

Klasifikasi Motif Batik Nusantara Menggunakan *EfficientNetB0* Berbasis Dekstop

¹Wahyu Setyo Budi, ²Patmi Kasih, ³Danang Wahyu Widodo

^{1,2,3}Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹bawangmerah110302, ²patmikasih@unpkdr.ac.id, ³danangwahyuwidodo@unpkediri.ac.id

Penulis Korespondens : Wahyu Setyo Budi

Abstrak—Identifikasi motif batik secara manual masih bergantung pada pengetahuan pengguna dan membutuhkan waktu, padahal kebutuhan pengenalan motif secara cepat diperlukan untuk edukasi budaya dan arsip digital. Penelitian ini membangun aplikasi klasifikasi motif batik nusantara menggunakan *EfficientNetB0* dengan pendekatan *transfer learning* dan implementasi *Streamlit*. Dataset terdiri atas 21 kelas motif batik dengan pembagian data latih dan data uji. Citra diproses melalui *resize*, *preprocessing EfficientNet*, dan augmentasi pada data latih. Model dilatih menggunakan *optimizer Adam*, *categorical crossentropy*, *early stopping*, *reduce learning rate*, dan *model checkpoint*. Hasil pengujian menunjukkan *Accuracy* 66,67 persen, *precision macro* 71,94 persen, *recall macro* 66,67 persen, dan *F1-score macro* 66,29 persen. Aplikasi dapat menerima gambar batik, kemudian menampilkan motif terdeteksi, *confidence*, deskripsi motif, serta top-5 prediksi. Hasil ini menunjukkan bahwa aplikasi mampu mendukung identifikasi awal motif batik secara praktis.

Kata Kunci—Aplikasi *Streamlit*, Batik Nusantara, *EfficientNetB0*, Klasifikasi Citra, *Transfer Learning*

Abstract—Manual batik motif identification still depends on user knowledge and requires considerable time, while fast motif recognition is needed for cultural education and digital archiving. This study developed a Nusantara batik motif classification application using *EfficientNetB0* with transfer learning and *Streamlit* implementation. The dataset consisted of 21 batik motif classes divided into training and testing data. Images were processed through resizing, *EfficientNet preprocessing*, and augmentation on training data. The model was trained using *Adam optimizer*, *categorical crossentropy*, *early stopping*, *reduce learning rate*, and *model checkpoint*. Testing results showed an *Accuracy* of 66.67 percent, *macro precision* of 71.94 percent, *macro recall* of 66.67 percent, and *macro F1-score* of 66.29 percent. The application can receive batik images and display the detected motif, *confidence score*, motif description, and top-5 predictions. These results indicate that the application can support practical initial identification of batik motifs.

Keywords—Batik Classification, *EfficientNetB0*, Image Classification, *Streamlit Application*, *Transfer Learning*

This is an open access article under the CC BY-SA License.



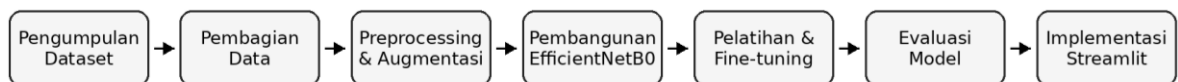
I. PENDAHULUAN

Batik merupakan salah satu warisan budaya Indonesia yang memiliki nilai historis, estetika, dan identitas daerah. Setiap motif batik memiliki karakteristik visual yang berbeda, baik dari segi pola, ornamen, warna, maupun tekstur, sehingga menjadikan batik sebagai objek yang menarik untuk dikaji menggunakan teknologi pengolahan citra digital. Dalam praktiknya, identifikasi motif batik masih sering dilakukan secara manual berdasarkan pengetahuan pengguna, yang dapat menyebabkan proses pengenalan memerlukan waktu lebih lama dan menghasilkan penilaian yang kurang konsisten [1], [2]. Oleh karena itu, diperlukan suatu sistem yang mampu mengenali motif batik secara otomatis melalui citra yang diunggah pengguna. Penelitian ini bertujuan mengembangkan aplikasi klasifikasi motif batik nusantara berbasis *Streamlit* yang dapat menampilkan hasil prediksi berupa nama motif, tingkat kepercayaan, deskripsi motif, dan lima prediksi tertinggi.

Salah satu metode yang banyak digunakan dalam klasifikasi citra adalah *Convolutional Neural Network* (CNN) karena mampu mengekstraksi dan mempelajari fitur visual secara otomatis melalui proses konvolusi [3], [4], [5]. Berbagai penelitian menunjukkan bahwa CNN dan turunannya memiliki performa yang baik dalam pengenalan objek dan klasifikasi citra [6], [7]. Pada penelitian ini digunakan arsitektur *EfficientNetB0* dengan pendekatan transfer learning untuk meningkatkan efektivitas pembelajaran pada dataset batik. Berbeda dengan penelitian sebelumnya yang umumnya berfokus pada evaluasi model klasifikasi [1], [2], penelitian ini mengintegrasikan model *EfficientNetB0* ke dalam aplikasi *Streamlit* sehingga dapat digunakan secara langsung oleh pengguna. Kinerja sistem dievaluasi menggunakan metrik *Accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix* untuk mengetahui tingkat keberhasilan model dalam mengenali motif batik secara akurat [8], [9], [10].

II. METODE

Penelitian ini menggunakan metode *Convolutional Neural Network* dengan arsitektur *EfficientNetB0* dan pendekatan *transfer learning*. Tahapan penelitian disesuaikan dengan proses yang terdapat pada aplikasi, mulai dari pengumpulan dataset, pembagian data, *preprocessing*, augmentasi, pembangunan model, pelatihan, evaluasi, hingga implementasi ke dalam aplikasi *Streamlit*.



Gambar 1. Alur penelitian

A. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini adalah dataset citra motif batik nusantara yang disusun dalam folder berdasarkan nama kelas motif. Berdasarkan struktur dataset pada aplikasi, terdapat 21 kelas motif batik, yaitu Aceh Pintu Aceh, Bali Barong, Bali Merak, DKI Ondel Ondel, Jawa Barat Megamendung, Jawa Timur Pring, Kalimantan Dayak, Kediri, Lampung Gajah, Madura Mataketeran, Maluku Pala, NTB Lumbung, Papua Asmat, Papua Cendrawasih, Papua Tifa, Solo Parang, Sulawesi Selatan Lontara, Sumatera Barat Rumah Minang, Sumatera Utara Boraspati, Yogyakarta Kawung, dan Yogyakarta Parang. Setiap kelas memiliki 32 citra, sehingga total dataset yang digunakan berjumlah 672 citra.

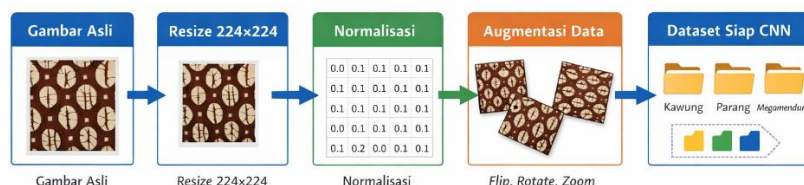
B. Pembagian Data

Pembagian data dilakukan secara bertahap. Dataset utama dibagi menjadi data latih dan data uji dengan proporsi 80:20. Hasil pembagian tersebut menghasilkan 525 citra pada folder train dan 147 citra pada folder test. Selanjutnya, data pada folder train dibagi kembali menjadi data pelatihan dan data validasi dengan proporsi 80:20 menggunakan stratifikasi label. Dengan demikian, model dilatih menggunakan 420 citra, divalidasi menggunakan 105 citra, dan diuji menggunakan 147 citra yang tidak digunakan saat proses pelatihan.

C. *Preprocessing* dan Augmentasi Data

Tahapan *preprocessing* dilakukan agar seluruh citra memiliki format input yang sesuai dengan *EfficientNetB0*. Setiap citra diubah menjadi format RGB dan di-resize ke ukuran 224 x 224 piksel.

Pada proses pelatihan, citra diproses menggunakan fungsi *preprocessing EfficientNet* agar skala nilai piksel sesuai dengan kebutuhan arsitektur model. *Preprocessing* ini penting karena perbedaan ukuran dan format citra dapat memengaruhi stabilitas proses pelatihan maupun akurasi prediksi aplikasi.



Gambar 2. *Preprocessing* dan Augmentasi Data

Augmentasi data seperti gambar 2 diterapkan pada data pelatihan untuk menambah variasi citra dan mengurangi risiko *overfitting*. Teknik augmentasi yang digunakan meliputi *rotation range* sebesar 15 derajat, *zoom range* 0,2, *width shift* dan *height shift* sebesar 0,1, *shear range* sebesar 0,1, horizontal *flip*, serta *brightness* range 0,8 sampai 1,2. Teknik tersebut membuat model belajar mengenali motif pada variasi posisi, skala, kemiringan, dan pencahayaan yang berbeda.

D. Pembangunan Model *EfficientNetB0*

Model yang digunakan adalah *EfficientNetB0* dengan bobot awal *ImageNet* dan *include top* bernilai false. Bagian *EfficientNetB0* pada gambar 3 berfungsi sebagai *feature extractor*, kemudian ditambahkan *Global Average Pooling 2D*, *Dropout* sebesar 0,5, *Dense layer* berisi 256 neuron dengan aktivasi ReLU, dan *Dense output* dengan aktivasi *softmax* sesuai 21 kelas motif batik. Model dikompilasi menggunakan optimizer Adam dengan learning rate 0,0001, *loss categorical crossentropy*, dan metrik *Accuracy*.

Pelatihan dilakukan dalam dua tahap. Tahap pertama melatih bagian *classifier* dengan base model dalam keadaan frozen selama maksimal 50 epoch. Tahap kedua melakukan *fine-tuning* dengan membuka sebagian layer akhir *EfficientNetB0* dan melatih ulang model selama maksimal 15 epoch. Proses pelatihan menggunakan *Early Stopping*, *Reduce Learning Rate on Plateau*, dan Model *Checkpoint* agar model terbaik tersimpan secara otomatis dalam file `best_model_EfficientNetB0.keras`.



Gambar 3. Arsitektur *EfficientNetB0*

E. Implementasi Aplikasi *Streamlit*

Aplikasi dibangun menggunakan *Streamlit* dengan judul Klasifikasi Motif Batik. Aplikasi menyediakan fitur upload gambar berformat JPG, JPEG, PNG, dan WEBP. Setelah gambar diunggah, aplikasi menampilkan pratinjau citra, ukuran lebar, tinggi, serta ukuran file. Citra kemudian dikonversi menjadi RGB, diubah ke ukuran 224 x 224 piksel, diproses menggunakan *preprocessing EfficientNet*, dan dimasukkan ke model *EfficientNetB0* untuk memperoleh probabilitas prediksi pada setiap kelas.

Output aplikasi berupa nama motif dengan probabilitas tertinggi, nilai confidence, kategori tingkat kepercayaan, waktu proses prediksi, deskripsi singkat motif, serta grafik top-5 prediksi. Informasi tersebut dibuat agar pengguna tidak hanya mengetahui satu kelas hasil prediksi, tetapi juga dapat melihat kemungkinan kelas lain yang memiliki kemiripan visual.

F. Evaluasi Model

Evaluasi dilakukan menggunakan data uji yang tidak digunakan dalam proses pelatihan dan validasi. Parameter evaluasi yang digunakan meliputi *Accuracy*, *precision*, *recall*, *F1-score*, dan confusion matrix. *Accuracy* digunakan untuk mengukur ketepatan prediksi secara keseluruhan. *Precision* digunakan untuk melihat ketepatan prediksi pada setiap kelas, *recall* digunakan untuk mengetahui kemampuan model menemukan data dari kelas yang benar, sedangkan *F1-score* digunakan untuk melihat keseimbangan antara *precision* dan *recall*. *Confusion matrix* digunakan untuk menganalisis kelas yang masih sering tertukar.

Rumus evaluasi yang digunakan adalah *Accuracy* sama dengan jumlah prediksi benar dibagi jumlah seluruh data, *Precision* sama dengan True Positive dibagi jumlah True Positive dan False Positive, *Recall* sama dengan True Positive dibagi jumlah True Positive dan False Negative, sedangkan *F1-score* merupakan rata-rata harmonis dari *Precision* dan *Recall*.

III. HASIL DAN PEMBAHASAN

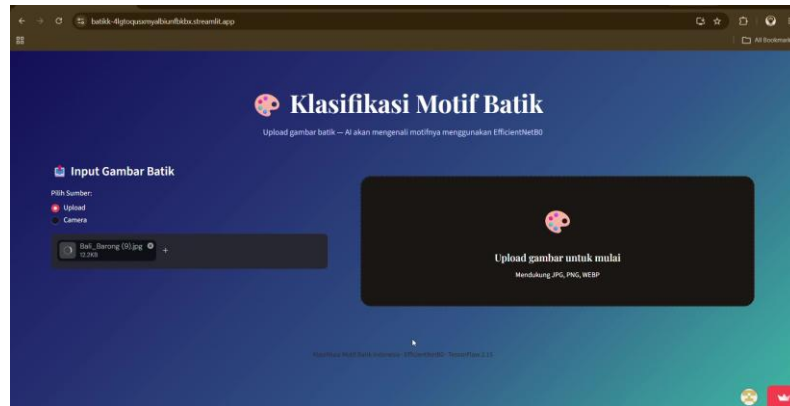
Hasil penelitian disusun berdasarkan proses yang telah diterapkan pada aplikasi. Dataset yang digunakan seperti pada tabel 1 berjumlah 672 citra dari 21 kelas motif batik. Setiap kelas memiliki 32 citra yang kemudian dibagi menjadi 25 citra untuk data latih awal dan 7 citra untuk data uji seperti pada tabel 1. Data latih awal selanjutnya dibagi menjadi data pelatihan dan data validasi agar proses training dapat dipantau sebelum dilakukan pengujian akhir.

Tabel 1. Pembagian Dataset

Bagian Data	Jumlah Citra	Jumlah Kelas	Keterangan
Data pelatihan	420	21	Digunakan untuk pembaruan bobot model.
Data validasi	105	21	Digunakan untuk memantau performa saat training.
Data uji	147	21	Digunakan untuk evaluasi akhir model.
Total	672	21	Seluruh citra motif batik nusantara.

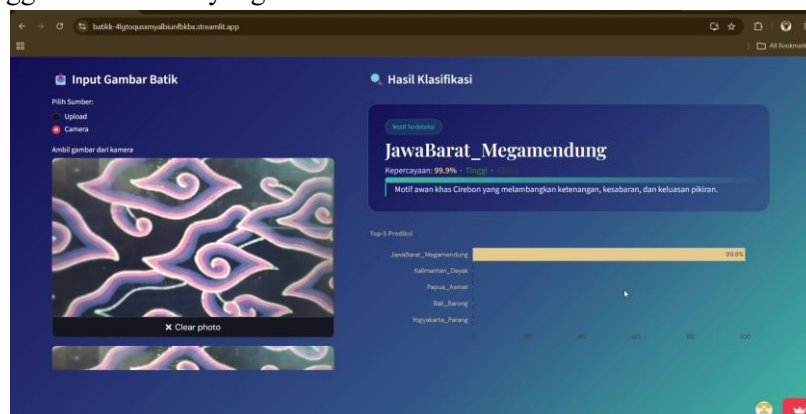
Proses *preprocessing* berhasil menyeragamkan ukuran citra menjadi 224 x 224 piksel dan mengubah citra ke format RGB. Pada tahap pelatihan, augmentasi menghasilkan variasi citra dari

sisi rotasi, pergeseran, *zoom*, *shear*, pencahayaan, dan pembalikan horizontal. Tahapan ini sesuai dengan kebutuhan aplikasi karena gambar yang diunggah pengguna dapat memiliki sudut pengambilan, pencahayaan, dan ukuran yang berbeda-beda.



Gambar 4. Tampilan awal aplikasi saat pengguna mengunggah gambar batik

Gambar 4 menunjukkan tampilan awal aplikasi klasifikasi motif batik. Pada sisi kiri aplikasi terdapat fitur input gambar batik melalui file uploader, sedangkan sisi kanan menampilkan area hasil klasifikasi. Tampilan ini menunjukkan bahwa aplikasi dirancang untuk alur penggunaan sederhana, yaitu pengguna memilih gambar batik, aplikasi membaca citra, kemudian sistem melakukan proses klasifikasi menggunakan model yang telah dilatih.



Gambar 5. Tampilan hasil klasifikasi motif batik

Gambar 5 menunjukkan hasil klasifikasi pada aplikasi. Setelah citra diunggah, aplikasi menampilkan nama motif terdeteksi, nilai confidence, deskripsi motif, dan grafik top-5 prediksi. Pada contoh pengujian, aplikasi mampu menampilkan prediksi motif beserta persentase kemungkinan pada beberapa kelas tertinggi. Penyajian top-5 prediksi membantu pengguna memahami kelas alternatif yang secara visual masih memiliki kemiripan dengan gambar input.

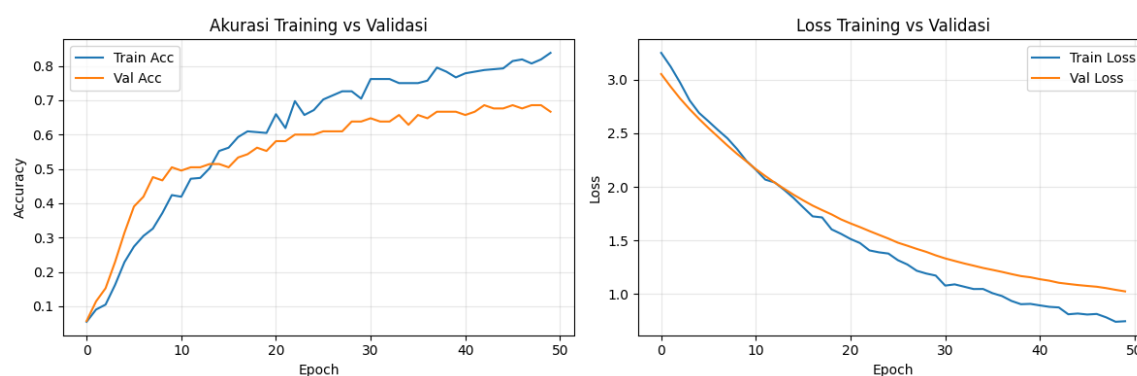
Tabel 2. Hasil Evaluasi Model *EfficientNetB0*

Metrik	Nilai	Keterangan
--------	-------	------------

<i>Accuracy</i>	0,6667	Model memprediksi benar 66,67 persen data uji.
Macro <i>Precision</i>	0,7194	Rata-rata ketepatan prediksi seluruh kelas.
Macro <i>Recall</i>	0,6667	Rata-rata kemampuan model mengenali setiap kelas.
Macro <i>F1-score</i>	0,6629	Keseimbangan <i>precision</i> dan <i>recall</i> .
Support	147	Jumlah citra pada data uji.

Berdasarkan Tabel 2, model *EfficientNetB0* memperoleh *Accuracy* sebesar 66,67 persen pada data uji. Nilai macro *precision* sebesar 71,94 persen menunjukkan bahwa prediksi model pada beberapa kelas cukup tepat, tetapi nilai macro *recall* dan macro *F1-score* yang berada pada kisaran 66 persen menunjukkan bahwa masih terdapat beberapa kelas yang belum dikenali secara stabil. Hasil ini menunjukkan bahwa model sudah dapat digunakan sebagai identifikasi awal, namun masih membutuhkan peningkatan data dan penyempurnaan pelatihan agar hasil klasifikasi menjadi lebih kuat.

Hasil classification report menunjukkan bahwa kelas Yogyakarta Kawung memperoleh *F1-score* 1,0000, Jawa Barat Megamendung memperoleh *F1-score* 0,9231, sedangkan Kalimantan Dayak dan Madura Mataketeran memperoleh *F1-score* 0,8571. Kelas-kelas tersebut cenderung memiliki ciri visual yang lebih khas sehingga lebih mudah dikenali oleh model. Sebaliknya, kelas NTB Lumbung memperoleh *F1-score* 0,0000, Sulawesi Selatan Lontara memperoleh *F1-score* 0,2353, dan Solo Parang memperoleh *F1-score* 0,4444. Rendahnya performa pada beberapa kelas dapat terjadi karena jumlah data masih terbatas, variasi motif dalam satu kelas belum cukup banyak, serta adanya kemiripan pola dengan kelas lain.



Gambar 6. Grafik akurasi dan loss training serta validasi model *EfficientNetB0*

Gambar 6 menunjukkan perubahan nilai akurasi dan loss pada data training serta validasi selama proses pelatihan model *EfficientNetB0*. Pada grafik akurasi, nilai training *Accuracy* meningkat secara bertahap dari epoch awal hingga mencapai sekitar 0,83 pada akhir pelatihan. Sementara itu, validation *Accuracy* juga mengalami peningkatan dan cenderung stabil pada kisaran 0,67 sampai 0,69. Kondisi tersebut menunjukkan bahwa model mampu mempelajari pola visual motif batik, meskipun masih terdapat selisih antara performa pada data training dan data validasi. Pada grafik loss, nilai *training loss* dan *validation loss* sama-sama mengalami penurunan dari awal hingga akhir pelatihan. *Training loss* turun hingga sekitar 0,75, sedangkan *validation loss* turun

hingga sekitar 1,02. Penurunan loss validasi menunjukkan bahwa proses pembelajaran berjalan stabil dan model tidak mengalami kegagalan konvergensi. Namun, jarak antara kurva training dan validasi mengindikasikan bahwa model masih berpotensi mengalami *overfitting* ringan. Oleh karena itu, peningkatan jumlah data, augmentasi yang lebih bervariasi, dan proses fine-tuning lanjutan dapat dilakukan untuk meningkatkan kemampuan generalisasi model terhadap citra batik baru.

Secara fungsional, aplikasi telah mampu menjalankan alur utama sistem, yaitu menerima input gambar, menampilkan pratinjau citra, melakukan *preprocessing*, memanggil model *best_model_EfficientNetB0*.keras, menghasilkan prediksi, dan menampilkan top-5 kelas. Seperti pada Tabel 3 hasil penelitian telah sesuai dengan tujuan pengembangan aplikasi, yaitu menyediakan sistem klasifikasi motif batik yang dapat digunakan secara langsung melalui antarmuka web sederhana.

Tabel 3. Pengujian Fungsional Aplikasi

No	Fitur	Skenario Pengujian	Hasil
1	Upload citra	Pengguna mengunggah gambar batik berformat JPG, JPEG, PNG, atau WEBP.	Berhasil
2	<i>Preprocessing</i>	Aplikasi mengubah citra ke RGB dan ukuran 224 x 224 piksel.	Berhasil
3	Prediksi motif	Model <i>EfficientNetB0</i> menghasilkan probabilitas untuk setiap kelas.	Berhasil
4	Hasil klasifikasi	Aplikasi menampilkan nama motif, confidence, deskripsi, dan top-5 prediksi.	Berhasil
5	Informasi citra	Aplikasi menampilkan lebar, tinggi, dan ukuran file gambar.	Berhasil

IV. KESIMPULAN

Penelitian ini menghasilkan aplikasi klasifikasi motif batik nusantara menggunakan *EfficientNetB0* berbasis *Streamlit*. Aplikasi telah disesuaikan dengan alur kerja sistem, yaitu menerima citra batik, melakukan *preprocessing*, memproses prediksi menggunakan model *EfficientNetB0*, dan menampilkan hasil berupa motif terdeteksi, *confidence*, deskripsi motif, serta top-5 prediksi. Berdasarkan pengujian pada 147 citra uji, model memperoleh *Accuracy* sebesar 66,67 persen, macro *precision* sebesar 71,94 persen, macro *recall* sebesar 66,67 persen, dan macro *F1-score* sebesar 66,29 persen. Hasil tersebut menunjukkan bahwa aplikasi dapat digunakan sebagai alat bantu identifikasi awal motif batik, tetapi peningkatan jumlah dataset, keseimbangan variasi antar kelas, dan *fine-tuning* lanjutan masih diperlukan untuk meningkatkan performa klasifikasi pada kelas yang memiliki pola visual mirip.

DAFTAR PUSTAKA

- [1] A. Muriagista and D. Kurniadi, "IMPLEMENTASI ARSITEKTUR RESNET50 PADA KLASIFIKASI MOTIF BATIK INDONESIA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)," 2025.
- [2] S. B. Santosa, A. R. Chrismanto, and B. Susanto, "Classification of Batik Keraton motifs using ResNet-50 fine-tuning architecture," *AITI*, vol. 23, no. 1, pp. 46–60, Feb. 2026, doi: 10.24246/aiti.v23i1.46-60.

- [3] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton, "NCBI Literature Resources." Accessed: Dec. 30, 2025. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/26017442/>
- [4] N. H. Harani, C. Prianto, and M. Hasanah, "Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python," 2019.
- [5] R. Pajes, *Detection of Corn Leaf Diseases using Convolutional Neural Network with OpenMP Implementation*. 2020. doi: 10.1109/HNICEM51456.2020.9400004.
- [6] D. Dhelviana, T. Amelia, J. Sulaksono, and D. W. Widodo, "Sistem Pendeteksi Kekerasan Berbasis Cnn (Convolutional Neural Network)," 2023.
- [7] L. Dwi Anggraini, U. Mahdiyah, and R. Wulanningrum, "Pengujian Simple CNN Dengan Arsitektur LeNet5 Pada Penyakit Daun Jagung," Online, 2025.
- [8] P. Arsitektur *et al.*, "Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi) 1626," Online, 2024.
- [9] M. R. Andrianto, F. Firdaus, E. Pratama, R. S. Purnama, and E. Y. Puspaningrum, "Klasifikasi Motif Batik Tradisional Mataraman Menggunakan CNN," 2025, [Online]. Available: <https://www.kaggle.com/datasets/dionisiusdh/indonesian->
- [10] M. M. A. Wona *et al.*, "Klasifikasi Batik Indonesia Menggunakan Convolutional Neural Network (CNN)," *Jurnal Rekayasa Teknologi Informasi (JURTI)*, vol. 7, no. 2, pp. 172–179, 2023, Accessed: Dec. 30, 2025. [Online]. Available: <https://e-journals.unmul.ac.id/index.php/INF/article/view/13694>