Penulis mengucapkan terima kasih kepada Fakultas Teknik dan Jurusan Sistem Informasi Universitas Musamus atas dukungan fasilitas dan infrastruktur penelitian.

DAFTAR PUSTAKA

- [1] E. I. R. Rhofta, "Optimalisasi Sumber Daya Pertanian Indonesia untuk Mendukung Program Ketahanan Pangan dan Energi Nasional," *Jurnal Ketahanan Nasional*, p., 2022, doi: 10.22146/jkn.71642.
- [2] F. Khan, N. Zafar, M. N. Tahir, M. Aqib, H. Waheed, and Z. Haroon, "A mobile-based system for maize plant leaf disease detection and classification using deep learning," *Front. Plant Sci.*, vol. Volume 14-2023, 2023, [Online]. Available: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1079366>
- [3] I. Pacal, "Enhancing crop productivity and sustainability through disease identification in maize leaves: Exploiting a large dataset with an advanced vision transformer model," *Expert Syst. Appl.*, vol. 238, p. 122099, 2024, doi: <https://doi.org/10.1016/j.eswa.2023.122099>.
- [4] M. Shoaib *et al.*, "An advanced deep learning models-based plant disease detection: A review of recent research," *Front. Plant Sci.*, vol. 14, p. 1158933, 2023.
- [5] I. Pacal *et al.*, "A systematic review of deep learning techniques for plant diseases," *Artif. Intell. Rev.*, vol. 57, no. 11, p. 304, 2024.
- [6] M. Tariq *et al.*, "Corn leaf disease: insightful diagnosis using VGG16 empowered by explainable AI," *Front. Plant Sci.*, vol. Volume 15-2024, 2024, [Online]. Available: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2024.1402835>
- [7] G. Sharma, V. Anand, R. Chauhan, N. Garg, and S. Gupta, "Leaf Watch: An Innovative Multi-Class Pre-Trained Model for Untying the Spectrum of Maize Leaf Diseases," in *2023 3rd International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2023*, 2023. doi: 10.1109/SMARTGENCON60755.2023.10442289.
- [8] M. Alkanan and Y. Gulzar, "Enhanced corn seed disease classification: leveraging MobileNetV2 with feature augmentation and transfer learning," *Front. Appl. Math. Stat.*, vol. Volume 9-2023, 2024, [Online]. Available: <https://www.frontiersin.org/journals/applied-mathematics-and-statistics/articles/10.3389/fams.2023.1320177>
- [9] S. Aboelenin, F. A. Elbasheer, M. M. Eltouky, W. M. El-Hady, and K. M. Hosny, "A hybrid Framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer," *Complex and Intelligent Systems*, vol. 11, no. 2, 2025, doi: 10.1007/s40747-024-01764-x.
- [10] J. Nakatumba-Nabende and S. Murindanyi, "Deep learning models for enhanced in-field maize leaf disease diagnosis," *Machine Learning with Applications*, vol. 20, p. 100673, 2025, doi: <https://doi.org/10.1016/j.mlwa.2025.100673>.
- [11] J. Liu, C. He, Y. Jiang, M. Wang, Z. Ye, and M. He, "A high-precision identification method for maize leaf diseases and pests based on LFMNet under complex backgrounds," *Plants*, vol. 13, no. 13, p. 1827, 2024.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
- [13] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds., Cham: Springer International Publishing, 2018, pp. 270–279.
- [14] A. Y. Ashurov *et al.*, "Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections," *Front. Plant Sci.*, vol. 15, p. 1505857, 2025.
- [15] R. Sharma, M. Mittal, V. Gupta, and D. Vasdev, "Detection of plant leaf disease using advanced deep learning architectures," *International Journal of Information Technology*, vol. 16, no. 6, pp. 3475–3492, 2024.
- [16] B. Jacob *et al.*, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2704–2713, 2017, [Online]. Available: <https://api.semanticscholar.org/CorpusID:39867659>
- [17] D. Tejaswi, T. Sri Vaishnavi, B. Nandini, P. Nuka Raju, and D. Jayanth Babu, "Plant disease detection using deep learning," *International Journal of Science and Research Archive*, vol. 12, no. 1, pp. 2476–2488, Jun. 2024, doi: 10.30574/ijrsra.2024.12.1.1043.
- [18] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Front. Plant Sci.*, vol. Volume 7-2016, 2016, [Online]. Available: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2016.01419>
- [19] A. A. Ahmed and G. H. Reddy, "A mobile-based system for detecting plant leaf diseases using deep learning," *AgriEngineering*, vol. 3, no. 3, pp. 478–493, 2021.
- [20] J. M. J. Ward, E. L. Stromberg, D. C. Nowell, and F. W. Nutter Jr, "Gray leaf spot: a disease of global importance in maize production," *Plant Dis.*, vol. 83, no. 10, pp. 884–895, 1999.
- [21] A. S. Hatem, M. S. Altememe, and M. A. Fadhel, "Identifying corn leaves diseases by extensive use of transfer learning: a comparative study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 2, pp. 1030–1038, 2023.
- [22] M. Masood *et al.*, "MaizeNet: A deep learning approach for effective recognition of maize plant leaf diseases," *IEEE Access*, vol. 11, pp. 52862–52876, 2023.
- [23] F. Rajeena PP, A. SU, M. A. Moustafa, and M. A. S. Ali, "Detecting plant disease in corn leaf using efficientnet architecture—an analytical approach," *Electronics (Basel)*, vol. 12, no. 8, p. 1938, 2023.
- [24] I. Pacal and G. Işık, "Utilizing convolutional neural networks and vision transformers for precise corn leaf disease identification," *Neural Comput. Appl.*, vol. 37, no. 4, pp. 2479–2496, 2025.