

Optimasi Hyperparameter Convolutional Neural Network untuk Klasifikasi Citra Makanan Selingan

¹Kunti Eliyen, ²Abidatul Izzah, ³Toga Aldila Cinderatama, ⁴Rinanza Zulmy Alhamri

^{1*,2,3,4}Manajemen Informatika, Politeknik Negeri Malang, Malang, Indonesia

*Korespondensi: kunti.eliyen@polinema.ac.id

Submit : 05 Jul 2026 | Diterima : 29 Jul 2026 | Terbit : 31 Jul 2026

ABSTRACT

Hyperparameter optimization is a fundamental step in improving the performance of a Convolutional Neural Network (CNN) model. This study aims to optimize hyperparameters in the EfficientNetB0 architecture using a Genetic Algorithm (GA) to improve the classification accuracy of snack images. The snack images dataset consists of 250 images across 10 snack food classes, with 80% allocated for training data and 20% for validation data. The optimization process is carried out by determining the best values of the learning rate, dropout rate, and batch size. The GA method is applied to find the optimal combination through selection, crossover, and mutation processes over multiple generations. The results show that the use of GA determines a combination of hyperparameters that improve model accuracy compared to manual selection. The best hyperparameter combination obtained was a learning rate of 0.000670, a dropout rate of 0.34, and a freeze ratio of 0.73. To forestall the overfitting, during training was applied early stopping and learning rate reduction. The model was trained for 40 epochs and obtained 92% of validation accuracy. These results demonstrate the effectiveness of GA in accelerating the process of finding optimal hyperparameters in EfficientNetB0-based CNNs, particularly for small-scale food image datasets prone to overfitting.

Keywords: Convolutional Neural Network, EfficientNetB0, Genetic Algorithm, hyperparameter, optimization

ABSTRAK

Optimasi hyperparameter adalah langkah mendasar dalam meningkatkan kinerja model *Convolutional Neural Network* (CNN). Penelitian ini bertujuan untuk mengoptimalkan hyperparameter dalam arsitektur EfficientNetB0 menggunakan *Genetic Algorithm* (GA) untuk meningkatkan akurasi klasifikasi gambar snack. Dataset citra makanan selingan terdiri dari 250 citra yang diklasifikasikan dalam 10 kelas, dengan 80% dialokasikan untuk data pelatihan dan 20% untuk data validasi. Proses optimasi dilakukan dengan menentukan nilai terbaik dari *learning rate*, *dropout rate*, dan *batch size*. Metode GA diterapkan untuk menemukan kombinasi optimal melalui proses seleksi, crossover, dan mutasi selama beberapa generasi. Hasilnya menunjukkan bahwa penggunaan GA dapat menentukan kombinasi hyperparameter yang meningkatkan akurasi model dibandingkan dengan seleksi manual. Kombinasi hyperparameter terbaik yang diperoleh adalah *learning rate* 0,000670, *dropout rate* 0,34, dan *freeze ratio* 0,73. Untuk mencegah overfitting, selama pelatihan diterapkan penghentian dini dan pengurangan tingkat *training*. Model ini dilatih untuk 40 epoh dan memperoleh 92% akurasi validasi. Hasil ini menunjukkan efektivitas GA dalam mempercepat proses menemukan hiperparameter optimal di CNN berbasis EfficientNetB0, terutama untuk kumpulan data citra makanan skala kecil yang rentan terhadap *overfitting*.

Kata Kunci: EfficientNetB0, *Fine-tuning*, *Freeze Ratio*, Genetic Algorithm, Transfer Learning

PENDAHULUAN

Convolutional Neural Network (CNN) adalah arsitektur *deep learning* yang banyak digunakan dalam pengenalan citra karena kemampuannya untuk mengekstrak fitur secara otomatis (Santoso et al, 2024). Namun, kinerja CNN sangat dipengaruhi oleh pemilihan hyperparameter seperti learning rate, dropout rate, and batch size. Pemilihan hyperparameter yang tidak tepat dapat menyebabkan model *overfitting* atau *underfitting* (Ansari et al., 2024).

Proses pencarian hyperparameter yang optimal biasanya dilakukan baik secara manual

atau melalui pendekatan sistematis seperti *grid search* dan *random search* (Franceschi et al., 2025)). Misalnya, dalam studi optimasi hiperparameter pada jaringan saraf konvolusional (CNN) untuk deteksi COVID-19 dari gambar sinar-X dada, akurasi 95,38% tercapai. Pencarian acak hanya mengevaluasi sejumlah kombinasi hiperparameter, yang umumnya membutuhkan lebih sedikit waktu daripada pencarian kisi untuk mengidentifikasi model yang memaksimalkan akurasi. (Santoso, Sediyo dan Purnomo, 2024).

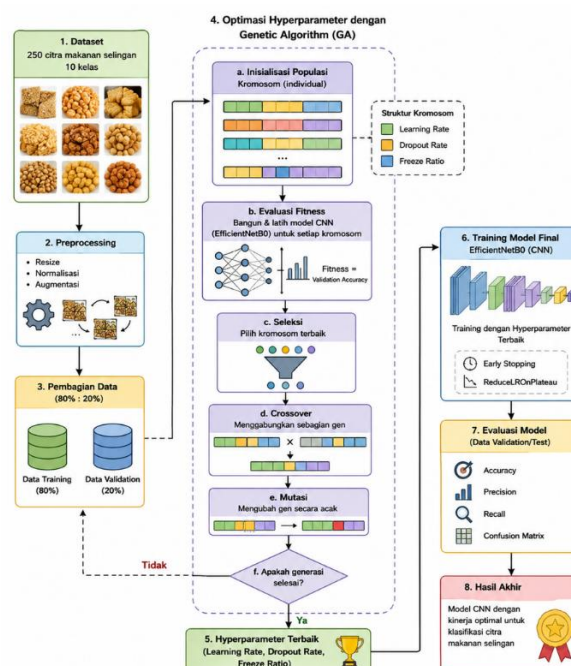
Metode augmentasi dan hiperparameter juga digunakan dalam mengoptimalkan arsitektur CNN EfficientNet-B0 dalam klasifikasi citra daging sapi dan babi (Jasril dan Sanjaya, 2024) akurasi tertinggi sebesar 0,93 dicapai dengan menggunakan dataset gabungan yang terdiri dari data asli dan augmented. Model dengan konfigurasi Adamax optimizer, fungsi aktivasi ReLU, dan learning rate 0,01 tidak menunjukkan indikasi overfitting, sehingga dapat disimpulkan bahwa model ini adalah yang paling optimal dibandingkan dengan model lainnya.

Pendekatan yang banyak digunakan dalam optimasi hiperparameter adalah GA yang merupakan algoritma optimasi berbasis evolusi yang meniru proses seleksi alam. Dengan konsep populasi, seleksi, persilangan, dan mutasi, GA dapat menemukan solusi optimal melalui eksplorasi adaptif ruang pencarian.

Banyak penelitian telah menerapkan GA untuk mengoptimalkan hiperparameter CNN. Han et al. (2020) menunjukkan efektivitas GA untuk menentukan hiperparameter CNN dalam klasifikasi citra. Shakibhamedan dkk. (2025) merekomendasikan gabungan metode GA-CNN untuk mengoptimalkan parameter arsitektur, seperti kernel size dan jumlah lapisan pada kumpulan data benchmark MNIST. Wojciuk et al. (2024) menganalisis hasil optimasi hiperparameter dalam transfer learning menggunakan teknik seperti optimasi Bayesian dan random search. Somani dan Thangavelu (2023) memperkenalkan GA dengan diversity-guided dalam mengoptimalkan CNN untuk mencegah konvergensi dini.

Terdapat dua keterbatasan utama pada penelitian-penelitian sebelumnya, yaitu kurangnya optimasi hiperparameter freezing rate dalam proses transfer learning untuk mengendalikan adaptasi arsitektur pre-training, serta masih terbatasnya pembahasan mengenai dataset citra makanan berskala kecil yang rentan terhadap overfitting. Untuk mengatasi kedua keterbatasan tersebut dengan memanfaatkan Genetic Algorithm (GA) untuk mengoptimalkan secara simultan hiperparameter learning rate, dropout rate, dan freezing rate pada arsitektur EfficientNet-B0. P

Pendekatan yang diusulkan diterapkan pada klasifikasi citra makanan ringan (snack food), yang merupakan kombinasi baru dalam literatur untuk selanjutnya GA digunakan untuk mengoptimalkan hiperparameter Convolutional Neural Network (CNN) dengan arsitektur EfficientNet-B0 pada tugas klasifikasi citra makanan ringan menggunakan dataset yang terdiri atas 250 citra makanan ringan yang terbagi ke dalam 10 kelas yang diilustrasikan pada Gambar 1.



Gambar 1. Gambaran Umum Sistem

METODE PENELITIAN

Studi Pustaka

Studi pustaka dilakukan untuk mengkaji beberapa disiplin ilmu terkait. Literatur ini mencakup optimasi hyperparameter, *Genetic Algorithm* (GA), implementasi *search algorithm* untuk optimasi, pengujian hasil optimasi hyperparameter, bahasa pemrograman dan pustaka Python. Beberapa studi pustaka terkait dengan penelitian adalah sebagai berikut:

1) *Convolutional Neural Network and Peran Hyperparameter*

Convolutional Neural Network (CNN) telah menjadi arsitektur utama untuk tugas pengolahan citra karena kemampuannya mengekstrak fitur spasial secara otomatis. Kinerja akhir CNN sangat dipengaruhi oleh konfigurasi hyperparameter seperti learning rate, batch size, jumlah epoch, arsitektur lapisan, dan pilihan optimizer; pengaturan yang kurang tepat dapat mengakibatkan konvergensi buruk atau overfitting. Kajian komprehensif tentang metode dan tantangan optimasi hyperparameter menyatakan bahwa pemilihan hyperparameter adalah komponen krusial dalam desain model deep learning modern (Bischl et al., 2023).

2) *Metode Konvensional untuk Optimasi Hyperparameter*

Metode konvensional, misalnya *Grid Search* dan *Random Search* masih banyak digunakan karena kesederhanaannya, tetapi keduanya memiliki keterbatasan: *Grid Search* tidak dapat diskalakan dalam ruang parameter berdimensi tinggi, sedangkan *Random Search* mungkin tidak efisien dalam menemukan solusi terbaik di ruang yang kompleks (Hasan, 2024). Oleh karena itu, pendekatan otomatis dan heuristik telah menjadi alternatif yang sering diteliti dalam literatur terkini (Bischl et al., 2023).

3) *Genetic Algorithm (GA) untuk Optimasi Hyperparameter*

GA merupakan metode optimasi berbasis populasi yang mengadopsi mekanisme evolusi biologis yaitu selection, crossover, dan mutation untuk mengeksplorasi ruang solusi. Dalam konteks Hyperparameter Optimization (HPO), setiap individu merepresentasikan sekumpulan hyperparameter, sedangkan fitness umumnya diukur menggunakan metrik kinerja model (misalnya, validation accuracy) (Mehdary et al., 2024). GA unggul dalam melakukan pencarian global pada ruang berdimensi tinggi dan telah menunjukkan kemampuan untuk menemukan kombinasi hyperparameter yang kompetitif tanpa harus menguji seluruh kemungkinan kombinasi (Han, Ji-Hoon; Choi, Dong-Jin; Park, Sang-Uk; Hong, 2020).

GA juga memiliki beberapa keterbatasan yang memengaruhi kinerjanya, yaitu: (1) biaya komputasi yang tinggi apabila evaluasi fitness memerlukan proses pelatihan model secara penuh; (2) desain chromosome dan operator GA (representation, mutation rate, dan crossover scheme) sangat memengaruhi hasil sehingga memerlukan proses tuning tambahan; serta (3) perbandingan langsung antara Genetic Algorithm (GA) dengan metode HPO modern (misalnya, Optuna, BOHB, atau kombinasi Bayesian + bandit) pada domain-specific dataset seperti penyakit padi dengan protokol pengujian yang konsisten masih relatif terbatas dalam literatur. Oleh karena itu, penelitian yang mengevaluasi GA menggunakan protokol eksperimen yang terstandarisasi (misalnya, cross-validation, ablation study terhadap operator GA, dan analisis biaya komputasi) masih tetap relevan (Vincent dan Jidesh, 2023).

a. *Dataset dan Preprocessing*

Dataset yang digunakan berjumlah 250 citra makanan selingan yang dibagi ke dalam 10 kelas, dengan 25 citra pada masing-masing kelas. 80% data digunakan untuk training data dan 20% untuk validasi. *Preprocessing* data citra untuk input pada EfficientNetB0 dengan mengubah ukuran menjadi 224 x 224 piksel. Normalisasi piksel pada citra input dilakukan pada rentang 0-1. Untuk menghindari *overfitting* dan meningkatkan generalisasi model maka dilakukan augmentasi data berupa rotasi maksimum 20°, zoom 20%, horizontal flip, dan pergeseran lebar serta tinggi sebesar 10%.

b. *Arsitektur Model*

EfficientNetB0 pretrained dipilih sebagai model dasar pada penelitian ini. GlobalAveragePooling2D, Dropout, dan Dense 10 neuron dengan fungsi aktivasi softmax digunakan sebagai pengganti dari lapisan klasifikasi asli. Proses transfer learning dilakukan dengan membekukan sebagian lapisan EfficientNetB0 berdasarkan nilai *freeze ratio* yang dihasilkan oleh *Genetic Algorithm*.

c. Optimasi Hyperparameter dengan *Genetic Algorithm*

Rentang optimasi hyperparameter adalah sebagai berikut: *learning rate* antara 0,0001-0,001; *dropout rate* antara 0,20-0,50; *freeze ratio* antara 0,30-0,90. Setiap individu pada populasi merepresentasikan satu kombinasi hyperparameter. Struktur GA untuk optimasi terdiri dari ukuran populasi 10 individu, jumlah generasi 15, crossover rate 0,8, mutation rate 0,1, dan seleksi menggunakan tournament selection. Nilai fitness ditentukan berdasarkan validation accuracy model setelah pelatihan singkat selama 10 epoch pada setiap individu.

d. Pelatihan Model

Model terbaik dari hasil optimasi GA dilatih kembali menggunakan *optimizer* Adam dengan *learning rate* optimal. Pelatihan dilakukan selama maksimum 40 epoch dengan batch size 32. Untuk mencegah overfitting digunakan callback EarlyStopping dengan patience 5 dan ReduceLROnPlateau dengan faktor 0,5 dan *patience*

e. Pengujian

Pengujian dilakukan untuk melihat hasil dari kinerja algoritma optimasi dalam menghasilkan hyperparameter yang optimal. Untuk pengujian kinerja algoritma digunakan parameter akurasi (sebagai perbandingan antara jumlah prediksi yang benar dengan jumlah total data prediksi), presisi (merupakan rasio antara jumlah prediksi benar terhadap suatu label dengan jumlah total prediksi yang benar maupun yang salah terhadap label tersebut), dan recall (bertujuan untuk mengukur seberapa baik model dalam menghindari false negative, didefinisikan sebagai rasio *True Positive* terhadap *Actual Positive*).

HASIL DAN PEMBAHASAN

Optimasi hyperparameter pada model Convolutional Neural Network (CNN) berbasis EfficientNetB0 dilakukan menggunakan Genetic Algorithm (GA) dengan tiga parameter utama, yaitu *learning rate*, *dropout rate*, dan *freeze ratio*. Ketiga parameter tersebut berperan penting dalam menentukan kecepatan training, tingkat regulasi untuk mencegah overfitting, serta proporsi lapisan yang dibekukan selama proses fine-tuning.

Proses evolusi GA dilakukan melalui beberapa generasi dengan mekanisme seleksi individu terbaik, crossover, dan mutasi untuk menghasilkan kombinasi hyperparameter yang optimal. Dari hasil optimasi, diperoleh kombinasi nilai hyperparameter terbaik yang kemudian digunakan untuk membangun ulang model EfficientNetB0 dengan nilai *learning rate*=0,000670; *dropout rate*=0,34 dan *freeze ratio*=0,73.

```

=== Generasi 1/5 ===
Ind: [9.717225809985369e-05, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.7200
Ind: [1.6324715385649734e-05, 0.3215418096821734, 0.8591165432384094], Val_Acc=0.1200
Ind: [2.2516464687425857e-05, 0.27280037478106656, 0.8310377750671285], Val_Acc=0.3800
Ind: [0.00034011143688165307, 0.522404651613695, 0.7992078001191126], Val_Acc=0.9000
Ind: [1.1443654776132105e-05, 0.46574570012949384, 0.7018329657790454], Val_Acc=0.2000
Ind: [0.00031744178769530716, 0.5397316163012922, 0.7898628581855496], Val_Acc=0.8600

=== Generasi 2/5 ===
Ind: [0.00034011143688165307, 0.522404651613695, 0.7992078001191126], Val_Acc=0.8400
Ind: [0.00031744178769530716, 0.5397316163012922, 0.7898628581855496], Val_Acc=0.8400
Ind: [9.717225809985369e-05, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.7000
Ind: [0.00034011143688165307, 0.522404651613695, 0.7898628581855496], Val_Acc=0.8800
Ind: [0.00034011143688165307, 0.5627979599588262, 0.7898628581855496], Val_Acc=0.9000
Ind: [0.0006700325254889617, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9400

=== Generasi 3/5 ===
Ind: [0.0006700325254889617, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9200
Ind: [0.00034011143688165307, 0.5627979599588262, 0.7898628581855496], Val_Acc=0.9200
Ind: [0.00034011143688165307, 0.522404651613695, 0.7898628581855496], Val_Acc=0.9200
Ind: [0.00034011143688165307, 0.522404651613695, 0.7250718886443107], Val_Acc=0.9600
Ind: [0.00034011143688165307, 0.5627979599588262, 0.7898628581855496], Val_Acc=0.9000
Ind: [1.4382340264760242e-05, 0.522404651613695, 0.7898628581855496], Val_Acc=0.0800

=== Generasi 4/5 ===
Ind: [0.00034011143688165307, 0.522404651613695, 0.7250718886443107], Val_Acc=0.9400
Ind: [0.0006700325254889617, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9600
Ind: [0.00034011143688165307, 0.5627979599588262, 0.7898628581855496], Val_Acc=0.8600
Ind: [0.0006700325254889617, 0.522404651613695, 0.8801474104477216], Val_Acc=0.8800
Ind: [0.00034011143688165307, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9400
Ind: [0.0006700325254889617, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9000

=== Generasi 5/5 ===
Ind: [0.0006700325254889617, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9200
Ind: [0.00034011143688165307, 0.522404651613695, 0.7250718886443107], Val_Acc=0.9200
Ind: [0.00034011143688165307, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9400
Ind: [0.0006700325254889617, 0.3392499757465251, 0.7250718886443107], Val_Acc=0.9600
Ind: [5.9433698609519536e-05, 0.522404651613695, 0.7250718886443107], Val_Acc=0.5400
Ind: [0.00034011143688165307, 0.522404651613695, 0.7250718886443107], Val_Acc=0.8800

Best Hyperparams: [0.0006700325254889617, 0.3392499757465251, 0.7250718886443107]

```

Gambar 1. Pencarian Hyperparameter dengan GA

Model akhir dilatih menggunakan strategi early stopping dan learning rate reduction agar proses pelatihan lebih stabil serta mencegah overfitting. Hasil akhir menunjukkan bahwa kombinasi hyperparameter yang diperoleh GA mampu meningkatkan performa model secara signifikan dibandingkan konfigurasi awal tanpa optimasi.

Model CNN hasil optimasi diuji menggunakan dataset citra makanan selingan yang terdiri dari 10 kelas dengan total 250 citra. Proses pelatihan dilakukan hingga 40 epoch, dengan proporsi data 80% untuk training dan 20% untuk validation. Hasil pelatihan menunjukkan bahwa model mencapai akurasi validasi sebesar 92%, menandakan bahwa proses optimasi berhasil meningkatkan kinerja model.

Learning rate sebesar 0,000670 merupakan nilai yang cukup kecil sehingga dapat menjaga stabilitas proses fine-tuning pada model EfficientNetB0 yang sebelumnya telah dilatih, namun nilai tersebut masih cukup besar untuk dapat menyesuaikan bobot secara efektif terhadap karakteristik citra makanan selingan. Sedangkan jika learning rate terlalu tinggi atau terlalu kecil akan menyebabkan proses pelatihan tidak stabil, jika terlalu rendah akan mengakibatkan konvergensi lambat dan terjebak pada solusi lokal. Dropout rate sebesar 0,34 dapat mencegah overfitting tanpa menghilangkan informasi penting karena nilai tersebut mampu mengurangi ketergantungan antar neuron. Freeze ratio bernilai 0,73 mengartikan bahwa 73% lapisan awal EfficientNetB0 dipertahankan, untuk mempelajari fitur dasar citra seperti tepi, tekstur, dan bentuk yang bersifat umum. Kombinasi ketiga hyperparameter yang dihasilkan tersebut menyeimbangkan proses training, kemampuan generalisasi, dan adaptasi model terhadap domain data yang digunakan.

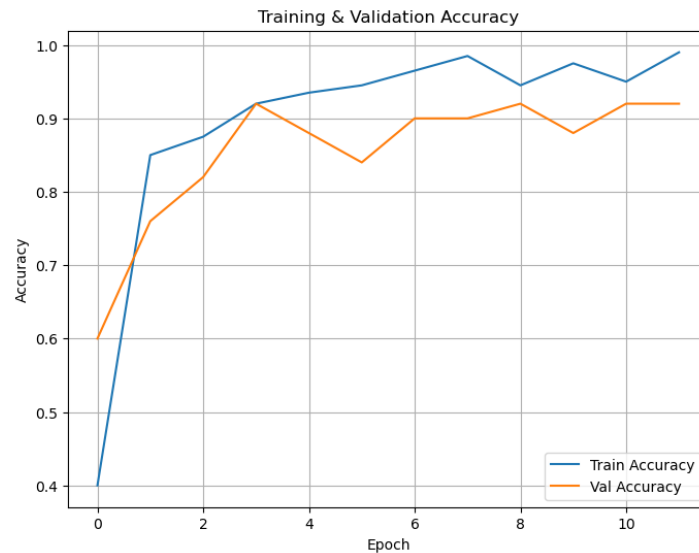
Penelitian ini memperoleh akurasi sebesar 92% pada dataset makanan selingan, hal ini sejalan dengan penelitian Jasril dan Sanjaya (2024) yang memperoleh akurasi sebesar 93% pada citra daging, penelitian ini menyatakan bahwa pengaturan hyperparameter dan strategi regularisasi pada EfficientNetB0 berpengaruh signifikan terhadap peningkatan akurasi klasifikasi citra.

```

Epoch 1/40
25/25 ----- 56s 1s/step - accuracy: 0.2600 - loss: 2.0803 - val_accuracy: 0.6000 - val_loss: 0.9295
- learning_rate: 6.7003e-04
Epoch 2/40
25/25 ----- 28s 1s/step - accuracy: 0.8524 - loss: 0.7297 - val_accuracy: 0.7600 - val_loss: 0.9117
- learning_rate: 6.7003e-04
Epoch 3/40
25/25 ----- 26s 1s/step - accuracy: 0.8448 - loss: 0.4750 - val_accuracy: 0.8200 - val_loss: 0.5633
- learning_rate: 6.7003e-04
Epoch 4/40
25/25 ----- 25s 1s/step - accuracy: 0.9494 - loss: 0.2690 - val_accuracy: 0.9200 - val_loss: 0.2974
- learning_rate: 6.7003e-04
Epoch 5/40
25/25 ----- 29s 1s/step - accuracy: 0.9410 - loss: 0.2229 - val_accuracy: 0.8800 - val_loss: 0.2787
- learning_rate: 6.7003e-04
Epoch 6/40
25/25 ----- 31s 1s/step - accuracy: 0.9108 - loss: 0.2169 - val_accuracy: 0.8400 - val_loss: 0.4854
- learning_rate: 6.7003e-04
Epoch 7/40
25/25 ----- 28s 1s/step - accuracy: 0.9571 - loss: 0.1637 - val_accuracy: 0.9000 - val_loss: 0.2102
- learning_rate: 6.7003e-04
Epoch 8/40
25/25 ----- 27s 1s/step - accuracy: 0.9838 - loss: 0.0889 - val_accuracy: 0.9000 - val_loss: 0.2630
- learning_rate: 6.7003e-04
Epoch 9/40
25/25 ----- 25s 998ms/step - accuracy: 0.9417 - loss: 0.1379 - val_accuracy: 0.9200 - val_loss: 0.33
00 - learning_rate: 6.7003e-04
Epoch 10/40
25/25 ----- 0s 780ms/step - accuracy: 0.9843 - loss: 0.0759
Epoch 10: ReduceLROnPlateau reducing learning rate to 0.0003350162587594241.
25/25 ----- 23s 911ms/step - accuracy: 0.9839 - loss: 0.0772 - val_accuracy: 0.8800 - val_loss: 0.27
65 - learning_rate: 6.7003e-04
Epoch 11/40
25/25 ----- 23s 907ms/step - accuracy: 0.9579 - loss: 0.1455 - val_accuracy: 0.9200 - val_loss: 0.21
99 - learning_rate: 3.3502e-04
Epoch 12/40
25/25 ----- 23s 903ms/step - accuracy: 0.9933 - loss: 0.0498 - val_accuracy: 0.9200 - val_loss: 0.18
99 - learning_rate: 3.3502e-04
  
```

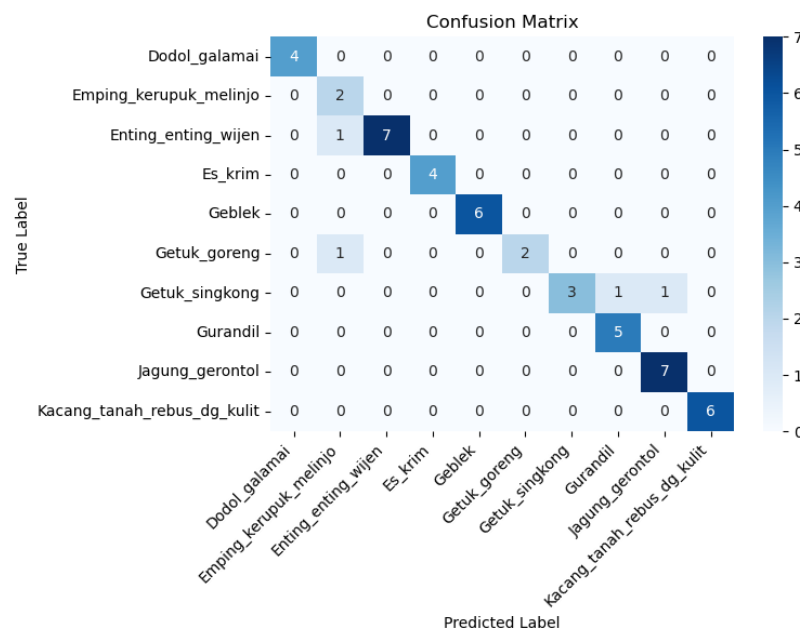
Gambar 2. Training Data

Grafik training dan validation accuracy pada Gambar 3 menunjukkan peningkatan pesat pada awal pelatihan (epoch 0–3) dan mencapai kestabilan setelah epoch ke-6. Nilai validation loss yang relatif rendah juga menunjukkan kemampuan generalisasi yang baik dari model terhadap data uji.



Gambar 3. Grafik Akurasi

Evaluasi lebih lanjut menggunakan confusion matrix memperlihatkan bahwa sebagian besar kelas citra makanan selingan berhasil diklasifikasikan dengan benar. Kelas seperti enting-enting wijen, jagung gerontol, dan kacang tanah rebus dengan kulit memiliki tingkat klasifikasi tertinggi. Hasil ini menunjukkan bahwa model yang telah dioptimasi menggunakan GA mampu mengenali pola citra dengan lebih akurat dan konsisten dibandingkan model tanpa optimasi. Hasil klasifikasi CNN dengan menggunakan hyperparameter optimal ditunjukkan oleh *Confusion Matrix* pada Gambar 4.



Gambar 4. Confusion Matrix

KESIMPULAN

Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa Genetic Algorithm (GA) efektif dalam meningkatkan kinerja model CNN dengan menemukan kombinasi hyperparameter yang optimal. Tingginya nilai akurasi validasi menunjukkan kemampuan GA dalam mencapai keseimbangan antara kecepatan konvergensi dan kemampuan generalisasi model. Selain itu, penerapan strategi early stopping dan learning rate scheduling terbukti efektif dalam mencegah overfitting tanpa mengurangi kinerja klasifikasi. Oleh karena itu, optimasi hyperparameter berbasis GA dapat dianggap sebagai pendekatan yang efisien dan adaptif untuk meningkatkan kinerja CNN, khususnya pada tugas klasifikasi citra dengan dataset yang terbatas, seperti

klasifikasi citra makanan ringan.

Penelitian selanjutnya dapat membandingkan kinerja GA dengan algoritma optimasi metaheuristik lainnya, seperti Particle Swarm Optimization (PSO) dan Differential Evolution (DE), untuk mengevaluasi efektivitas masing-masing metode dalam proses hyperparameter tuning pada CNN. Selain itu, studi komparatif yang lebih komprehensif antara GA, PSO, dan Bayesian Optimization untuk hyperparameter tuning pada EfficientNet-B0 dalam klasifikasi citra makanan ringan dapat memberikan pemahaman yang lebih mendalam mengenai keunggulan dan keterbatasan masing-masing metode optimasi.

REFERENSI

- Ansari, M.M. et al. (2024) "Evaluating CNN Architectures and Hyperparameter Tuning for Enhanced Lung Cancer Detection Using Transfer Learning," *Journal of Electrical and Computer Engineering*, 2024. Available at: <https://doi.org/10.1155/2024/3790617>.
- Bischl, B. et al. (2023) "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2), pp. 1–43. Available at: <https://doi.org/10.1002/widm.1484>.
- Chuhan Xu, P.C.-P.X.J. (2022) "Chuhan XU," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–14.
- Franceschi, L. et al. (2025) "Hyperparameter Optimization in Machine Learning." Available at: <https://doi.org/10.1561/22000000088>.
- Han, Ji-Hoon; Choi, Dong-Jin; Park, Sang-Uk; Hong, S.-K. search by orcid (2020) "Hyperparameter Optimization Using a Genetic Algorithm Considering Verification Time in a Convolutional Neural Network," *Journal of Electrical Engineering & Technology*, 15(2), pp. 721–726.
- Hasan, Md.T. (2024) "HyperQ-Opt: Q-learning for Hyperparameter Optimization." Available at: <https://arxiv.org/abs/2412.17765>.
- Jasril and Sanjaya, S. (2024) "Optimalisasi Convolutional Neural Network Menggunakan Augmentasi dan Hyperparameter untuk Klasifikasi Daging Sapi dan Daging Babi," *Jurnal Sistem dan Teknologi Informasi (Justin)*, 12(4), pp. 623–629. Available at: <https://doi.org/10.26418/justin.v12i4.80337>.
- Mehdary, A. et al. (2024) "Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection," *Sensors*, 24(4). Available at: <https://doi.org/10.3390/s24041230>.
- Santoso, F.Y., Sediyo, E. and Purnomo, H.D. (2024) "Optimalisasi Hyper Parameter Convolutional Neural Networks Menggunakan Ant Colony Optimization," *Jurnal Teknologi Informasi dan Ilmu Komputer*, 11(2), pp. 243–248. Available at: <https://doi.org/10.25126/jtiik.20241127105>.
- Vincent, A.M. and Jidesh, P. (2023) "An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms," *Scientific Reports*, 13(1), pp. 1–19. Available at: <https://doi.org/10.1038/s41598-023-32027-3>.
- Zhao, X. et al. (2024) "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, 57(4). Available at: <https://doi.org/10.1007/s10462-024-10721-6>.