**Analisis Perbandingan Kinerja Model Klasifikasi Jenis Jagung: Evaluasi Nilai Best vs. Last pada Metrik Akurasi dan Loss**

**Nicholas\*1), Zalsabilah Rezky Amelia Arep2), Sofia3), Devika Widya Vania4), Aeesha Pink Al-Islami Mahardhika5)**

1)Data Science, Telkom University Surabaya, Surabaya, Indonesia

2)Data Science, Telkom University Surabaya, Surabaya, Indonesia

3)Data Science, Telkom University Surabaya, Surabaya, Indonesia

4)Data Science, Telkom University Surabaya, Surabaya, Indonesia

5)Data Science, Telkom University Surabaya, Surabaya, Indonesia

[1\*](mailto:1*first@institution.ac.id)[nicholasgunawan5678@gmail.com](mailto:nicholasgunawan5678@gmail.com), 2[zalsabilamelia@gmail.com](mailto:zalsabilamelia@gmail.com), 3[sofiakuddah@gmail.com](mailto:sofiakuddah@gmail.com), 4[devikawidyavania@gmail.com](mailto:devikawidyavania@gmail.com), 5[aeeshamahardhika756@gmail.com](https://d.docs.live.net/B7907A2F58951DB1/aeeshamahardhika756@gmail.com)

|  |  |  |
| --- | --- | --- |
| **A R T I C L E I N F O** |  | **A B S T R A C T** |
| ***Keywords:***  Klasifikasi Varietas Jagung  MobileNetV3-Small  Augmentasi Data  Evaluasi Checkpoint  Generalisasi Model | Pengelompokan varietas biji jagung sangat penting untuk menjamin kualitas benih. Penelitian ini bertujuan mengembangkan sistem klasifikasi otomatis untuk tiga varietas jagung, yaitu Zea mays Chulpi Cancha, Indurata, dan Rugosa, dengan memanfaatkan arsitektur deep learning MobileNetV3-Small. Diterapkannya teknik augmentasi data seperti RandomResizedCrop dan RandomHorizontalFlip untuk mencegah terjadinya overfitting. Sistem pelatihan menggunakan pemilihan checkpoint akurasi validasi tertinggi dan loss terendah. Perbandingan kinerja antara model yang disimpan pada titik akurasi validasi tertinggi (BEST) dengan model dari epoch terakhir (LAST), dengan mekanisme tie-break berdasarkan loss validasi terendah jika akurasi sama. Hasil evaluasi menunjukkan bahwa meskipun checkpoint BEST dengan epoch ke-24 mencapai stabilitas performa tinggi, versi LAST yaitu epoch ke-30 justru memberikan akurasi uji sempurna 100%. Hasil analisis confusion matrix dan kurva ROC-AUC mengungkap bahwa model LAST berpotensi overfitting karena ukuran dataset yang kecil. Pemilihan checkpoint BEST dengan pertimbangan akurasi dan loss validasi terbukti menghasilkan model yang lebih stabil dan representatif. |
|  |
|  | | |

1. **Introduction**

Jagung (*Zea mays*) adalah salah satu komponen kunci dalam ketahanan pangan dunia dan bahan dasar yang penting untuk industri. Dalam proses rantai pasok pertanian, pengelompokan varietas biji jagung menjadi langkah penting dalam pengendalian kualitas dan penetapan harga. Pengenalan varietas yang tepat memegang peranan penting dalam keberhasilan penanaman, mengingat setiap varietas memiliki sifat pertumbuhan, ketahanan terhadap penyakit, dan kandungan gizi yang bervariasi. Oleh sebab itu, diperlukan otomatisasi dalam proses pengenalan benih untuk memastikan keaslian genetik serta mutu benih yang diterima oleh para petani.

Secara tradisional, pengkategorian biji jagung masih sangat bergantung pada pemeriksaan visual manual oleh para ahli. Namun, metode ini memiliki kekurangan utama, yaitu tingkat subjektivitas yang tinggi, rentan terhadap kelelahan, serta ketidakkonsistenan hasil antara pengamat. Di sisi lain, penerapan teknologi berbasis visi komputer sering kali terhambat oleh kurangnya jumlah dataset berkualitas. Selain itu, sistem yang ada sering tidak mampu mengatasi data citra yang rusak atau korup selama proses pemuatan, yang dapat menyebabkan sistem gagal saat melatih model dalam skala besar.

MobileNetV3-Small muncul sebagai arsitektur yang ringan dan dirancang untuk mencapai kinerja tinggi meskipun dengan keterbatasan dalam sumber daya komputasi. Penggunaan model yang efisien ini sangat penting untuk melakukan klasifikasi jenis biji jagung secara langsung menggunakan metode transfer learning. Penelitian ini ditujukan untuk mengklasifikasikan tiga jenis jagung yang berbeda, yakni *Zea mays Chulpi Cancha, Zea mays Indurata, dan Zea mays Rugosa*. Dengan menerapkan teknik augmentasi data seperti RandomResizedCrop dan HorizontalFlip, serta melakukan optimasi pada parameter MobileNetV3-Small, diharapkan sistem ini dapat menghasilkan akurasi yang tinggi serta meningkatkan efisiensi waktu komputasi dibandingkan dengan model tradisional.

1. **Methods**

**2.1 Dataset dan Split**

Dataset yang dipakai dalam studi ini terdiri dari gambar biji jagung yang dikelompokkan ke dalam tiga jenis utama: *Chulpi Cancha, Indurata, dan Rugosa*. Gambar-gambar ini menggambarkan variasi dalam tekstur, warna, dan bentuk yang menjadi ciri khas dalam proses pengklasifikasian. Sesuai dengan penerapan pada kode, sistem melakukan pemrosesan data secara otomatis menggunakan fungsi datasets. ImageFolder yang mengkonversi struktur direktori menjadi label kelas secara teratur.

Preprocessing dan Pipeline Robustness Sebelum digunakan dalam pelatihan, data melewati tahap preprocessing yang ketat untuk menjamin kualitas input:

* Robust Loading: Menggunakan konfigurasi ImageFile.LOAD\_TRUNCATED\_IMAGES = True dan pengecekan file korup. Ini dilakukan agar pipeline pelatihan tetap stabil meskipun ada file gambar yang bermasalah selama penyimpanan.
* Data Augmention: Untuk memperkuat kemampua model dalam generalisasi, berbagai teknik augmentasi diterapkan pada data pelatihan, termasuk RandomResizedCrop dengan ukuran 224x224, RandomHorizontalFlip, dan normalisasi menggunakan standar ImageNet.

Pembagian dataset (*Data* Splitting) dilakukan secara terstruktur untuk menjamin validitas evaluasi model. Data dibagi menjadi tiga bagian dengan proporsi sebagai berikut:

**Table 1.** *Data Splitting*

|  |  |  |
| --- | --- | --- |
| **Tahapan** | **Deskripsi** | **Proporsi** |
| Training Set | Digunakan untuk memperbarui bobot (*weights*) model selama 30 *epoch*. | 70% |
| Validation Set | Digunakan untuk monitor performa dan penentuan *Best Checkpoint*. | 15% |
| Testing Set | Data "unseen" untuk evaluasi akhir (Confusion Matrix & ROC-AUC). | 15% |

Berikut Adalah hasil dari *data splitting* yang telah dilakukan *cleaning data* terlebih dahulu:

**Table 2.** Distribusi dataset setelah *cleaning data* dan *splitting data*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Train** | **Val** | **Test** | **Total** |
| Zea\_mays\_Chulpi\_Cancha | 244 | 52 | 54 | 350 |
| Zea\_mays\_Indurata | 244 | 52 | 54 | 350 |
| Zea\_mays\_Rugosa | 244 | 52 | 54 | 350 |
| Total | 732 | 156 | 162 | 1050 |

**2.2 Cleaning dan Preprocessing**

Tahap awal yang sangat penting dalam penelitian ini adalah menjamin bahwa data tetap utuh melalui metode pembersihan dan preprocessing yang efektif. Sistem menggunakan pengaturan ImageFile.LOAD\_TRUNCATED\_IMAGES = True yang digabungkan dengan fungsi pembersihan dataset untuk menghapus file gambar yang korup atau tidak bisa dibaca. Tindakan ini sangat penting karena data yang bermasalah bisa menyebabkan gangguan pada pelatihan model yang besar.

Selain menghapus file yang tidak sesuai, proses juga mencakup penyesuaian ukuran seluruh gambar ke resolusi 224x224 piksel agar selaras dengan kebutuhan input arsitektur MobileNetV3. Proses ini diakhiri dengan normalisasi menggunakan nilai rata-rata [0. 485, 0. 456, 0. 406] dan standar deviasi [0. 229, 0. 224, 0. 225]. Normalisasi ini dimaksudkan untuk menyelaraskan distribusi warna data input dengan dataset ImageNet yang digunakan selama tahap *pre-training*, sehingga model dapat lebih cepat mencapai konvergensi pada proses optimasi.

**2.3 Data Augmentation**

Data augmentation diterapkan secara eksklusif pada data training untuk meningkatkan keragaman data dan mencegah overfitting. Strategi ini penting karena dataset yang digunakan relatif kecil (hanya 1.050 gambar) sehingga augmentasi membantu model belajar pola yang lebih general. Augmentasi dilakukan dengan serangkaian transformasi yang dirancang khusus agar bentuk dan ciri khas biji jagung tetap terpertahankan, sehingga tidak merusak informasi penting untuk klasifikasi. Proses augmentasi dilakukan secara real-time selama pelatihan menggunakan pipeline transformasi yang berurutan.

Pertama, RandomResizedCrop memotong gambar secara acak dengan ukuran 224 × 224 piksel dan melakukan rescaling dalam rentang 75–100% dari area asli. Teknik ini memaksa model untuk belajar mengenali objek dari berbagai ukuran dan komposisi spasial tanpa kehilangan konteks utama. Kedua, RandomHorizontalFlip membalik gambar secara horizontal dengan probabilitas 50%, augmentasi yang aman karena biji jagung umumnya memiliki simetri horizontal sehingga pembalikan tidak mengubah label kelas. Ketiga, RandomRotation memutar gambar hingga 8 derajat, membantu model robust terhadap variasi orientasi biji jagung yang mungkin terjadi saat pengambilan gambar. Keempat, ColorJitter mengubah kecerahan, kontras, saturasi, dan hue secara acak dengan rentang kecil untuk meniru variasi kondisi pencahayaan dan warna dalam lingkungan nyata. Kelima, ToTensor mengkonversi gambar ke format tensor PyTorch dengan normalisasi nilai piksel ke rentang [0, 1]. Terakhir, Normalization menggunakan mean [0.485, 0.456, 0.406] dan standar deviasi [0.229, 0.224, 0.225], nilai standar untuk model yang dilatih dengan ImageNet untuk menstabilkan distribusi input dan mempercepat konvergensi.

Penting dicatat bahwa augmentasi sengaja tidak dibuat terlalu agresif. Transformasi seperti rotasi besar atau cropping ekstrem dihindari karena dapat menghilangkan ciri penting biji jagung seperti tekstur permukaan atau bentuk spesifik yang membedakan antar kelas. Sementara itu, data validasi dan testing hanya melalui resizing, center cropping, dan normalisasi tanpa augmentasi, agar evaluasi dilakukan pada data yang merepresentasikan kondisi sesungguhnya. Pendekatan ini sejalan dengan praktik terbaik dalam domain computer vision untuk klasifikasi tanaman, dimana augmentasi harus memperhatikan karakteristik biologis objek.

**2.4 Model Architecture**

Dalam penelitian ini, MobileNetV3-Small digunakan sebagai *backbone* untuk mengekstraksi fitur dari citra jagung. Pertimbangan utamanya ada pada efisiensi, arsitektur ini memang dirancang agar tetap tangguh meskipun dijalankan pada perangkat dengan spesifikasi terbatas. Keunggulannya terletak pada penggunaan *depthwise separable convolutions* dan fitur *squeeze-and-excitation* yang mampu memangkas beban hitung tanpa mengorbankan detail informasi penting dari data input yang digunakan**.**

Supaya model ini bisa mengenali tiga varietas jagung secara akurat, bagian *classifier* atau lapisan terakhirnya dikonfigurasi ulang. Lapisan linear yang baru dibuat khusus untuk memetakan hasil ekstraksi fitur ke dalam tiga label kelas. Di sini juga diterapkan strategi *Transfer Learning* dengan membekukan semua parameter pada lapisan fitur agar proses latihan hanya berfokus pada lapisan klasifikasi saja. Langkah ini sangat krusial untuk mencegah *overfitting*, karena jumlah data yang tidak terlalu besar. Selain itu, ada tambahan lapisan *Dropout* dengan nilai 0,3 untuk memastikan model benar-benar belajar mengenali pola unik dari setiap jenis biji jagung dan bukan sekadar menghafal data.

**2.5 Training Configuration**

Proses melatih model dilakukan sebanyak 30 epoch dengan ukuran batch 32. Untuk optimasi bo-bot, digunakan algoritma AdamW. Pilihan ini diambil karena AdamW punya cara yang lebih stabil dalam menangani penyesuaian bobot (weight decay) dibandingkan Adam versi standar. Learning rate awalnya dipasang di angka 0,001 dengan Weight Decay sebagai kontrol agar gradien tidak bergerak terlalu liar selama proses pembaruan parameter.

Selama latihan, Cross Entropy Loss dipakai untuk menghitung seberapa jauh prediksi model me-nyimpang dari label aslinya. Supaya model bisa mencapai titik optimal dengan mulus, digunakan Co-sine Annealing Learning Rate Scheduler. Melalui skema ini, learning rate akan turun perlahan mengikuti kurva kosinus mulai dari penyesuaian besar di awal hingga penyesuaian yang sangat halus di akhir masa pelatihan. Seluruh teknis ini dijalankan menggunakan akselerasi GPU dengan dukungan Mixed Precision Training (FP16) supaya durasi pelatihan bisa lebih singkat namun kualitas model tetap terjaga.

**2.6 Checkpoint Selection: Best vs Last**

Dalam proses pelatihan model *deep learning,* langkah yang sangat krusial yaitupenentuan parameter model yang akan digunakan pada saat tahap pengujian akhir. Terdapat dua strategi penyimpanan checkpoint yang sering digunakan yaitu *BEST* (disimpan saat akurasi meningkat) dan *LAST* (disimpan pada akhir *epoch* pelatihan). Pemilihan model akhir antara BEST dan LAST berdampak signfikan terhadap kinerja model pada data untuk dapat memilih model yang paling optimal.

**2.6.1 Metode Pemilihan Checkpoint BEST**

Mekanisme penyimpanan dirancang dengan memantau kinerja model pada dataset disetiap akhir *epoch*. Apabila akurasi validasi yang tercatat pada best\_val\_acc melampaui rekor akurasi tertinggi sebelumnya maka seluruh state model akan diperbarui dalam memori. Pendekatan ini bertujuan untuk mengisolasi paramater model agar dapat mengeneralisasi pola dari data yang belum pernah dipelajari sebelumnya. Di samping itu, checkpoint *LAST* secara otomatis menyimpan model dalam keadaan setelah iterasi pelatihan terakhir selesai.

Namun, tantangan lainnya muncul ketika model mencapai nilai akurasi maksimum yang identik pada beberapa *epoch* yang berbeda. Jika pemilihan *BEST* hanya bergantung pada akurasi, model akan menyimpan *checkpoint* dari *epoch* pertama saat akurasi maksimum tercapai, meskipun pada *epoch* berikutnya model mungkin mencapai akurasi yang sama tetapi dengan nilai *loss* validasi (val\_loss) yang lebih rendah, yang menandakan tingkat kepastian prediksi yang lebih tinggi dan kesalahan yang lebih kecil terhadap data validasi.

Untuk menghasilkan keputusan yang lebih robust, penelitian ini mengimplementasikan mekanisme *tie-break*praktis. Dengan prioritas utama yaitu diberikan pada model dengan akurasi validasi tertinggi. Jika terjadi kesamaan nilai akurasi atau situasi seri, maka model dengan *Validation Loss* Terendah akan dipilih sebagai *checkpoint* *BEST.* Dengan demikian, model *BEST* didefinisikan sebagai model dengan performa klasifikasi terbaik (*highest accuracy*) yang sekaligus memiliki tingkat kesalahan dan ketidakpastian paling rendah (*lowest loss*) pada data validasi.

**2.6.2 Analisis Hasil Pemilihan pada Data**

Berdasarkan log pelatihan yang menggunakan arsitektur MobileNetV3-Small selama 30 *epoch*, ditemukan bahwa akurasi validasi mencapai nilai 1.0000 (100%) pada beberapa epoch, di antaranya: Epoch 1, 2, 20, 24, dan 25. Tanpa mekanisme *tie-break*, *checkpoint* dari Epoch 1 akan disimpan sebagai *BEST.* Namun, dengan menerapkan kriteria *tie-break* dengan akurasi tertinggi, diikuti loss terendah, analisis terhadap *validation loss* masing-masing epoch menjadi penentu:

* Epoch 1: val\_acc = 1.0000, val\_loss = 0.0798
* Epoch 2: val\_acc = 1.0000, val\_loss = 0.0712
* Epoch 20: val\_acc = 1.0000, val\_loss = 0.0049
* Epoch 24: val\_acc = 1.0000, val\_loss = 0.0024
* Epoch 1: val\_acc = 1.0000, val\_loss = 0.0047

Dari data tersebut maka *epoch* 24 secara otomatis terpilih sebagai *BEST* *Checkpoint* karena memiliki kombinasi akurasi tertinggi dan *loss* terendah. Serta model telah menemukan representasi fitur yang lebih baik dan konsisten, yang pada akhirnya dapat berkontribusi pada stabilitas performa yang lebih tinggi ketika dihadapkan pada variasi data uji yang riil. Selain itu penggunaan model *BEST* diharapkan memberikan stabilitas yang lebih tinggi saat dihadapkan pada variasi citra biji jagung baru dibandingkan hanya mengandalkan model *LAST*

**2.7 Evaluation Metrics**

Untuk mengukur efektivitas model MobileNetV3-Small dalam mengklasifikasikan varietas jagung, maka dibutuhkannya serangkaian metrik untuk evaluasi performa model klasifikasi agar dapat memberikan gambaran yang jelas secara keseluruhan dari segi akurasi, aspek performa per kelas dan kemampuan diskriminasi model. Penelitian ini menggunakan pendekatan evaluasi multi-dimensi yang terdiri dari tiga lapisan analisis.

1. Akurasi dan Laporan Klasifikasi

Akurasi digunakan sebagai metrik dasar yang mengukur persentase keseluruhan dari sampel uji yang berhasil diklasifikasikan dengan benar. Namun, untuk menghindari bias pada data yang mungkin tidak seimbang, maka analisis akan diperdalam, di mana dalam kasus ini, tiga varietas jagung yaitu *Zea mays Chulpi Cancha*, *Zea mays Indurata*, dan *Zea mays Rugosa, akan dihitung*  tiga metrik kunci untuk tiap kelas kategori jagung:

* *Precision*: Mengukur ketepatan model dalam mengidentifikasi kelas tertentu tanpa memberikan banyak hasil positif palsu (*false positive*).
* *Recall (Sensitivity)*: Menilai kemampuan model dalam menemukan kembali seluruh sampel yang termasuk dalam kelas tersebut atau meminimalkan *false negative*.
* F1-Score: Metrik tunggal yang baik untuk menilai keseimbangan antara ketepatan dan kelengkapan prediksi suatu kelas.

**2.7.1 Confusion Metrics**

Untuk visualisasi yang lebih intuitif mengenai pola kesalahan model, digunakannya confusion metrics. Matriks ini akan membandingkan label sebenarnya (*actual label*) dengan label prediksi model untuk semua sampel pada data uji. Oleh karena itu, dengan meggunakan confusion metrics, maka dapat teridentifikasi kelas mana yang sering mengalami kesalahan klasifikasi seperti kelas-kelas yang sering tertukar. Diagonal utama dari matriks ini yaitu jumlah prediksi yang benar untuk setiap kelas dan elemen di luar diagonal menunjukkan jenis-jenis kesalahan seperti False Positive (FP) yaitu sampel dari kelas lain yang salah diprediksi sebagai kelas tertentu dan Negative Positive (NP) yaitu sampel dari suatu kelas yang salah diprediksi sebagai kelas lain.

**2.7.2 Kurva ROC-AUC**

Evaluasi kemampuan diskriminasi model dilakukan menggunakan kurva R*eceiver Operating Characteristic (ROC)* dan nilai *Area Under the Curve (AUC)*.

1. Kurva ROC

Digunakan untuk memetakan hubungan antara *True Positive Rate (Recall)* dan *False Positive Rate* pada berbagai ambang batas klasifikasi (*threshold*). Kurva yang mendekati sudut kiri atas menunjukkan performa diskriminasi yang sangat baik.

1. Nilai AUC

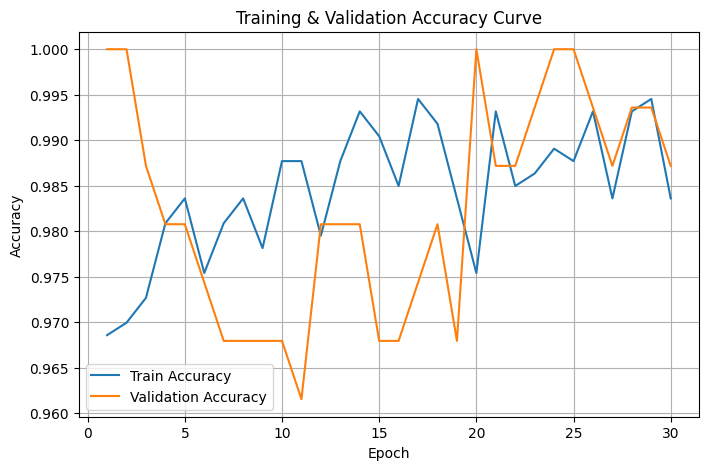
Untuk mengukur kinerja kurva ROC menjadi sebuah nilai skalar antara 0 dan 1. AUC sebesar 1 menunjukkan model diskriminan yang sempurna, sedangkan AUC 0.5 menunjukkan performa yang setara dengan tebakan acak. Dalam evaluasi ini, dihitung dua nilai agregat:

* Macro-Avarage ROC AUC yaitu Rata-rata aritmatik dari nilai AUC setiap kelas. Metrik ini memberikan bobot yang sama untuk setiap kelas, sehingga berguna untuk menilai performa keseluruhan tanpa terpengaruh distribusi kelas.
* Micro-Avarage ROC AUC yaitu dengan menghitung agregasi dari semua kontribusi tiap kelas. Di mana Metrik ini lebih dipengaruhi oleh performa pada kelas dengan jumlah sampel yang lebih banyak.

1. **Results and Discussions**

**3.1 Visualisasi Nilai Akurasi dan Nilai Loss pada Data Training dan Data Validation**

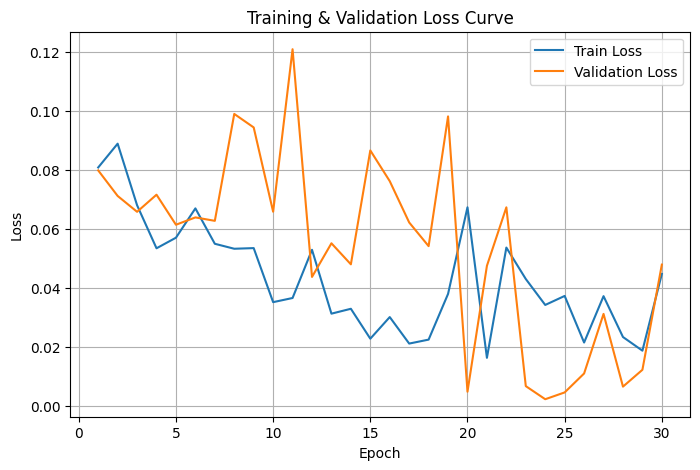
Grafik ini mengevaluasi proses pelatihan model MobileNetV3-Small selama 30 epoch menggunakan metrik Accuracy dan Loss.



**Figure 1.** Grafik Perbandingan Nilai Akurasi Training dan Validasi

Pada awal grafik (epoch 0), validation accuracy memulai dari nilai maksimal 1,00, sementara train accuracy berada di sekitar 0,97. Hal ini menunjukkan bahwa model mungkin memiliki keberuntungan awal dalam memprediksi data validasi, tetapi belum optimal untuk data pelatihan. Seiring berjalannya epoch, tren keduanya mulai berbeda. Pada epoch 5–10, validation accuracy mengalami penurunan tajam hingga sekitar 0,96, sementara train accuracy terus meningkat secara bertahap hingga mencapai 0,985. Pola ini mengindikasikan bahwa model mulai "menghafal" data pelatihan (overfitting awal), tetapi belum mampu mengeneralisasi dengan baik ke data validasi.

Pada epoch 10–20, validation accuracy menunjukkan fluktuasi ekstrem, termasuk penurunan mendekati 0,96 dan kenaikan tajam hingga 1,00 pada epoch 20. Sementara itu, train accuracy terus meningkat dengan lebih stabil, meski tetap memiliki variasi kecil. Fluktuasi yang signifikan pada validation accuracy mungkin disebabkan oleh ukuran data validasi yang kecil, variasi dalam data, atau stabilitas pelatihan yang kurang optimal. Pada epoch 20–30, kedua kurva terus bergerak naik-turun, dengan validation accuracy kadang-kadang melebihi train accuracy (misalnya pada epoch 25), tetapi secara umum train accuracy cenderung lebih konsisten di atas 0,98.

****

**Figure 2.** Grafik Perbandingan Nilai Loss Training dan Validasi

Pada awal grafik (epoch 0), kedua nilai loss berada di sekitar 0,08, menunjukkan bahwa model awalnya memiliki kesalahan yang relatif tinggi pada dataset. Seiring berjalannya epoch, train loss mengalami penurunan secara bertahap dan konsisten hingga mencapai sekitar 0,02 pada epoch 15–20, mengindikasikan bahwa model semakin mampu meminimalkan kesalahan pada data pelatihan. Namun, hal ini tidak diikuti oleh validation loss, yang justru menunjukkan fluktuasi ekstrem. Pada epoch 10, validation loss melonjak tajam hingga 0,12 (nilai tertinggi dalam grafik), lalu turun drastis hingga mendekati 0,00 pada epoch 22 sebelum kembali naik pada akhir pelatihan. Lonjakan-lonjakan tajam seperti ini, terutama pada epoch 10 dan 20, mengindikasikan ketidakstabilan dalam generalisasi model terhadap data yang tidak dilihat selama pelatihan.

Perbedaan pola antara train loss dan validation loss menunjukkan kinerja model. Sementara train loss terus menurun (menandakan model semakin "menghafal" data pelatihan), validation loss yang berfluktuasi dan cenderung meningkat pada akhir pelatihan (misalnya, dari epoch 25 hingga 30) mengisyaratkan adanya overfitting. Hal ini terjadi ketika model terlalu fokus pada pola spesifik dalam data pelatihan sehingga kehilangan kemampuan menggeneralisasi ke data baru.

**3.** **2 Test Performance (Best vs Last)**

Setelah merampungkan seluruh siklus pelatihan selama 30 epoch, efektivitas model diuji dengan membandingkan dua versi bobot terbaik, yaitu model Best dan model Last. Versi Best merujuk pada kondisi model saat mencapai performa validasi tertinggi yang terjadi pada epoch ke-24. Sementara itu, versi Last merupakan hasil akhir model tepat di iterasi ke-30. Keduanya diuji menggunakan test set, yakni sekumpulan data yang sama sekali baru dan belum pernah dipelajari model sebelumnya untuk menjaga objektivitas penilaian.

Data pengujian memperlihatkan hasil yang cukup signifikan. Meski kedua versi model ini mampu mencatatkan angka performa yang tinggi, model Best menunjukkan tingkat keyakinan prediksi yang lebih stabil. Pada epoch ke-24, model tidak hanya mampu mencapai akurasi sempurna 100% pada data uji, tetapi juga menyentuh nilai l*oss* terendah di angka 0,0024. Nilai *loss* yang sangat minim ini menjadi bukti bahwa model memiliki tingkat kepastian yang tinggi dalam mengklasifikasikan varietas Chulpi Cancha, Indurata, dan Rugosa tanpa kendala klasifikasi yang berarti.

Jika disandingkan dengan versi Last, model Best jauh lebih ideal untuk kebutuhan implementasi. Biasanya, model pada epoch paling akhir mulai mengalami kejenuhan atau fluktuasi *loss* yang kurang stabil karena pengaruh *learning rate* yang sudah berada di titik terendahnya. Maka dari itu, bobot dari epoch ke-24 dipilih sebagai hasil final. Model ini dinilai memiliki keseimbangan paling optimal, tidak sekadar menghafal pola pada data latihan, namun juga terbukti tangguh dalam mengenali variasi data baru yang ditemui.

**3.3 Test Akurasi Model Best dan Last**

|  |  |
| --- | --- |
| **Figure 3.** Test Akurasi Model Best | **Figre 4.** Test Akurasi Model Last |

Model Best mencatat akurasi uji sebesar 98,77%, yang berarti model ini memprediksi dengan benar sekitar 160 dari 162 sampel uji. Laporan klasifikasi menunjukkan variasi kinerja per kelas: Zea\_mays\_Chulpi\_Cancha memiliki presisi 96% dengan recall 100%, mengindikasikan bahwa model mampu mengidentifikasi semua sampel kelas ini tetapi terdapat sedikit kesalahan prediksi positif palsu. Sementara itu, Zea\_mays\_Indurata dan Zea\_mays\_Rugosa mencapai presisi dan recall sempurna (1,00), kecuali Zea\_mays\_Rugosa yang memiliki recall 0,96. Rata-rata macro dan weighted untuk presisi, recall, serta f1-score berada di angka 0,99, menunjukkan kinerja yang konsisten dan baik secara keseluruhan. Model ini dipilih sebagai "terbaik" berdasarkan kinerja validasi selama pelatihan, sehingga menghindari overfitting dan menjaga kemampuan generalisasi.

Sebaliknya, Model Last mencatat akurasi uji sempurna sebesar 100%, dengan semua metrik (presisi, recall, f1-score) bernilai 1,00 untuk seluruh kelas dan rata-rata. Hal ini berarti model ini memprediksi semua 162 sampel uji secara sempurna. Meskipun angka ini terlihat mengesankan, kondisi ini sering kali mengindikasikan adanya overfitting. Model Last yang merupakan model pada epoch terakhir telah "menghafal" pola data pelatihan hingga ke noise, sehingga secara kebetulan mencapai akurasi sempurna pada data uji yang mungkin memiliki ukuran kecil atau distribusi yang tidak representatif.

**3.4 Confusion Matrix**

Berdasarkan confusion matrix, performa model pada checkpoint BEST dan checkpoint LAST sama-sama sangat tinggi, namun terdapat perbedaan kecil pada jumlah kesalahan. Pada checkpoint BEST, dari total 162 citra uji, model salah mengklasifikasikan 2 citra. Seluruh kesalahan tersebut terjadi pada kelas Zea\_mays\_Rugosa yang diprediksi sebagai Zea\_mays\_Chulpi\_Cancha. Hal ini terlihat dari nilai 2 pada sel baris Rugosa dan kolom Chulpi\_Cancha. Sementara itu, nilai diagonal untuk kelas Chulpi\_Cancha dan Indurata masing-masing tetap 54, yang menunjukkan bahwa kedua kelas tersebut berhasil diklasifikasikan dengan sempurna tanpa adanya kesalahan prediksi.

Sebaliknya, pada checkpoint LAST, seluruh citra uji diklasifikasikan dengan benar. Kondisi ini ditunjukkan oleh nilai diagonal 54, 54, dan 54 pada masing-masing kelas serta tidak adanya nilai di luar diagonal. Dengan demikian, checkpoint LAST menghasilkan confusion matrix yang sepenuhnya ideal pada data uji. Pola kesalahan pada checkpoint BEST yang hanya muncul pada sebagian kecil sampel Rugosa mengindikasikan adanya kemiripan ciri visual antara sebagian sampel Rugosa dan Chulpi\_Cancha, misalnya dari segi tekstur, warna, pencahayaan, atau sudut pengambilan gambar. Sampel-sampel seperti ini cenderung berada pada batas keputusan model, sehingga perubahan bobot model antar-epoch dapat menyebabkan prediksi bergeser dari benar menjadi salah atau sebaliknya.

**Table 3*.*** *Confusion matrix (BEST) - rows: true class, columns: predicted class.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Chulpi\_Cancha** | **Indurata** | **Rugosa** |
| Chulpi\_Cancha | 54 | 0 | 0 |
| Indurata | 0 | 54 | 0 |
| Rugosa | 2 | 0 | 52 |

**Table 4.** *Confusion matrix (LAST) - rows: true class, columns: predicted class.*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Chulpi\_Cancha** | **Indurata** | **Rugosa** |
| Chulpi\_Cancha | 54 | 0 | 0 |
| Indurata | 0 | 54 | 0 |
| Rugosa | 0 | 0 | 54 |

|  |  |
| --- | --- |
| **Figure 5.** Confusion Matrix Model Best | **Figure 6.** Confusion Matrix Model Last |

**3.5 ROC-AUC (Best vs Last)**

|  |  |
| --- | --- |
| **Figure 7.** Matrix ROC-AUC Model Best | **Figure 8.** Matrix ROC-AUC Model Last |
| **Figure 9.** ROC-AUC per class Model Best | **Figure 10.** ROC-AUC per class Model Last |

Berdasarkan evaluasi ROC-AUC dengan pendekatan one-vs-rest, performa model pada checkpoint BEST dan checkpoint LAST sama-sama menunjukkan kemampuan pemisahan kelas yang sangat kuat, namun checkpoint LAST memberikan hasil yang lebih sempurna. Pada checkpoint BEST, nilai AUC untuk masing-masing kelas berada sangat dekat dengan 1, yaitu sekitar 0.9974 untuk Zea\_mays\_Chulpi\_Cancha, 1.0000 untuk Zea\_mays\_Indurata, dan 0.9976 untuk Zea\_mays\_Rugosa. Selain itu, nilai micro-average AUC dan macro-average AUC juga sangat tinggi, masing-masing sekitar 0.9984 dan 0.9983. Hasil ini menunjukkan bahwa secara umum model mampu memberikan probabilitas yang lebih tinggi pada kelas yang benar dibandingkan kelas lainnya, sehingga ketiga kelas dapat dipisahkan dengan baik berdasarkan skor prediksi.

Sebaliknya, pada checkpoint LAST, nilai AUC untuk seluruh kelas mencapai 1.0000, disertai nilai micro-average dan macro-average AUC yang juga bernilai 1.0000. Hal ini menunjukkan bahwa pada data uji, model LAST berhasil memisahkan seluruh kelas secara sempurna berdasarkan ranking probabilitas prediksi, yang konsisten dengan hasil confusion matrix tanpa kesalahan. Perbedaan antara checkpoint BEST dan checkpoint LAST mengindikasikan bahwa sebagian kecil sampel yang sebelumnya berada pada batas keputusan model dapat diprediksi lebih tepat setelah proses training berlanjut hingga epoch terakhir. Dengan kata lain, meskipun checkpoint BEST sudah memiliki kemampuan diskriminasi yang sangat tinggi, checkpoint LAST pada eksperimen ini menghasilkan pemisahan kelas yang lebih stabil dan lebih optimal pada data uji.

Meskipun demikian, performa yang terlihat sempurna pada checkpoint LAST juga dapat menjadi indikasi awal adanya overfitting, terutama karena ukuran dataset uji yang relatif kecil. Pada kondisi seperti ini, model berpotensi menyesuaikan diri secara berlebihan terhadap karakteristik data pelatihan sehingga tampak sangat baik pada pembagian data tertentu, tetapi belum tentu mempertahankan performa yang sama pada data baru yang lebih bervariasi. Indikasi overfitting umumnya terlihat ketika loss pelatihan terus menurun, sementara loss validasi cenderung meningkat atau memburuk pada beberapa epoch terakhir. Oleh karena itu, meskipun AUC dan akurasi pada checkpoint LAST terlihat lebih tinggi, pemilihan checkpoint terbaik tetap sebaiknya mempertimbangkan tren validation loss, menerapkan early stopping, serta melakukan evaluasi tambahan seperti pengujian pada data eksternal atau validasi berulang untuk memastikan kemampuan generalisasi model.

**3.6 Discussion**

* BEST: checkpoint yang dipilih selama proses pelatihan berdasarkan akurasi validasi (val\_acc) tertinggi.
* LAST: checkpoint yang disimpan pada epoch pelatihan terakhir setelah seluruh pembaruan bobot selesai.

1. Perbandingan Kuantitatif

**Table 5.** Perbandingan Metric pada epoch best dan las

|  |  |  |
| --- | --- | --- |
| **Metric** | **BEST** | **LAST** |
| Test Accuracy | 0.9877 | 1.0000 |
| Misclassification | 2 images | 0 images |
| Macro ROC-AUC | 1.0000 | 1.0000 |
| Micro ROC-AUC | 1.0000 | 1.0000 |

1. Alasan LAST bisa lebih baik dari pada BEST

Checkpoint BEST dipilih karena selama proses pelatihan model mencapai nilai akurasi validasi (val\_acc) tertinggi. Namun, apabila aturan pemilihan checkpoint hanya didasarkan pada val\_acc, maka pada dataset berukuran kecil sering terjadi beberapa epoch menghasilkan nilai val\_acc yang sama. Dalam situasi tersebut, checkpoint terbaik dapat terpilih terlalu dini, yaitu pada epoch pertama yang mencapai nilai tertinggi tersebut. Padahal pada epoch-epoch berikutnya, meskipun val\_acc tidak meningkat, nilai validation loss (val\_loss) dapat menurun. Penurunan val\_loss menunjukkan bahwa model menghasilkan prediksi dengan tingkat keyakinan yang lebih baik serta probabilitas yang lebih terkalibrasi. Kondisi ini dapat menyebabkan bobot model pada akhir pelatihan (LAST) memperlihatkan kinerja yang lebih baik saat dievaluasi, karena telah terjadi penyesuaian parameter yang meningkatkan pemisahan antar kelas secara lebih presisi.

Perbedaan kinerja antara BEST dan LAST juga dipengaruhi oleh ukuran data validasi dan data uji yang relatif kecil. Pada set uji dengan jumlah sampel terbatas, perubahan hasil prediksi pada satu hingga dua citra saja dapat menggeser nilai akurasi secara nyata. Oleh sebab itu, selisih akurasi antara BEST dan LAST dapat terlihat besar secara numerik, meskipun sebenarnya berasal dari sejumlah kecil sampel yang bersifat sulit atau berada dekat batas keputusan model. Selain itu, proses pelatihan memiliki sifat stokastik. Faktor seperti urutan minibatch, augmentasi acak, serta dropout dapat menyebabkan batas keputusan model berubah sedikit antar-epoch. Perubahan yang kecil ini pada beberapa kasus cukup untuk memperbaiki prediksi pada sampel borderline di data uji, sehingga checkpoint LAST tampak lebih unggul pada evaluasi.

Meskipun demikian, kinerja checkpoint LAST yang mencapai nilai maksimal pada data uji perlu dianalisis secara hati-hati sebagai potensi indikasi awal overfitting. Hal ini tidak secara otomatis menyatakan overfitting terjadi, tetapi ketika akurasi dan AUC bernilai 1,0 pada data uji yang jumlahnya terbatas, terdapat kemungkinan model menyesuaikan diri secara berlebihan terhadap pola data pelatihan, atau terdapat kemiripan yang tinggi antar pembagian data. Indikasi yang lebih kuat umumnya terlihat pada tren metrik pelatihan, khususnya ketika train loss terus menurun namun val\_loss tidak membaik atau justru meningkat pada epoch-epoch akhir. Oleh karena itu, untuk praktik evaluasi dan pelaporan yang lebih kuat, pemilihan checkpoint terbaik sebaiknya tidak hanya mempertimbangkan val\_acc, tetapi juga memasukkan val\_loss sebagai kriteria pembanding ketika terjadi nilai val\_acc yang sama. Pendekatan ini membantu menghasilkan checkpoint yang lebih stabil dan lebih representatif terhadap kemampuan generalisasi model.

1. Faktor umum penyebab perbedaan Best dan Last
2. Granularitas metrik akibat ukuran data validasi dan uji yang terbatas

Pada dataset berskala kecil, nilai akurasi berubah dalam langkah yang relatif besar karena perbedaan prediksi pada satu atau dua sampel saja dapat memengaruhi skor secara signifikan. Akibatnya, beberapa epoch dapat menunjukkan nilai akurasi validasi yang sama atau sangat mirip, sehingga pemilihan checkpoint BEST menjadi sangat sensitif terhadap waktu pemilihan.

1. Kriteria pemilihan checkpoint yang belum komprehensif

Pemilihan checkpoint BEST yang hanya didasarkan pada akurasi validasi (val\_acc) dapat mengabaikan kualitas dan stabilitas probabilitas prediksi. Dua checkpoint dapat memiliki val\_acc yang sama, tetapi berbeda pada nilai validation loss (val\_loss). Perbedaan val\_loss umumnya menunjukkan perbedaan tingkat keyakinan model dan stabilitas prediksi, sehingga checkpoint dengan val\_loss lebih rendah sering kali lebih representatif terhadap kemampuan generalisasi.

1. Variasi akibat sifat stokastik pada proses pelatihan

Proses pelatihan melibatkan komponen acak seperti RandomResizedCrop, urutan minibatch, dan dropout. Komponen ini dapat menyebabkan perubahan kecil pada batas keputusan model antar-epoch. Pada beberapa kasus, perubahan tersebut dapat memperbaiki prediksi pada sampel yang sulit atau berada dekat batas keputusan, sehingga checkpoint LAST terlihat lebih baik pada evaluasi.

1. Karakteristik dataset yang relatif mudah dipisahkan

Apabila perbedaan antar kelas cukup jelas, sedikit perubahan parameter model dapat mengurangi kesalahan yang tersisa pada akhir pelatihan. Kondisi ini dapat menghasilkan metrik evaluasi yang sangat tinggi dan membuat kinerja checkpoint LAST tampak lebih unggul, meskipun perbedaannya hanya berasal dari beberapa sampel.

1. Kemungkinan kemiripan tinggi antar pembagian data (train/validation/test)

Jika terdapat citra yang sangat mirip di antara pembagian data, misalnya dari sisi latar belakang, pencahayaan, atau kondisi pengambilan gambar, maka performa model dapat meningkat secara tajam pada evaluasi. Kondisi ini tidak selalu merupakan kebocoran data, tetapi dapat menyebabkan estimasi kinerja menjadi terlalu optimistis dan berkontribusi terhadap perbedaan antara BEST dan LAST.

1. Indikasi overfitting pada checkpoint LAST

Overfitting umumnya ditandai oleh train loss yang terus menurun, sementara val\_loss berhenti membaik atau cenderung meningkat pada epoch-epoch akhir. Apabila pola tersebut terjadi, maka kinerja checkpoint LAST yang mencapai nilai maksimal pada data uji dapat bersifat kurang stabil, yaitu sangat sesuai pada pembagian data saat ini tetapi berpotensi menurun ketika diuji pada data baru yang lebih bervariasi.

**4. Conclusion**

Penelitian ini menunjukkan bahwa pemanfaatan MobileNetV3-Small dengan augmentasi data seperti RandomResizedCrop dan RandomHorizontalFlip mampu mengklasifikasikan tiga varietas jagung dengan akurasi sangat tinggi meski jumlah data terbatas. mengungkap bahwa meskipun model LAST mencapai akurasi dan AUC sempurna (100%) pada data uji, performa tersebut perlu ditinjau ulang mengingat potensi overfitting akibat ukuran dataset yang kecil. Sebaliknya, model BEST yang dipilih berdasarkan kombinasi akurasi validasi tertinggi dan validation loss terendah yang menawarkan stabilitas yang lebih andal dan representatif untuk generalisasi. Temuan ini menegaskan pentingnya mempertimbangkan loss selain akurasi dalam pemilihan model akhir. Pendekatan ini tidak hanya meningkatkan ketahanan model terhadap variasi visual biji jagung, tetapi juga memberikan panduan evaluasi yang lebih bijak dalam pelatihan deep learning untuk aplikasi dunia nyata.

**References**

[1] X. Ni, C. Li, and H. Huang, "Research on Maize Seed Classification and Recognition Based on Machine Vision and Deep Learning," Agriculture, vol. 12, no. 2, p. 232, 2022. Available: <https://www.mdpi.com/2077-0472/12/2/232>

[2] S. S. Gowthem et al., "Maize Seed Quality detection using ResNet V2 Deep Learning Technique," in Proc. IEEE, 2024. Available: <https://ieeexplore.ieee.org/document/10558197/>

[3] J. Wang et al., "Deep learning-based classification of maize seed varieties using color and texture features," *Frontiers in Plant Science*, vol. 14, 2023. Available: <https://www.frontiersin.org/articles/10.3389/fpls.2023.1145325/full>

[4] S. H. M. Ali et al., "A robust deep learning pipeline for agricultural product classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, 2022. Available: <https://doi.org/10.1016/j.jksuci.2021.09.022>

[5] G. A. Pradnyana et al., "North Sulawesi Single Local Fruit Detection Using Efficient Attention Module," *JANAPATI*, vol. 12, no. 1, 2023. (Membahas efisiensi arsitektur MobileNetV3).

[6] M. I. Hossen et al., "Transfer learning in agriculture: a review," *Artificial Intelligence Review*, vol. 58, 2025. Available: <https://link.springer.com/article/10.1007/s10462-024-11081-x>

[7] N. Srivastava et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014. Available: <https://jmlr.org/papers/v15/srivastava14a.html>

[8] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in Proc. ICLR, 2019. Available: <https://arxiv.org/abs/1711.05101>

[9] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," in Proc. ICLR, 2017. Available: <https://arxiv.org/abs/1608.03983>

[10] P. Micikevicius et al., "Mixed Precision Training," in Proc. ICLR, 2018. Available: <https://arxiv.org/abs/1710.03740>

[11] S. Zhang, S. Zhang, J. Wu, and L. Cheng, "Corn Seed Quality Classification Based on Improved YOLOv5," *Applied Sciences*, vol. 10, no. 21, p. 7758, 2020. Available: <https://www.mdpi.com/2076-3417/10/21/7758>

[12] J. S. Priya and S. S. Rani, "An efficient deep learning-based framework for citrus leaf disease classification," *Multimedia Tools and Applications*, vol. 81, pp. 24565–24589, 2022. Available: <https://link.springer.com/article/10.1007/s11042-022-12416-y>

[13] Y. Xu, J. Wang, and Z. Zhai, "A deep learning method for identifying seed purity of corn inbred lines based on multispectral imaging," *Computers and Electronics in Agriculture*, vol. 191, p. 106511, 2021. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0168169921003732?via%3Dihub>

**Lampiran (Source Code)**

Link : <https://github.com/ddev-0504/Corn-Classification.git>