

PENGENALAN AKSARA LONTARA TULIS TANGAN MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORKS BERBASIS ANDROID

Ridwan Nur¹, Septi Andryana², Winarsih³

¹²³Teknik Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional
Jl. Sawo Manila, Pasar Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 1252
Telp. (021) 7806700

e-mail : riduvannur@gmail.com¹, septi.andryana@civitas.unas.ac.id², winarsih@civitas.unas.ac.id³

ABSTRACT

Convolutional Neural Network (CNN) is the latest method in object recognition which is the development of Multilayer Perceptron (MLP). CNN has a special layer of convolution layer and layer max pooling to process feature training hierarchically from data, with spatial data inputting. As a method of handwriting recognition method, CNN method is the best compared to method or similar model at this time. Therefore the CNN method is used in this research, for the recognition of Lontara script handwriting character. Using the CNN M8 model and TensorFlow open-source library for Lontara script image data training. After that made simple android interface as a medium of input Lontara script write hand through canvas for performance evaluation. From the test results obtained accuracy of up to 90%, by testing as many as 30 forms of Lontara.

Keywords— handwritten recognition, CNN, lontara, tensorflow, android

ABSTRAK

Convolutional Neural Network (CNN) adalah metode terbaru di bidang pengenalan objek yang merupakan pengembangan dari Multilayer Perceptron (MLP). CNN memiliki layer khusus berupa layer konvolusi dan layer max pooling untuk memproses pelatihan fitur secara hierarki dari data, dengan penginputan data yang bersifat spasial. Sebagai metode pengenalan karakter tulisan tangan, metode CNN adalah yang terbaik dibandingkan metode ataupun model sejenis saat ini. Karena itu metode CNN dimanfaatkan dalam penelitian ini, untuk pengenalan karakter tulisan tangan aksara Lontara. Dengan menggunakan model CNN M8 dan pustaka open-source TensorFlow untuk pelatihan data citra aksara Lontara. Setelah itu dibuat antarmuka android sederhana sebagai media penginputan aksara Lontara tulis tangan melalui kanvas untuk evaluasi kinerja pelatihan. Dari hasil pengujian didapatkan hasil akurasi 90%, dengan menguji sebanyak 30 bentuk huruf Lontara.

Kata Kunci— pengenalan tulisan tangan, CNN, lontara, tensorflow, android

I. PENDAHULUAN

Indonesia sebagai negara kepulauan memiliki berbagai warisan budaya, salah

satunya adalah aksara, yang digunakan untuk menuliskan bahasa daerah tertentu.

Salah satu dari banyak aksara yang masih bertahan hingga kini adalah aksara

Lontara yang merupakan aksara tradisional masyarakat Bugis-Makassar. Aksara Lontara terdiri dari 23 huruf konsonan, 5 huruf vokal, dan 2 tanda baca. Gambar 1 menunjukkan 23 huruf konsonan dengan penulisan Roman.

ka	ga	nga	ngka	pa	ba	ma	mpa
ta	da	na	nra	ca	ja	nya	nca
ya	ra	la	wa	sa	a	ha	

Gambar 1. Konsonan Aksara Lontara

Dengan sistem penulisan alfasilabis, penulisan huruf Lontara didasarkan pada huruf konsonan dengan dibantu notasi vokal sekunder. Gambar 2 menunjukkan 5 huruf vokal beserta contoh penggabungan dengan huruf konsonan Ra.

i	u	e	o	ə
ri	ru	re	ro	rə

Gambar 2. Huruf Vokal Lontara dan Contoh Penggunaan

Tanda baca aksara Lontara terdiri dari 2, yaitu Pallawa dan Tanda Akhir Bagian. Dimana Tanda Pallawa dapat berfungsi sebagai koma, titik ataupun pengulangan.

Pallawa	Akhir Bagian
---------	--------------

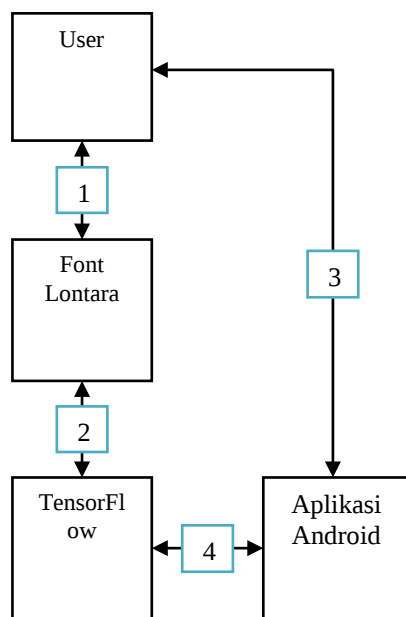
Gambar 3. Tanda Baca Pallawa dan Tanda Akhir Bagian

Dalam kebudayaan masyarakat Bugis-Makassar, pemanfaatan aksara Lontara sering dijadikan sebagai alat komunikasi “rahasia” yang hanya dapat dipahami secara terbatas bagi si pengirim dan si penerima [1]. Namun seperti pada aksara daerah di Indonesia lainnya, yang dalam keseharian di masyarakat penggunaan aksara sangat jarang digunakan dan diharapkan aksara Lontara dapat menjadi alat komunikasi tertulis seperti aksara pada umumnya. Di era teknologi dan masyarakat milenial kini, usaha yang dapat dilakukan untuk mendukung pelestarian dan pengembangan aksara Lontara adalah dengan melakukan penelitian untuk pengenalan aksara Lontara tulis tangan seperti yang dilakukan oleh Alwi dan Wardoyo [2]. Dengan antarmuka android untuk menarik minat pemuda/i untuk mempelajari aksara Lontara.

Dalam pengenalan citra aksara Lontara, metode yang digunakan dalam penelitian ini adalah *Convolutional Neural Networks (CNN)* yang merupakan model pengembangan dari pengolah data 2D *Multilayer Perceptron (MLP)* [3], dimana metode *CNN* menjadi metode yang termutakhir dalam peng-klasifikasian citra saat ini [4].

II. METODE PENELITIAN

Tujuan dari penelitian ini adalah membuat sebuah aplikasi android untuk pengenalan aksara Lontara tulis tangan dengan menggunakan metode *CNN*. Di bawah ini merupakan *flowchart* aplikasi pengenalan aksara Lontara tulis tangan:



Gambar 4. Flowchart Aplikasi

Tahap awal yakni mengumpulkan data *fonts* yang berisi aksara Lontara dengan bentuk huruf yang bervariasi (1). Kemudian *fonts* dilatih dengan metode *CNN* menggunakan pustaka TensorFlow (2). User menginput huruf Lontara pada kanvas android (3). Huruf Lontara yang ditulis pada kanvas android dikenali dengan database TensorFlow yang telah dilatih sebelumnya (4).

A. Dataset

Pembuatan dataset dilakukan dengan mengumpulkan variasi *fonts* yang di-

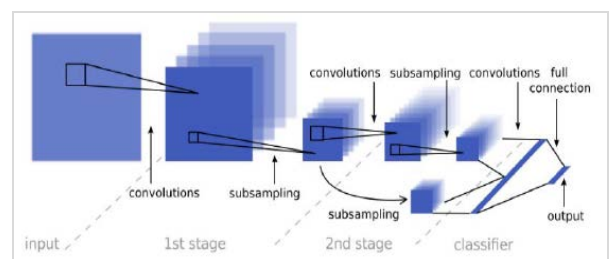
dalamnya terdapat 30 huruf Lontara. Dan terkumpul sebanyak 8 bentuk huruf yang berbeda dalam format *TrueType Font*, sehingga variasi citra aksara Lontara yang terkumpul berjumlah 240.



Gambar 5. Variasi Bentuk Huruf ꨀꩃ (Ca)

Kemudian *fonts* di-generate dengan ukuran piksel 64x64 *grayscale*, ditambah distorsi 3x pada setiap bentuk huruf dan melabeli data citra dengan label *unicode* aksara Lontara yang sesuai. Total dari pen-generate-an menghasilkan data citra berjumlah 960 citra dengan format *JPEG*, kemudian data citra beserta labelnya dikonversi ke format standar input TensorFlow (TFRecords).

B. Metodologi



Gambar 6. Contoh Arsitektur CNN [4]

Arsitektur *CNN* terdiri dari 3 kombinasi yaitu layer konvolusi, layer *max-pooling*, dan klasifikasi. Layer konvolusi dan layer *max-pooling* adalah dua jenis lapisan pada

tingkat rendah dan menengah pada jaringan. Layer nomor genap berhubungan dengan layer konvolusional dimana layer nomor yang ganjil bekerja untuk layer *max-pooling* di dalam arsitektur. Output dari keduanya disatukan pada bidang 2 dimensi [8].

Tujuan dari operasi konvolusi dan *max-pooling* adalah untuk mengekstraksi fitur dari data input. Untuk itu, operasi konvolusi berupa perkalian matriks kernel kecil dan area tertentu dari input matriks 2 dimensi dilakukan. Untuk menghasilkan satu dimensi yang lebih kecil dari peta fitur input matriks, kernel digeser dan dilakukan perkalian dari kiri ke kanan dan dari atas ke bawah pada area dalam matriks input. Persamaan untuk operasi konvolusi didefinisikan pada (1) yang dinyatakan dalam [9]:

$$Q_j = f\left(\sum_{i=1}^N I_{i,j} * K_{i,j} + B_j\right), \quad (1)$$

Dimana Q_j merupakan elemen dari output matriks tunggal dari operasi konvolusi. Output matriks dihasilkan dari fungsi aktivasi f . Pertama, jumlah semua perkalian matriks kernel $K_{i,j}$ dan $I_{i,j}$ dihitung, selanjutnya nilai bias B_j ditambahkan ke elemen matriks yang dihasilkan. Dalam penelitian ini digunakan fungsi aktivasi ReLU (*Rectified Linear Unit*) dengan rumus pada (2).

$$f(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0). \end{cases} \quad (2)$$

Setelah operasi konvolusi, dilakukan operasi *subsampling* atau *pooling* ke masing-masing peta fitur untuk mereduksi dimensi. Fungsi penggabungan *max-pooling* sebagai *subsampling* digunakan dalam penelitian ini untuk memperoleh fitur yang dominan. Reduksi dimensi dari peta fitur tunggal dilakukan dengan satu kernel $m \times n$ 2 dimensi yang akan memilih nilai paling tinggi dari $(m \times n)$ elemen-elemen yang berdekatan dan memproduksi elemen tunggal dalam sebuah matriks fitur map yang baru. Hampir sama dengan operasi konvolusi, yaitu kernel akan digeser dari kiri ke kanan dan dari atas ke bawah untuk memproduksi peta fitur yang baru.

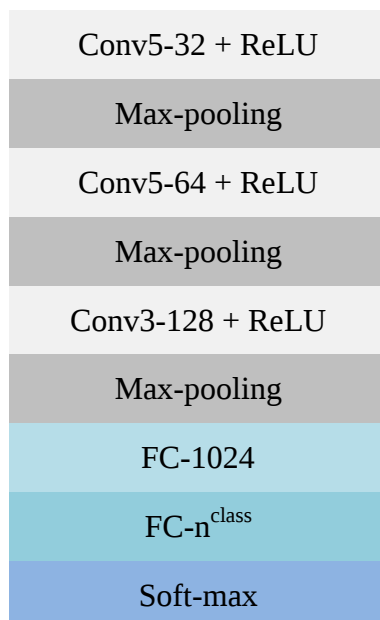
Setelah beberapa kali operasi konvolusi dan *subsampling*, peta fitur akan menjadi rata sehingga dapat diklasifikasikan dengan jaringan saraf tiruan yang telah terkoneksi penuh (*Fully-connected Neural Networks*) dari setiap lapisan atau layer. Neuron dari setiap layer-layer kemudian melakukan perkalian matriks antara input matriks dan bobot internal dengan rumus (3):

$$u_j = f\left(\sum_{i=1}^n w_{ji}x_i + b_j\right), \quad (3)$$

b_j adalah nilai bias, n adalah jumlah neuron dalam satu layer tunggal, dan f adalah fungsi aktivasi, yakni fungsi ReLU seperti yang didefinisikan pada (2). Setelah diproses dalam beberapa layer, fitur peta akan diproses pada layer output dengan fungsi *softmax* yang menghasilkan kemungkinan pada kelas-kelas $p(x)$ dari hasil input *CNN* yang ada. Rumus fungsi *softmax* (4).

$$p(x) = \frac{e^x}{\sum_{k=1}^K e^x} \quad (4)$$

Dalam penelitian ini, model *CNN* yang digunakan untuk pelatihan data citra adalah model M8 dengan 3 layer konvolusi [10] seperti yang terlihat pada gambar 7 dengan sedikit penyesuaian.



Gambar 7. Model CNN M8 [10]

Conv5-32 berarti matrix pada layer konvolusi 5x5 dengan 32 fitur output.

Matrix pada layer *max-pooling* 2x2. FC-1024 berarti layer *Fully Connected* dengan 1024 fitur output. Layer *softmax* untuk mendapatkan hasil klasifikasi [11].

C. Pelatihan

Pembuatan model *CNN* menggunakan pustaka TensorFlow dengan bahasa pemrograman utama yang digunakan dan didukung adalah Python. Format standar input pada TensorFlow adalah TFRecords, yang merupakan format biner yang dapat digunakan untuk menyimpan data citra mentah beserta label pada satu tempat. Dibuat beberapa file TFRecords untuk memberi data yang lebih baik pada model TensorFlow.

TensorFlow yang digunakan dalam penelitian ini adalah versi CPU, dan CPU yang digunakan adalah Intel Core i5-2400. Pelatihan dataset dengan TensorFlow mencakup beberapa iterasi dari seluruh data latih. Dalam penelitian ini data citra terdiri dari 960 citra, $960 / 100 = 9,6 \text{ steps}$, 100 adalah ukuran *batch* default. Kemudian hasil tersebut dikalikan dengan jumlah pelatihan data, yakni 30 kali, menjadi $9,6 * 30 = 288 \text{ steps}$. Dari hasil pelatihan data citra mendapatkan akurasi latihan 64,9% dan akurasi pengujian model 95%. Hasil pelatihan dengan TensorFlow kemudian disimpan dalam satu format *Protocol Buffer* [12] yang

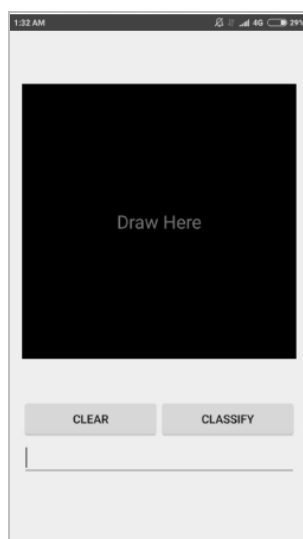
merepresentasikan hasil pelatihan untuk digunakan sebagai database ke dalam aplikasi android.

```
Step 0, Training Accuracy 0.04
Step 100, Training Accuracy 0.83
Step 200, Training Accuracy 1
Testing model...
Testing Accuracy 0.95
```

Gambar 8. Hasil Pelatihan Dataset dengan CNN TensorFlow

D. Aplikasi

Sebuah aplikasi android sederhana dibuat untuk mengevaluasi kinerja pengenalan aksara Lontara yang telah dilatih. Dengan media kanvas untuk menginput huruf yang diinginkan secara tertulis.



Gambar 9. Antarmuka Pengenalan Aksara Lontara

Antarmuka terdiri dari kanvas untuk menulis huruf Lontara, tombol *CLEAR* untuk membersihkan kanvas, dan tombol *CLASSIFY* untuk mengklasifikasikan aksara yang tertulis pada kanvas. Output yang dihasilkan berupa label huruf yang

sesuai dengan dataset citra yang telah dilatih.

III. HASIL DAN PEMBAHASAN

A. Pengujian Aplikasi

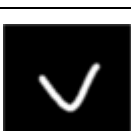
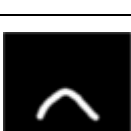
Pengujian aplikasi dilakukan dengan menginput pada kanvas secara tulis tangan bentuk huruf Lontara dengan piksel citra 64x64. Setelah diklasifikasikan, maka bentuk huruf yang diinput dikenali kesesuaian pola dengan dataset citra terlatih. Seperti yang terlihat pada Gambar 10.

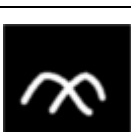
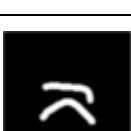
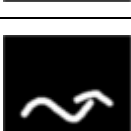
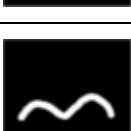
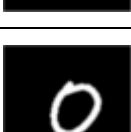


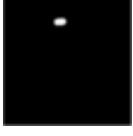
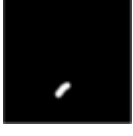
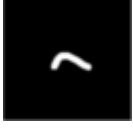
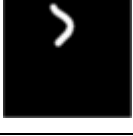
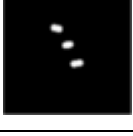
Gambar 10. Pengujian Menginput Huruf □ (Nra)

Huruf-huruf dibawah output merupakan alternatif huruf yang disarankan bila terjadi kesalahan pengenalan huruf. Berikut hasil evaluasi kinerja aplikasi pengenalan aksara Lontara dengan menginput 30 bentuk huruf Lontara satu persatu.

Tabel 1. Hasil Pengujian 30 Input Lontara

No.	Input	Output	Keterangan
1			Benar
2			Salah
3			Benar
4			Benar
5			Benar
6			Benar
7			Benar
8			Benar
9			Benar
10			Benar

11			Salah
12			Benar
13			Benar
14			Benar
15			Benar
16			Benar
17			Benar
18			Benar
19			Benar
20			Benar
21			Benar

22			Benar
23			Benar
24			Salah
25			Benar
26			Benar
27			Benar
28			Benar
29			Benar
30			Benar

B. Pembahasan

Pada Tabel 1 dapat dilihat bahwa hasil pengujian untuk pengenalan 30 input aksara Lontara, yakni 27 input yang sesuai dengan output. Dapat dianalisa kesalahan pada input 11 dan 24 dalam Tabel 1

disebabkan kurang bervariasinya dataset, ditambah dengan bentuk huruf yang hanya berupa simbol titik. Selain itu karena bersifat spasial, kelemahan dalam mengenali pola aksara dengan benar pada bentuk huruf yang posisinya berada kurang pas dengan dataset, yaitu berada ditengah kanvas. Hal tersebut terjadi seperti pada kesalahan pengenalan pada input 2.

C. SIMPULAN

1. Dalam penelitian ini telah dibuat suatu aplikasi sederhana untuk pengenalan aksara Lontara berbasis android. Dengan tahap praproses diantaranya tahap pembuatan dataset dari data berupa *font*, tahap pelatihan dataset menggunakan metode *CNN* dengan pustaka TensorFlow, hingga pembuatan aplikasi android sebagai media kanvas untuk menginput tulisan tangan yang digunakan untuk pengujian dari hasil pelatihan menggunakan metode *CNN*. Hasil evaluasi kinerja aplikasi aksara Lontara melalui media input kanvas pada android, berhasil mendapatkan tingkat akurasi 90% dengan dataset yang diolah dari 8 jenis *fonts* berbeda.
2. Namun karena pengenalan bersifat spasial, artinya bila penulisan pada kanvas tidak tepat pada posisi dataset

yakni di tengah kanvas. Maka huruf tidak dapat dikenali dengan tepat. Untuk itu agar akurasi pengenalan dalam pembuatan dataset disarankan menambah letak citra aksara pada posisi-posisi yang lebih beragam.

DAFTAR PUSTAKA

- [1] Ahmad, A. A. (2014). Melestarikan Budaya Tulis Nusantara: Kajian tentang Aksara Lontara. *Jurnal Budaya Nusantara*, 1(2), 148-153.
- [2] Alwi, A., & Wardoyo, R. (2010). *Pengenalan pola huruf-huruf lontara Bugis-Makassar dengan menggunakan jaringan syaraf tiruan metode backpropagation* (Doctoral dissertation, Universitas Gadjah Mada).
- [3] Palangkaraya, M. H. Q. S. (2017). Pengenalan Huruf Dan Angka Tulisan Tangan Menggunakan Metode Convolution Neural Network (CNN). *Speed-Sentra Penelitian Engineering dan Edukasi*, 9(2).
- [4] Dewa, C. K., Fadhillah, A. L., & Afiahayati, A. Convolutional Neural Networks for Handwritten Javanese Character Recognition. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 12(1), 83-94.
- [5] Khadijah, Rismiyati, and Adi Nurhadiyatna. "Deep learning for handwritten Javanese character recognition." *Informatics and Computational Sciences (ICICoS), 2017 1st International Conference on*. IEEE, 2017.
- [6] Fitriati, D. (2016). PERBANDINGAN KINERJA CNN LeNet 5 DAN EXTREME LEARNING MACHINE PADA PENGENALAN CITRA TULISAN TANGAN ANGKA. *Jurnal Teknologi Terpadu*, 2(1).
- [7] Ertam, Fatih, and Galip Aydın. "Data classification with deep learning using Tensorflow." *Computer Science and Engineering (UBMK), 2017 International Conference on*. IEEE, 2017.
- [8] Tanya Makkar, Yogesh, Ashwani Kumar Dubey, Álvaro Rocha, and Ayush Goyal, "Analogizing Time Complexity of KNN and CNN in Recognizing Handwritten Digits", Accepted for 2017 Fourth International Conference on Image Information Processing (ICIIP - 2017), IEEE, at Jaypee University of Information Technology, Solan, Himachal Pradesh, India, 21st -23rd Dec. 2017.
- [9] Li, Qing, et al. "Medical image classification with convolutional neural network." *Control Automation Robotics & Vision (ICARCV), 2014*

- 13th International Conference on.
IEEE, 2014.
- [10] Tsai, Charlie. "Recognizing handwritten Japanese characters using deep convolutional neural networks." (2016): 1-7.
- [11] Susilo, Mellysa Margarita, Daniel Martomanggolo Wonohadidjojo, and Nehemia Sugianto. "Pengenal Pola Karakter Bahasa Jepang Hiragana Menggunakan 2D Convolutional Neural Network." *Jurnal Informatika dan Sistem Informasi* 3.2 (2018): 28-36.
- [12] Goldsborough, Peter. "A tour of tensorflow." arXiv preprint arXiv:1610.01178 (2016).
- [13] <https://developer.ibm.com/code/patterns/create-mobile-handwritten-hangul-translation-app/>
"Create a mobile application leveraging TensorFlow that will recognize and translate handwritten Korean characters", 12 Maret 2018.
- [14] <https://medium.com/@samuelsena/pengenal-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>
"Pengenal Deep Learning Part 7 : Convolutional Neural Network (CNN)", 5 Maret 2018.