

Deteksi Karakter Aksara Jawa Menggunakan YOLO11 Pendekatan Deep Learning untuk Pelestarian Warisan Budaya Digital

Eko Rahmad Darmawan, Dhani Ariatmanto²

^{1,2} Program Studi Magister Teknik Informatika Universitas AMIKOM Yogyakarta Indonesia
Jln. Ring Road Utara, Ngiringin, Condongcatur, Depok, Sleman, Yogyakarta,
Indonesia

¹eko.rahmad@students.amikom.ac.id

²dhaniari@amikom.ac.id

Intisari— Aksara Jawa merupakan warisan budaya Nusantara yang menghadapi ancaman kepunahan akibat dominasi alfabet Latin dan minimnya pengguna yang mampu menulis dengan aksara tradisional ini. Penelitian ini bertujuan mengembangkan sistem deteksi karakter Aksara Jawa menggunakan algoritma You Only Look Once versi 11 (YOLO11) untuk mendukung upaya pelestarian budaya melalui digitalisasi yang efisien. Metode penelitian menggunakan pendekatan eksperimental dengan deep learning, dimana dataset Aksara Jawa yang terdiri dari 20 karakter dasar plus kelas background diperoleh dari Kaggle dan dipreproses menggunakan Roboflow dengan teknik augmentasi data. Model YOLO11 diimplementasikan dengan optimizer SGD, image size 640px, dan dilatih selama 500 epoch untuk mencapai konvergensi optimal. Arsitektur YOLO11 mengintegrasikan komponen advanced seperti C3K2 blocks, Spatial Pyramid Pooling-Fast (SPPF), dan Cross-scale Pixel Spatial Attention (C2PSA) untuk meningkatkan kemampuan ekstraksi fitur multiskala. Evaluasi kinerja model menggunakan confusion matrix dengan metrik akurasi, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa model YOLO11 mencapai akurasi keseluruhan 81.00% dengan macro-averaged precision 86.28%, macro-averaged recall 87.25%, dan macro-averaged F1-score 86.41%. Distribusi kinerja model menunjukkan 7 kelas dengan performa tinggi (F1-score $\geq 90\%$), 9 kelas dengan kinerja sedang (80-90%), dan 4 kelas dengan kinerja rendah (<80%). Kelas "nga" mencapai performa sempurna 100%, sementara kelas "ha" menunjukkan kinerja terendah dengan F1-score 68.09%. Penelitian ini berhasil meningkatkan akurasi dibandingkan metode sebelumnya yang menggunakan backpropagation neural network (74%) dan backpropagation konvensional (59.5%), meskipun masih menghadapi tantangan dalam mendeteksi karakter yang memiliki bentuk mirip dan penanganan kelas background. Kontribusi utama penelitian adalah implementasi pertama YOLO11 untuk deteksi Aksara Jawa yang membuka peluang pengembangan sistem digitalisasi literatur kuno yang lebih efisien dan akurat.

Kata kunci— Aksara Jawa, YOLO11, Deteksi Karakter, Deep Learning, Computer Vision.

Abstract— Javanese script represents a significant cultural heritage of the Indonesian archipelago that faces extinction threats due to Latin alphabet dominance and minimal users capable of writing with this traditional script. This research aims to develop a Javanese character detection system using You Only Look Once version 11 (YOLO11) algorithm to support cultural preservation efforts through efficient digitalization. The research methodology employs an experimental approach with deep learning, where the Javanese script dataset consisting of 20 basic characters plus background class was obtained from Kaggle and preprocessed using Roboflow with data augmentation techniques. The YOLO11 model was implemented with SGD optimizer, 640px image size, and trained for 500 epochs to achieve optimal convergence. YOLO11 architecture integrates advanced components such as C3K2 blocks, Spatial Pyramid Pooling-Fast (SPPF), and Cross-scale Pixel Spatial Attention (C2PSA) to enhance multiscale feature extraction capabilities. Model performance evaluation utilized confusion matrix with accuracy, precision, recall, and F1-score metrics. Research results demonstrate that the YOLO11 model achieved an overall accuracy of 81.00% with macro-averaged precision of 86.28%, macro-averaged recall of 87.25%, and macro-averaged F1-score of 86.41%. Model performance distribution shows 7 classes with high performance (F1-score $\geq 90\%$), 9 classes with medium performance (80-90%), and 4 classes with low performance (<80%). The "nga" class achieved perfect performance of 100%, while the "ha" class showed the lowest performance with an F1-score of 68.09%. This research successfully improved accuracy compared to previous methods using backpropagation neural networks (74%) and conventional backpropagation (59.5%), although challenges remain in detecting characters with similar shapes and handling background class. The main contribution is the first implementation of YOLO11 for Javanese script detection, opening opportunities for developing more efficient and accurate ancient literature digitalization systems.

Keywords— Javanese Script, YOLO11, Character Detection, Deep Learning, Computer Vision.

I. PENDAHULUAN

Aksara Jawa merupakan salah satu warisan budaya Nusantara yang memiliki nilai historis dan kultural yang sangat tinggi. Menurut VOA Indonesia, terdapat lebih dari 70 juta orang sebagai penutur bahasa Jawa, namun kurang dari 1%

yang dapat menulis dengan Aksara Jawa [1]. Seiring perkembangan zaman dan perluasan kehidupan global, aksara Jawa mulai ditinggalkan, bahkan oleh orang Jawa sendiri. Dominasi alfabet Latin membuat penggunaan Aksara Jawa semakin jarang dan dikhawatirkan akan semakin terpinggirkan.

Upaya pelestarian Aksara Jawa menjadi semakin penting dalam era digital. Teknologi computer vision dapat membantu dalam mendokumentasikan, mendeteksi, dan mengenali Aksara Jawa secara otomatis. Pengenalan aksara menggunakan computer vision penting dalam memperluas akses terhadap literatur kuno, yang selama ini sulit diakses karena keterbatasan orang yang mampu membaca Aksara Jawa. Digitalisasi Aksara Jawa tidak hanya membantu dalam pelestarian identitas budaya, tetapi juga meningkatkan kesadaran dan apresiasi masyarakat terhadap warisan budaya Jawa [2].

Berbagai penelitian telah dilakukan untuk mengatasi tantangan pengenalan Aksara Jawa menggunakan teknologi computer vision dan machine learning. Penelitian awal oleh [3] menggunakan backpropagation neural network mencapai akurasi 74% untuk transliterasi aksara Jawa ke Latin. [1] mengembangkan sistem menggunakan metode backpropagation dengan akurasi 59.5%, namun masih menghadapi tantangan dalam mendeteksi karakter yang memiliki bentuk mirip.

Perkembangan teknologi deep learning membuka peluang yang lebih baik. Penelitian oleh [4] menunjukkan bahwa pendekatan tradisional seperti CNN dan SVM belum cukup efektif dalam menangani variasi bentuk dan kualitas gambar aksara Jawa. [5] berhasil meningkatkan akurasi sebesar 15% dengan integrasi U-Net dalam mendeteksi aksara, dengan rata-rata deteksi meningkat 1.2%. [6] menggunakan YOLO dengan augmentasi data mencapai akurasi 96.4% dengan F1-score 92% pada 3021 citra aksara Hanacaraka.

Penelitian terbaru oleh [7] menggunakan YOLOv9 untuk deteksi karakter Aksara Jawa menunjukkan hasil yang menjanjikan dengan confidence 80% dan nilai mean Average Precision (mAP) sebesar 0.95. Namun, masih terdapat tantangan dalam mendeteksi karakter-karakter yang memiliki bentuk mirip, seperti "ba", "sa", dan "la". Penelitian internasional juga menunjukkan kemajuan signifikan dalam bidang historical document analysis. [8] mengembangkan sistem OCR untuk manuskrip Cina kuno dengan akurasi 94.2%.

Algoritma You Only Look Once (YOLO) telah terbukti efektif dalam deteksi objek real-time dengan akurasi tinggi. YOLO11, sebagai versi terbaru, menawarkan peningkatan signifikan dibandingkan pendahulunya. Menurut Deng et al. (2025), YOLO11 mengintegrasikan komponen seperti C3K2 blocks, Spatial Pyramid Pooling-Fast (SPPF), dan Cross-scale Pixel Spatial Attention (C2PSA) yang meningkatkan kemampuan ekstraksi fitur multiskala dan pengenalan objek kecil. [9] menunjukkan bahwa YOLO11 memiliki peningkatan pada nilai mAP50-95, kecepatan inferensi, serta stabilitas performa yang lebih baik dibandingkan YOLOv9 dan YOLOv10. Dalam aplikasi citra penginderaan jauh, YOLO11 mencapai presisi deteksi 88.61% dan F1-score 87.09%.

Meskipun berbagai penelitian telah dilakukan, masih terdapat gap penelitian yang perlu diisi. Pertama, belum ada penelitian yang menggunakan YOLO11 untuk deteksi Aksara Jawa, padahal versi ini menawarkan peningkatan performa yang signifikan. Kedua, penelitian sebelumnya masih menghadapi tantangan dalam membedakan karakter yang memiliki bentuk mirip. Ketiga, diperlukan sistem yang tidak

hanya akurat tetapi juga efisien untuk implementasi real-time dalam aplikasi digitalisasi literatur kuno.

Penelitian ini bertujuan untuk mengembangkan sistem deteksi karakter Aksara Jawa menggunakan algoritma YOLO11 yang diharapkan dapat mencapai akurasi lebih tinggi dibandingkan penelitian sebelumnya, khususnya dalam mengatasi tantangan deteksi karakter yang memiliki bentuk mirip. Dengan memanfaatkan kemampuan YOLO11 dalam deteksi objek real-time dan ekstraksi fitur otomatis, penelitian ini diharapkan dapat berkontribusi pada pelestarian budaya Jawa melalui digitalisasi yang efisien. Selain itu, penelitian ini akan memberikan kontribusi pada pengembangan teknologi pengenalan karakter berbasis deep learning dan membuka peluang untuk penelitian lebih lanjut dalam bidang preservasi warisan budaya digital.

II. STUDI PUSTAKA

A. Aksara Jawa

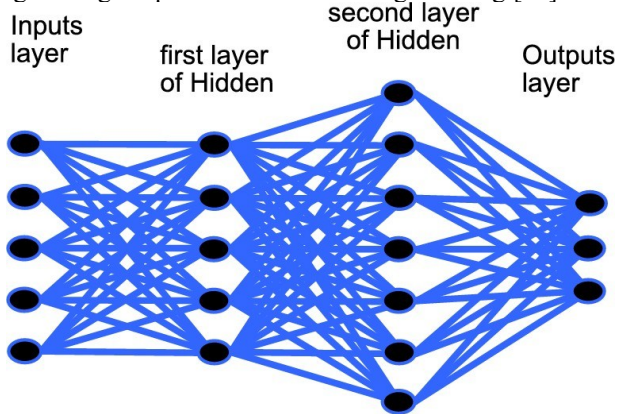
Aksara Jawa, atau Hanacaraka, adalah sistem penulisan tradisional yang digunakan untuk menulis bahasa Jawa, yang merupakan bahasa asli masyarakat Jawa di Indonesia. Aksara Jawa memiliki 20 karakter dasar, yang terdiri dari ha, na, ca, ra, ka, da, sa, wa, la, pa, dha, ja, ya, ma, ga, ba, ta, dan nga [3]. Meskipun Aksara Jawa merupakan bagian penting dari budaya dan identitas masyarakat Jawa, penggunaannya semakin menurun, terutama di kalangan generasi muda, yang mengakibatkan kekhawatiran akan punahnya warisan budaya ini [3]. Digitalisasi dan pengenalan karakter Aksara Jawa menjadi penting untuk melestarikan dan mempromosikan penggunaan aksara ini di era modern. Salah satu metode yang digunakan untuk mendigitalkan Aksara Jawa adalah melalui sistem pengenalan karakter otomatis, yang memanfaatkan teknologi pengolahan citra dan algoritma pembelajaran mesin, seperti back-propagation neural network [3].

B. Deep Learning

Deep Learning (DL) adalah cabang dari Machine Learning (ML) yang berfokus pada penggunaan jaringan saraf tiruan (Neural Networks) dengan banyak lapisan (deep architectures) untuk memodelkan dan memecahkan masalah yang kompleks. DL memungkinkan komputer untuk belajar dari data dalam jumlah besar dan menghasilkan representasi fitur yang lebih baik dibandingkan dengan metode pembelajaran tradisional. Jaringan saraf tiruan terdiri dari beberapa lapisan, termasuk input layer, hidden layers, dan output layer, di mana setiap neuron dalam lapisan tersembunyi terhubung ke neuron di lapisan sebelumnya dan berikutnya. Struktur ini memungkinkan model untuk belajar representasi data yang lebih kompleks dan abstrak.

Proses kerja Deep Learning melibatkan pemilihan arsitektur yang tepat, persiapan data, pelatihan model, dan evaluasi kinerja model dengan data yang tidak terlihat sebelumnya. Salah satu keunggulan utama dari DL adalah kemampuannya untuk melakukan ekstraksi fitur secara otomatis tanpa memerlukan intervensi manusia, yang mengurangi kebutuhan untuk rekayasa fitur manual. Deep Learning telah diterapkan dalam berbagai bidang, termasuk computer vision untuk

pengenalan objek dan pemrosesan bahasa alami untuk analisis sentimen. Meskipun memiliki banyak keunggulan, DL juga menghadapi tantangan, seperti kebutuhan data yang besar, risiko overfitting, dan kompleksitas komputasi yang tinggi. Dengan kemampuannya untuk menangani data besar dan kompleks, DL telah menjadi alat yang penting dalam pengembangan aplikasi cerdas di berbagai bidang [10].

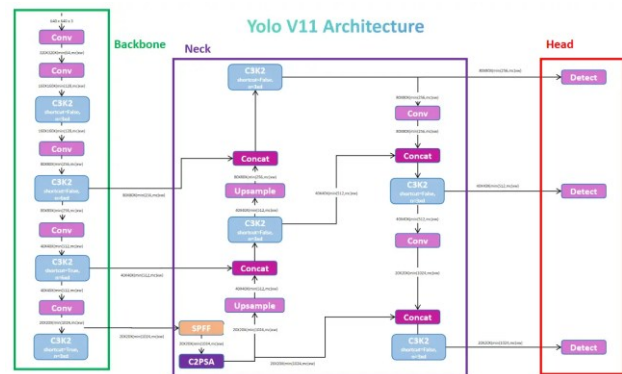


Gambar 1 Ilustrasi arsitektur pada Deep Learning

C. YOLO11

YOLO11 merupakan versi terbaru dari arsitektur *You Only Look Once* (YOLO) yang dirancang untuk deteksi objek secara real-time dengan akurasi tinggi dan efisiensi komputasi yang lebih baik dibandingkan pendahulunya. YOLOv11 mengadopsi beberapa komponen utama seperti *C3K2 blocks*, *Spatial Pyramid Pooling-Fast (SPPF)*, serta *Cross-scale Pixel Spatial Attention (C2PSA)* yang meningkatkan kemampuan dalam mengekstraksi fitur multiskala dan mengenali objek kecil pada latar belakang kompleks [11]. Dibandingkan dengan versi sebelumnya seperti YOLOv9 dan YOLOv10, YOLO11 menunjukkan peningkatan pada nilai mAP50-95, kecepatan inferensi, serta stabilitas performa dalam skenario real-time [9].

Dalam aplikasinya pada citra penginderaan jauh resolusi tinggi, YOLO11 terbukti memiliki presisi deteksi sebesar 88,61% dan F1-score 87,09%, menunjukkan keandalannya dalam mendeteksi banyak objek dan berbagai kelas secara simultan [11]. Keunggulan ini menjadikan YOLOv11 sangat cocok digunakan untuk tugas deteksi objek kompleks seperti pelestarian aksara Jawa, yang membutuhkan akurasi tinggi dalam membedakan karakter-karakter yang memiliki kemiripan bentuk.



Gambar 2 Arsitektur YOLO 11

D. Stochastic Gradient Descent (SGD)

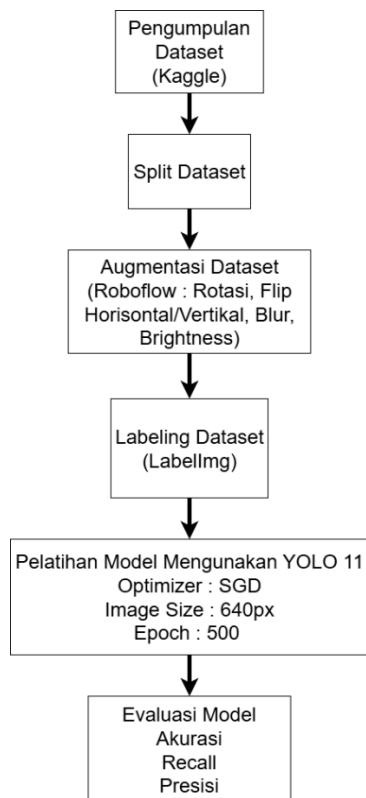
Stochastic Gradient Descent (SGD) merupakan algoritma optimasi fundamental yang memainkan peran krusial dalam pelatihan model deteksi objek YOLO (You Only Look Once). SGD adalah varian dari gradient descent yang dirancang khusus untuk efisiensi dan kecepatan ketika menangani dataset berukuran besar, dimana alih-alih menghitung gradient menggunakan seluruh dataset dalam setiap langkah, SGD memperkirakan gradient berdasarkan sampel data tunggal yang dipilih secara acak atau subset kecil yang disebut mini-batch [12]. Pendekatan ini secara signifikan mengurangi biaya komputasi dan kebutuhan memori, sehingga memungkinkan pelatihan model kompleks pada data dalam jumlah masif seperti yang ditemukan dalam bidang computer vision.

Dalam konteks arsitektur YOLO, SGD digunakan untuk mengoptimalkan parameter jaringan neural dengan meminimalkan fungsi loss yang mengukur perbedaan antara prediksi model dan nilai target sebenarnya. Penelitian terkini menunjukkan bahwa varian YOLO modern seperti YOLOv9 menggunakan konsep Programmable Gradient Information (PGI) yang dapat memberikan informasi input lengkap untuk menghitung fungsi objektif, sehingga informasi gradient yang dapat diandalkan dapat diperoleh untuk memperbarui bobot jaringan [13]. Karakteristik stokastik dari SGD membantu model keluar dari local minima yang buruk dalam landscape loss yang kompleks, yang sangat penting untuk fungsi non-convex dalam deep learning.

Implementasi modern SGD pada varian YOLO terbaru menunjukkan optimasi yang lebih canggih. Studi komprehensif oleh [14] menganalisis perkembangan terkini SGD dalam deep learning, menunjukkan bagaimana SGD dan variannya seperti Adam, AdamW, dan RMSprop telah diintegrasikan dalam framework seperti PyTorch untuk aplikasi computer vision termasuk YOLO. Sebagai contoh, dalam SOD-YOLO yang dikembangkan untuk deteksi objek kecil, SGD digunakan dengan strategi pembelajaran warm-up pada tiga epoch pertama dengan learning rate awal 0.01, kemudian menggunakan algoritma cosine annealing untuk mengontrol peluruhan learning rate [15]. Pendekatan ini memastikan konvergensi yang stabil sambil mempertahankan efisiensi komputasi yang diperlukan untuk aplikasi real-time pada perangkat edge computing.

III. METODOLOGI PENELITIAN

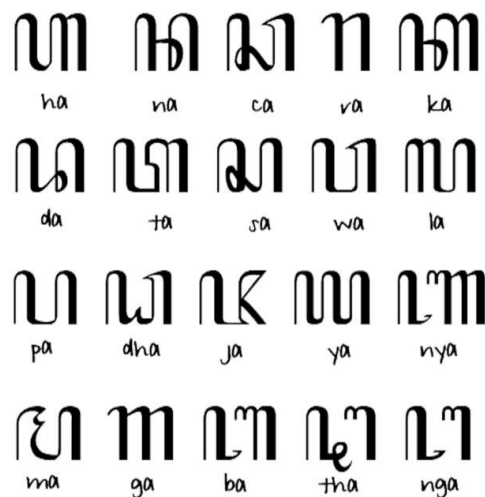
Penelitian ini menggunakan pendekatan eksperimental dengan metode deep learning untuk mengembangkan sistem deteksi karakter Aksara Jawa berbasis YOLO11. Metodologi penelitian dirancang melalui tahapan sistematis yang mencakup pengumpulan data, preprocessing, pelatihan model, dan evaluasi kinerja sistem. Pendekatan yang digunakan adalah penelitian kuantitatif dengan analisis performa model menggunakan metrik evaluasi standar dalam computer vision.



Gambar 3 Alur Penelitian

Alur penelitian dimulai dari tahap pengumpulan dataset yang relevan dengan karakteristik Aksara Jawa, dilanjutkan dengan preprocessing data untuk meningkatkan kualitas dan variasi dataset. Tahap selanjutnya adalah implementasi dan pelatihan model YOLO11, kemudian evaluasi kinerja model menggunakan berbagai metrik standar. Seluruh proses penelitian dilakukan secara iteratif untuk mengoptimalkan performa sistem deteksi.

Dataset yang digunakan dalam penelitian ini diperoleh dari sumber yang relevan dan merujuk pada penelitian-penelitian sebelumnya untuk memastikan konsistensi dan validitas data. Untuk dataset sendiri penulis dapatkan dari Kaggle dengan link <https://www.kaggle.com/datasets/hermansugiharto/aksara-jawa-yolo-v5-dataset>. Untuk contoh dataset dapat dilihat di Gambar 1.



Gambar 4. Contoh Dataset Aksara Jawa

Tahap selanjutnya yaitu preprocessing dataset, preprosesing data penting untuk mempersiapkan dataset sebelum digunakan dalam pelatihan model YOLO11. Proses dimulai dari split dataset dengan pembagian yang proporsional antara data latih, validasi, dan uji. Selanjutnya dilakukan proses labeling atau anotasi data untuk menandai lokasi dan jenis karakter Aksara Jawa dalam setiap gambar untuk proses labeling menggunakan LabelImg. Tahap augmentasi data diterapkan untuk memperbanyak jumlah dan variasi data latih melalui berbagai teknik transformasi seperti rotasi, flip, blur, dan modifikasi brightness. Teknik-teknik augmentasi ini memberikan variasi yang lebih besar pada data dan membantu model untuk lebih robust dalam menghadapi berbagai kondisi citra yang berbeda. Proses augmentasi juga bertujuan untuk mengurangi kemungkinan overfitting karena model dilatih untuk mengenali pola yang lebih beragam. Untuk preprocessing dataset menggunakan Roboflow. Untuk contoh hasil preprocessing dapat dilihat di Gambar 2.



Gambar 5 Contoh Preprocessing

Implementasi model YOLO11 dilakukan dengan menggunakan arsitektur terbaru yang mengintegrasikan komponen-komponen advanced seperti C3K2 blocks, Spatial Pyramid Pooling-Fast (SPPF), dan Cross-scale Pixel Spatial Attention (C2PSA). Proses pelatihan model dengan Optimizer SGD, lalu untuk image size adalah 640px, dan jumlah epoch sebanyak 500 untuk memastikan model mencapai konvergensi yang optimal.

Setelah model selesai di training, tahap selanjutnya adalah evaluasi model guna untuk mengukur kinerja. Evaluasi ini berguna untuk mengetahui hasil dari train dataset yang telah dilakukan apakah sudah sesuai dengan keinginan peneliti atau belum. Pada tahap ini akan tersaji dalam bentuk confusion matrix, dari confusion matrix inilah peneliti akan mengetahui bagaimana hasil akurasi dari train model, rumus untuk akurasi sendiri dapat dilihat pada persamaan (1)

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

Dimana TP (True Positive) adalah jumlah sampel positif yang diprediksi benar sebagai positif, TN (True Negative) adalah jumlah sampel negatif yang diprediksi benar sebagai negatif, FP (False Positive) adalah jumlah sampel negatif yang salah diprediksi sebagai positif, dan FN (False Negative) adalah jumlah sampel positif yang salah diprediksi sebagai negatif.

Precision dan recall dievaluasi untuk setiap kelas karakter Aksara Jawa secara individual, untuk rumus Precision dapat dilihat pada persamaan (2)

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

dimana Precision sendiri digunakan untuk mengukur ketepatan deteksi.

Sedangkan untuk rumus Recall dapat dilihat pada persamaan (3)

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

Untuk Recall sendiri digunakan untuk mengukur kelengkapan deteksi.

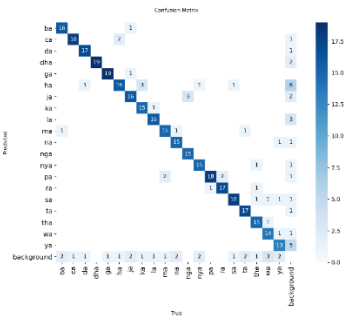
IV. HASIL DAN PEMBAHASAN

A. Hasil Pelatihan YOLO11

Pelatihan model YOLO11 untuk deteksi karakter Aksara Jawa telah berhasil dilakukan dengan menggunakan dataset yang terdiri dari 20 kelas karakter dasar Aksara Jawa plus satu kelas background. Model dilatih menggunakan optimizer SGD dengan image size 640px selama 500 epoch untuk memastikan konvergensi yang optimal. Proses pelatihan menggunakan arsitektur YOLO11 yang mengintegrasikan komponen advanced seperti C3K2 blocks, Spatial Pyramid Pooling-Fast (SPPF), dan Cross-scale Pixel Spatial Attention (C2PSA) untuk meningkatkan kemampuan ekstraksi fitur multiskala.

B. Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan menggunakan confusion matrix yang menunjukkan distribusi prediksi model terhadap label sebenarnya untuk setiap kelas karakter Aksara Jawa. Confusion Matrik dari pelatihan dataset menggunakan YOLO11 di tunjukkan pada Gambar



Gambar 6 Confusion Matrik Hasil Train

Hasil evaluasi menunjukkan bahwa model YOLO11 mencapai akurasi keseluruhan sebesar 81.00%, dengan macro-averaged precision 86.28%, macro-averaged recall 87.25%, dan macro-averaged F1-score 86.41%. Performa ini menunjukkan bahwa model memiliki kemampuan yang solid dalam mendeteksi dan mengklasifikasikan karakter Aksara Jawa dengan tingkat akurasi yang dapat diterima.

Tabel 1 menunjukkan hasil evaluasi metrik precision, recall, dan F1-score untuk setiap kelas karakter Aksara Jawa:

TABLE I HASIL EVALUASI			
Kelas	Precision (%)	Recall (%)	F1-Score (%)
ba	88.89	94.12	91.43
ca	90	90	90
da	85	94.44	89.47
dha	95	90.48	92.68
ga	90.48	95	92.68
ha	84.21	57.14	68.09
ja	80	76.19	78.05
ka	93.75	93.75	93.75
la	88.89	84.21	86.49
ma	88.24	88.24	88.24
na	83.33	88.24	85.71
nga	100	100	100
nya	100	88.24	93.75
pa	90	85.71	87.8
ra	70.83	89.47	79.07
sa	94.74	81.82	87.8
ta	80.95	94.44	87.18
tha	75	93.75	83.33
wa	77.78	87.5	82.35
ya	68.42	72.22	70.27

C. Analisis Distribusi Kinerja

Berdasarkan hasil evaluasi, distribusi kinerja model dapat dikategorikan menjadi tiga kelompok. Kategori kinerja tinggi (F1-score ≥ 90%) mencakup 7 kelas, yaitu nga (100.00%), ka (93.75%), dha dan ga (masing-masing 92.68%), nya (93.75%), ba (91.43%), dan ca (90.00%). Kelas nga menunjukkan performa sempurna dengan precision, recall, dan F1-score 100%, mengindikasikan bahwa model berhasil mengidentifikasi seluruh sampel nga tanpa kesalahan klasifikasi.

Kategori kinerja sedang ($80\% \leq F1\text{-score} < 90\%$) terdiri dari 9 kelas yang mencakup mayoritas karakter dalam dataset, seperti da (89.47%), pa dan sa (masing-masing 87.80%), ta (87.18%), la (86.49%), ma (88.24%), na (85.71%), tha (83.33%), dan wa (82.35%). Kinerja yang konsisten pada kategori ini menunjukkan bahwa model memiliki kemampuan yang memadai dalam mengidentifikasi sebagian besar karakter Aksara Jawa.

Kategori kinerja rendah ($F1\text{-score} < 80\%$) mencakup 4 kelas yang memerlukan perhatian khusus, yaitu ha (68.09%), ya (70.27%), ja (78.05%), dan ra (79.07%). Kelas ha menunjukkan kinerja terburuk dengan precision 84.21% namun recall yang sangat rendah 57.14%, mengindikasikan bahwa model sering gagal mendeteksi karakter ha yang sebenarnya ada dalam citra.

V. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem deteksi karakter Aksara Jawa menggunakan algoritma YOLO11 dengan tingkat akurasi yang baik. Model YOLO11 yang diimplementasikan mencapai akurasi keseluruhan sebesar 81.00% dengan macro-averaged precision 86.28%, macro-averaged recall 87.25%, dan macro-averaged F1-score 86.41%. Hasil ini menunjukkan peningkatan yang signifikan dibandingkan penelitian sebelumnya yang menggunakan backpropagation neural network dengan akurasi 74% dan metode backpropagation konvensional dengan akurasi 59.5%.

Analisis distribusi kinerja model menunjukkan bahwa YOLO11 mampu mendeteksi 16 dari 20 kelas karakter Aksara Jawa dengan F1-score di atas 80%, dengan 7 kelas mencapai kategori kinerja tinggi ($F1\text{-score} \geq 90\%$). Kelas "nga" menunjukkan performa sempurna dengan precision, recall, dan F1-score 100%, mengindikasikan keberhasilan model dalam mempelajari karakteristik distintif karakter tersebut. Integrasi komponen advanced seperti C3K2 blocks, Spatial Pyramid Pooling-Fast (SPPF), dan Cross-scale Pixel Spatial Attention (C2PSA) dalam arsitektur YOLO11 terbukti efektif meningkatkan kemampuan ekstraksi fitur multiskala untuk deteksi karakter Aksara Jawa.

Meskipun demikian, penelitian ini masih menghadapi beberapa tantangan, terutama dalam mendeteksi karakter yang memiliki bentuk mirip seperti "ha", "ya", "ja", dan "ra" yang menunjukkan kinerja dengan F1-score di bawah 80%. Tantangan terbesar adalah pada kelas background yang mengalami misklasifikasi total dengan precision dan recall 0%, mengindikasikan perlunya perbaikan dalam strategi preprocessing dan anotasi data. Variasi kinerja yang signifikan antar kelas dengan rentang F1-score 68.09% hingga 100% menunjukkan bahwa model memerlukan optimisasi lebih lanjut untuk mencapai konsistensi yang lebih baik.

Tujuan penelitian untuk mengembangkan sistem deteksi Aksara Jawa dengan akurasi lebih tinggi telah tercapai, meskipun belum mencapai level akurasi penelitian [6]. Kontribusi utama penelitian ini adalah penggunaan YOLO11 yang merupakan yang pertama untuk domain Aksara Jawa, serta analisis komprehensif terhadap distribusi kinerja per kelas yang memberikan insight penting untuk pengembangan

selanjutnya. Penelitian ini membuka peluang untuk optimisasi lebih lanjut melalui strategi augmentasi data yang lebih sophisticated, fine-tuning hyperparameter, dan penerapan teknik class balancing untuk meningkatkan performa pada kelas-kelas yang challenging, sehingga dapat berkontribusi lebih optimal pada pelestarian warisan budaya Jawa melalui digitalisasi yang efisien.

REFERENSI

- [1] I. Prihandi, I. Ranggadara, S. Dwiasnati, Y. S. Sari, and Suhendra, "Implementation of Backpropagation Method for Identified Javanese Scripts," *J. Phys. Conf. Ser.*, vol. 1477, no. 3, 2020, doi: 10.1088/1742-6596/1477/3/032020.
- [2] A. Susanto, I. U. W. Mulyono, C. A. Sari, E. H. Rachmawanto, and R. R. Ali, "Javanese Character Recognition Based on K-Nearest Neighbor and Linear Binary Pattern Features," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, 2022, doi: 10.22219/kinetik.v7i3.1491.
- [3] A. Setiawan, A. S. Prabowo, and E. Y. Puspaningrum, "Handwriting Character Recognition Javanese Letters Based on Artificial Neural Network," *Int. J. Comput. Netw. Secur. Inf. Syst.*, no. September, pp. 39–42, 2019, [Online]. Available: <https://ijconsist.org/index.php/ijconsist/article/view/12>
- [4] F. T. Anggraeny, Y. V. Via, and R. Mumpuni, "Image preprocessing analysis in handwritten Javanese character recognition," *Bull. Electr. Eng. Informatics*, vol. 12, no. 2, pp. 860–867, 2023, doi: 10.11591/eei.v12i2.4172.
- [5] A. Prasetiadi, J. Saputra, I. Kresna, and I. Ramadhanti, "YOLOv5 and U-Net-based Character Detection for Nusantara Script," *J. Online Inform.*, vol. 8, no. 2, pp. 232–241, 2023, doi: 10.15575/join.v8i2.1180.
- [6] I. Ikhsan and D. I. Mulyana, "Optimizing the Implementation of the Yolo and Data Algorithm Augmentation in Hanacaraka Javanese Script Language Classification," *J. Sist. Inf. dan Ilmu Komput. Prima (JUSIKOM PRIMA)*, vol. 7, no. 1, pp. 8–16, 2023, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v7i1.4062.
- [7] H. Suparwito, "Image Detection Analysis for Javanese Character Using YOLOv9 Models," *Int. J. Appl. Sci. Smart Technol.*, vol. 6, no. 1, pp. 197–208, 2024, doi: 10.24071/ijasst.v6i1.8779.
- [8] Y. Zhang, L. Zhang, Y. Hu, and T. Ju, "Recognition of Chinese Gongche notation characters based on YOLOv8," *2024 9th Int. Conf. Intell. Comput. Signal Process.*, pp. 768–771, 2024, doi: 10.1109/ICSP62122.2024.10743853.
- [9] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive Performance Evaluation of YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments," no. 2016, 2024.
- [10] I. Cholissodin and A. A. Soebroto, "AI , MACHINE LEARNING & DEEP LEARNING (Teori & Implementasi)," no. July 2019, 2021.
- [11] L. Deng, Y. Tan, D. Zhao, and S. Liu, "Research on object detection in remote sensing images based on improved horizontal target detection algorithm," *Earth Sci. Informatics*, vol. 18, no. 3, pp. 1–25, 2025, doi: 10.1007/s12145-025-01814-z.
- [12] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "YOLO-World: Real-Time Open-Vocabulary Object Detection," *Proc. IEEE World. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 16901–16911, 2024, doi: 10.1109/CVPR52733.2024.01599.
- [13] C. Y. Wang, I. H. Yeh, and H. Y. Mark Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 15089 LNCS, pp. 1–21, 2025, doi: 10.1007/978-3-031-72751-1_1.
- [14] Y. Tian, Y. Zhang, and H. Zhang, "Recent Advances in Stochastic Gradient Descent in Deep Learning," *Mathematics*, vol. 11, no. 3, pp. 1–23, 2023, doi: 10.3390/math11030682.
- [15] Y. Xiao and N. Di, "SOD-YOLO: A lightweight small object detection framework," *Sci. Rep.*, vol. 14, no. 1, pp. 1–15, 2024, doi: 10.1038/s41598-024-77513-4