

Klasifikasi Aksara Lontara Dari Sulawesi Selatan Menggunakan CNN

1st Abdul Rahim

Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

abdulrahim@student.telkomuniversity.
ac.id

2nd Febryanti Sthevanie

Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

febryantisthevanie@telkomuniversity.a
c.id

3rd Kurniawan Nur Ramadhani

Fakultas Informatika
Universitas Telkom
Bandung, Indonesia

kurniawannurramadhani@telkomuniver
sity.ac.id

Abstrak — Sebagai negara kepulauan dengan keberagaman budaya dan bahasa, Indonesia memegang peran penting dalam menyimpan dan merawat warisan budaya. Salah satu warisan tersebut adalah aksara Lontara, sebuah sistem tulisan tradisional yang telah digunakan secara luas di Sulawesi Selatan. Penelitian ini melakukan perbandingan beberapa arsitektur CNN untuk klasifikasi aksara Lontara dalam konteks OCR. Hasil eksperimen menunjukkan bahwa arsitektur CNN yaitu VGG16 mencapai performa terbaik dengan akurasi training validation sebesar 97% dan testing sebesar 90% dibandingkan dengan arsitektur VGG19, ResNet, dan ResNetV2.

Kata kunci— aksara Lontara, citra digital, ocr, klasifikasi, cnn.

I. PENDAHULUAN

Sebagai negara kepulauan yang kaya akan budaya dan bahasa, Indonesia memiliki peran krusial dalam melestarikan warisan budaya. Salah satu warisan penting tersebut adalah aksara Lontara, sistem tulisan tradisional yang banyak digunakan di Sulawesi Selatan. Aksara Lontara ini merupakan simbol dan memiliki peran signifikan dalam menyimpan catatan sejarah, sastra, dan pengetahuan tradisional masyarakat setempat[1]. Mengingat nilai budaya yang tinggi dan pentingnya pelestarian aksara ini, diperlukan teknologi yang mampu membantu memperkenalkan aksara ini kepada masyarakat luas. Salah satu teknologi tersebut adalah *Optical Character Recognition* (OCR), yang mengubah gambar menjadi teks, sehingga memudahkan pengenalan aksara Lontara oleh Masyarakat. OCR berfungsi untuk memecahkan masalah dalam ekstraksi informasi tulisan dari gambar[2].

Untuk mengenali pola dan mengklasifikasikan model yang telah dilatih, algoritma *deep learning* seperti *Convolutional Neural Network* (CNN) bisa digunakan. Saat ini, algoritma CNN merupakan solusi dalam pengolahan citra untuk mendeteksi objek pada gambar. Penelitian menunjukkan bahwa CNN efektif dalam mengklasifikasi aksara Lontara[5]. CNN merupakan pengembangan dari *Multilayer Perceptron* (MLP), dirancang khusus untuk memproses gambar atau data. Dalam *deep learning*, MLP telah dikembangkan lebih lanjut untuk menangani data yang kompleks, menghasilkan model berlapis yang dikenal

sebagai CNN. Penggunaan CNN dalam pengenalan citra dapat meningkatkan efisiensi pembelajaran aksara Lontara dengan membangun model yang efektif untuk mengklasifikasikan aksara tersebut.

Penelitian ini bertujuan untuk Penelitian ini bertujuan untuk mengklasifikasikan aksara Lontara dan membantu pengenalan serta pemahaman bentuk setiap karakter aksara. Metode CNN diterapkan dalam penelitian ini untuk mengklasifikasikan aksara Lontara dengan akurat.

II. KAJIAN TEORI

Pada penelitian[4] klasifikasi aksara Jawa menggunakan metode *Convolutional Neural Network* (CNN) berhasil mencapai tingkat akurasi sebesar 85%. Hasil menunjukkan kemampuan CNN dalam mengelompokkan aksara Jawa 'Ka' dan 'Nya'.

Penelitian [5] menunjukkan bahwa pengenalan tulis tangan dan segmentasi karakter aksara Lontara menggunakan *Convolutional Neural Network* (CNN) memberikan hasil yang baik dengan nilai akurasi sebesar 96% dan untuk kombinasi segmentasi karakter pada kumpulan data terpisah memperoleh akurasi 75%.

Penelitian [6] CNN digunakan untuk klasifikasi aksara Lampung dan menunjukkan bahwa model sudah baik yaitu dengan hasil evaluasi nilai *training* akurasi sebesar 57% serta nilai presisi sebesar 87%.

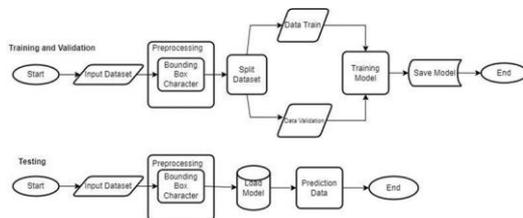
Penelitian [18] CNN digunakan untuk pengenalan pola tulis tangan aksara Arab menggunakan CNN dengan nilai akurasi sebesar 78.10%. Hal ini menunjukkan bahwa metode CNN memiliki potensi yang baik untuk dikembangkan sebagai media pembelajaran aksara Arab

A. Aksara Lontara

serta fungsi aktivasi ReLU. Dan lapisan *dense* dengan 23 *neuron* dengan fungsi aktivasi *softmax*. Ini digunakan untuk klasifikasi multi kelas yaitu sebanyak 23 kelas atau huruf.

III. METODE

A. Rancangan Sistem

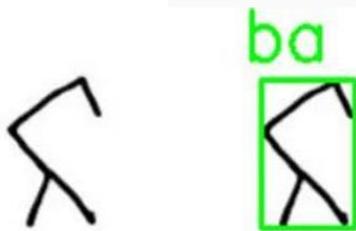


GAMBAR 5 (Flowchart)

Gambar 5 menunjukkan rancangan sistem di mana penelitian ini melibatkan 2 proses utama yaitu *training validation* dan *testing*. Sebelum memulai kedua proses tersebut, dilakukan tahapan *preprocessing* dengan *bounding box character*. Proses *training validation* dimulai dengan melatih model CNN sebanyak 100 epoch untuk menghasilkan model yang akan digunakan dalam proses *testing*. Pada proses *testing*, model yang telah dilatih akan digunakan untuk memprediksi huruf dari data *input* serta melakukan klasifikasi untuk setiap huruf.

B. Bounding Box Character

Bounding box disini digunakan untuk menandai karakter yang ingin diidentifikasi dalam sebuah gambar, memungkinkan komputer untuk melakukan pengenalan objek dengan lebih efektif. Saat komputer digunakan untuk mengklasifikasikan gambar dan mendeteksi objek, *bounding box* membantu visualisasi dengan memberi garis pembatas di sekitar objek yang ditemukan. Ini dilakukan dengan menghasilkan koordinat yang menunjukkan lokasi objek di dalam gambar serta memberi label dan batas pada objek tersebut[17].



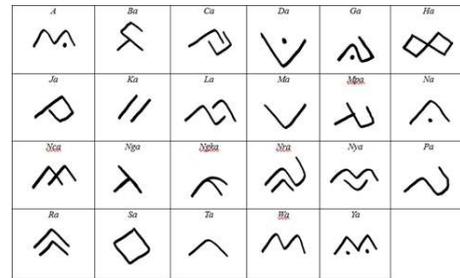
GAMBAR 6 (Bounding Box Character)

C. Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari data yang didapatkan dari *website Kaggle.com* dan data yang telah dikumpulkan peneliti secara langsung dengan cara mengambil sampel sebanyak 15 orang berbeda lalu digabungkan dengan dataset yang telah diperoleh dari *website* yang akan dijalankan di proses *testing*. Tujuan adanya dataset tambahan tersebut untuk melakukan variasi tulisan didalam dataset agar meningkatkan proses pembelajaran dari model.

Dataset terkumpul sebanyak 12.282 gambar yang dibagi menjadi data *training* 7.452 gambar, data *validation* 3.726

gambar, dan data *testing* 1.104 gambar. Didalam dataset terdapat 23 kelas yaitu *a, ba, ca, da, ga, ha, ja, ka, la, ma, mpa, na, nca, nga, ngka, nra, nya, pa, ra, sa, ta, wa, ya*. Masing-masing kelas berisi 324 gambar untuk data *training*, 162 gambar untuk data *validation*, dan 48 gambar untuk data *testing*.



GAMBAR 7 (Dataset)

D. Pelatihan Model

Untuk melatih model yang akan dibangun menggunakan *TensorFlow* dan *Keras* untuk model VGG16, VGG19, ResNet, dan ResNetV2. Proses *preprocessing* data *input* adalah *bounding box character* untuk memberi garis pembatas pada objek. Lalu mempersiapkan data *training* dan *validation* dengan ukuran gambar 64x64. Model dilatih sebanyak 100 *epoch* pada tahap *training* dan data *validation* digunakan untuk mengevaluasi model selama pelatihan. Model yang telah dilatih akan disimpan sesuai nama dari model tersebut, lalu pada tahap *testing* dilakukan untuk mengevaluasi performa model untuk klasifikasi huruf dengan data baru berupa tulis tangan yang telah peneliti kumpulkan.

E. Pengujian Model

Pengujian model dimulai dengan membandingkan grafik akurasi saat pelatihan di tahap *training* dan *validation* untuk mengetahui perbedaan antara model tersebut. Model yang digunakan ada 4 yaitu model VGG16, VGG19, ResNet, dan ResNetV2. Kemudian model akan mengklasifikasi dataset yang telah di *input* dan hasilnya akan dipetakan ke dalam *confusion matrix multiclass*. Dengan representasi visual tersebut, dapat dibandingkan performa dari semua model yang telah dibuat. Selanjutnya, model tersebut akan dibandingkan dengan menghitung nilai presisi dan *recall* untuk masing-masing kelas dalam model. Untuk menghitung *matrix* tersebut diperlukan tiga *matrix* penilaian, yaitu *True Positive (TP)*, *False Positive (FP)*, dan *False Negative (FN)*. Dari ketiga *matrix* tersebut dapat ditentukan nilai akurasi, presisi, *recall*, dan *F1-score* yang dapat digunakan untuk menilai kemampuan model secara keseluruhan. Rumus perhitungan untuk masing-masing nilai adalah :

1. Akurasi

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

2. Presisi

$$Precision = \frac{TP}{TP+FP}$$

3. Recall

$$Recall = \frac{TP}{TP+FN}$$

4. F1-Score

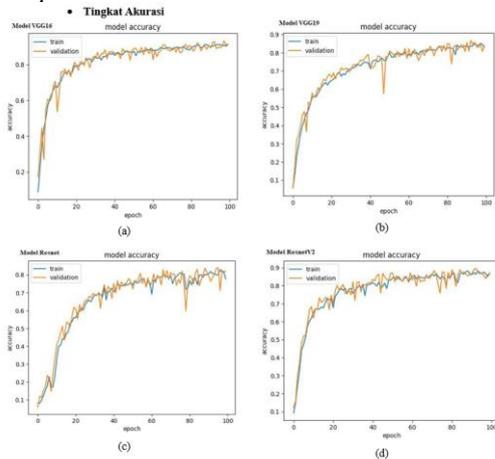
$$F1-Score = \frac{2 \times precision \times recall}{precision+recall}$$

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian

1. Training dan Validation

Pada tahap ini dilakukan skenario *training* dan *validation* pada 100 *epoch*.



GAMBAR 8 (Tingkat Akurasi)

Dari gambar 8 dapat dilihat bahwa tingkat akurasi dari setiap model terus meningkat dan tidak jauh berbeda. Hal ini dapat dipengaruhi dari jumlah data *training* dan *validation* yang sama. Untuk tingkat akurasi yang paling stabil yaitu model VGG16.

B. Tabel

TABEL 1 PERFORMANSI MODEL

Model	Akurasi	Presisi	Recall	F1-Score
VGG16	0.90	0.91	0.90	0.90
VGG19	0.83	0.84	0.83	0.83
ResNet	0.85	0.86	0.85	0.85
ResNetV2	0.86	0.88	0.86	0.86

Pada tabel 1 dapat dilihat model VGG16 memiliki tingkat akurasi, presisi, *recall*, *f1-score* yang terbaik dibandingkan model lainnya yaitu sebesar 90%.

C. Analisis Hasil Pengujian

Analisis adalah tahap krusial yang memungkinkan peneliti untuk memilih model terbaik dari berbagai model yang diuji dalam proses klasifikasi aksara Lontara. Model yang terbaik adalah model VGG16 dengan nilai akurasi 97% pada proses *training* dan *validation* serta 90% pada proses *testing*.

TABEL 2 Huruf aksara Lontara tertinggi dan terendah

	Huruf	Prediksi
Akurasi tertinggi		

Akurasi terendah		
------------------	--	--

Dapat dilihat pada tabel 2 huruf dengan akurasi tertinggi adalah huruf 'ka', sedangkan akurasi terendah adalah huruf 'wa' dikarenakan huruf 'wa' tersebut memiliki kemiripan dengan beberapa huruf seperti huruf 'a' dan huruf 'ya'.

V. KESIMPULAN

Berdasarkan hasil penelitian, disimpulkan bahwa metode CNN efektif dalam merancang dan mengimplementasikan sistem klasifikasi aksara Lontara. Hal ini terbukti dari hasil perbandingan performa beberapa arsitektur atau model CNN. Dari hasil perbandingan tersebut, model VGG16 menunjukkan tingkat akurasi terbaik, dengan nilai akurasi mencapai 97% pada tahap *training* dan *validation*, serta 90% pada tahap *testing*.

Saran untuk penelitian selanjutnya adalah mengembangkan model klasifikasi menggunakan metode yang berbeda dan dataset yang lebih beragam. Hal ini akan memperluas kemampuan klasifikasi aksara Lontara.

REFERENSI

- [1] Abd. Aziz Ahmad., 'Melestarikan Budaya Tulis Nusantara'. 2014.
- [2] Aldi Setiawan, Herry Sujaini, Arif Bijaksana, 'Implementasi Optical Character Recognition (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris', 2017.
- [3] Hardian Oktavianto, Henny Wahyu Sulisty, 'Optical Character Recognition Untuk Ekstraksi Teks Rambu Lalu Lintas', 2018.
- [4] Ivan Sukma Hanindria, Hendry, 'Pengklasifikasian Aksara Jawa Metode Convolutional Neural Network', 2022.
- [5] Asri Hidayat, Ingrid Nurtanio, Zulkifli Tahir, 'Segmentation and Recognition of Handwritten Lontara Characters Using Convolutional Neural Network', 2020.
- [6] Mulyanto, A. et al., 'Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR)', *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 2021, 7(1), pp. 52-57.
- [7] Umam, C. and Budi Handoko, L., 'Convolutional Neural Network (CNN) Untuk Identifikasi Karakter Hiragana', *Prosiding Seminar Nasional Lppm Ump*, 0(0), 2020, pp. 527-533. Available at: <https://semnaslppm.ump.ac.id/index.php/semnaslppm/article/view/199>.
- [8] Agi Prasetyadi, Julian Saputra, Iqsyahiro Kresna, Imada Ramadhanti, 'Deep Learning Approaches for Nusantara Scripts Optical Character Recognition using Convolutional Neural Network, ConvMixer, and Visual Transformer', 2023.

- [9] Bora, M. B. *et al.* (2020) 'Handwritten Character Recognition from Images using CNN-ECOC', *Procedia Computer Science*, 167, 2019, pp. 2403–2409. doi: 10.1016/j.procs.2020.03.293.
- [10] Fonda, H., 'Klasifikasi Batik Riau Dengan Menggunakan Convolutional Neural Networks (Cnn)', 2020, *Jurnal Ilmu Komputer*, 9(1), pp. 7–10. doi: 10.33060/jik/2020/vol9.iss1.144.
- [11] Disa Ainun Safitri, 'Pengenalan Pola Aksara Lontara dengan Metode Optical Character Recognition', 2018.
- [12] Viktar Atliha, Dmitrij Sesok, 'Comparison of VGG and ResNet used as Encoders for Image Captioning', 2020.
- [13] Andreanov Ridhovan, Aries Suharso, 'Penerapan Metode *Residual Network* (ResNet) Dalam Klasifikasi Penyakit Pada Daun Gandum', 2022.
- [14] Keiron O'Shea, Ryan Nash, 'An Introduction to Convolutional Neural Network', 2020.
- [15] J.T. Townsend, 'Theoretical of an alphabetic confusion matrix', 1971.
- [16] Ngo Le Huy Hien, Van Hieu N., 'Recognition of Plant Species using Deep Convolutional Feature Extraction', 2020.
- [17] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, Toby Sharp, 'Image Segmentation with A Bounding Box Prior', 2009.
- [18] Nanang Kasim, Gibran Satya Nugraha, 'Pengenalan Pola Tulisan Tangan Aksara Arab Menggunakan Metode CNN', 2021.