

Pengenalan Karakter Braille Memanfaatkan Convolutional Neural Network

Marindo Andriansyah^{*1}, Hartarto Junaedi²

^{1,2} Teknologi Informasi, Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terpadu Surabaya
Email : *marindoandriansyah@gmail.com

ABSTRACT

Braille is a character designed for the blind. Braille letters consist of six raised dots arranged in three rows and two columns. Braille is read by touch, so finger sensitivity is very important. Braille combinations need to be memorized, making them very difficult to learn. This study discusses the introduction of braille characters using the Convolutional Neural Network (CNN) method. CNN will process 3 data sets, 60, 100, and 150 data with each data using 5, 10, 25, and 50 epochs. The highest accuracy value in the training process is 99.87% with a loss value of 0.232. In the recognition process, the highest accuracy is 99.62% with a recognition error of 1 image out of 260 images.

Keywords: Braille, CNN, Character, Recognition.

ABSTRAK

Huruf braille merupakan karakter yang dirancang untuk orang buta. Huruf braille terdiri dari enam titik timbul yang tersusun dalam tiga baris dan dua kolom. Huruf braille dibaca dengan menggunakan sentuhan, oleh sebab itu sensitifitas jari sangat penting. Kombinasi huruf braille perlu dihafalkan, sehingga sangat sulit untuk dipelajari. Penelitian ini membahas pengenalan karakter braille dengan menggunakan metode Convolutional Neural Network (CNN). CNN akan memproses 3 kelompok data, 60, 100, dan 150 data dengan masing-masing data menggunakan 5, 10, 25, dan 50 epoch. Nilai akurasi tertinggi pada proses training sebesar 99.87% dengan nilai loss sebesar 0.232. Dalam proses pengenalan akurasi tertinggi sebesar 99.62% dengan kesalahan pengenalan 1 gambar dari 260 gambar.

Kata Kunci: Braille, CNN, Karakter, Pengenalan.

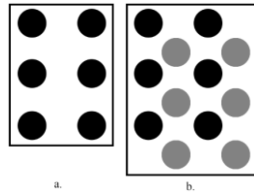
1. Pendahuluan

Membaca merupakan salah satu cara untuk mendapatkan informasi, namun bagi mereka yang buta atau tunanetra akan sulit jika membaca karakter biasa. Buta atau tunanetra adalah hilangnya kemampuan untuk melihat dalam indra pengelihatan. Tunanetra terbagi menjadi dua yaitu buta total dan yang masih mempunyai sisa penglihatan. Oleh karena itu, karakter untuk orang buta dirancang khusus, bernama karakter braille [1].

Karakter Braille merupakan karakter yang dirancang untuk penyandang tunanetra. Karakter braille

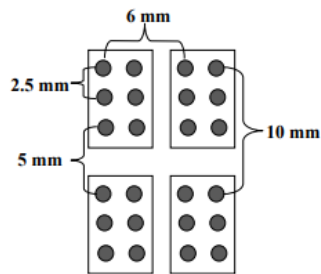
terdiri dari enam kombinasi titik timbul yang diatur dalam matriks tetap dengan ukuran 3 baris dua kolom yang disebut sel. Setiap titik dapat ditata atau dihapus sehingga dapat memberikan 64 kombinasi. Karakter Braille memiliki kepadatan informasi yang rendah, rata-rata halaman berukuran 25 x 29 cm, dan memiliki 34 karakter dalam satu baris dan 26 baris dalam satu halaman [2].

Terdapat dua jenis kombinasi karakter Braille yaitu satu sisi dan dua sisi (bolak-balik) seperti yang ditunjukkan pada Gambar 1 [3].



Gambar 1. Sel Braille a) Satu Sisi b) Dua Sisi.

Ciri-ciri fisik sel Braille yang digunakan untuk pengenalan karakter Braille ditunjukkan pada Gambar 2. Banyak penelitian tentang karakter Braille telah dilakukan di banyak negara. Tujuan dari penelitian ini adalah memudahkan membaca teks Braille untuk pembelajaran. Sehingga dapat membantu siapa saja yang tidak tahu Bahasa Braille namun berurusan dengan orang tunanetra [4].



Gambar 2. Ciri-ciri Fisik Sel Braille.

Perkembangan teknologi komputer saat ini banyak memberikan kemudahan dalam berbagai hal karena teknologi komputer sudah menyatu dengan keseharian manusia [5]. Penelitian tentang pengenalan karakter Braille banyak yang memakai metode *computer vision*. Saat ini *machine learning* menjadi metode yang paling sering digunakan, dimana komputer mempelajari karakteristik model dari banyak data. *Convolutional Neural Network* (CNN) merupakan algoritma yang paling digemari dalam pemrosesan gambar. Dalam penelitian ini CNN digunakan untuk melatih data sehingga dapat mengenali karakter Braille. Diharapkan dengan

menggunakan CNN nilai akurasi yang didapat menjadi lebih tinggi.

2. Tinjauan Pustaka

Penelitian tentang pengenalan karakter braille telah dilakukan sebelumnya dan mendapatkan hasil dengan akurasi yang baik.

De Silva, et al., (2019) Dalam penelitian dengan judul *Optical Braille Recognition Platform for Sinhala*, [6] hasil penelitian menunjukkan bahwa sistem yang diusulkan mampu mengekstrak titik-titik Braille dan konversi dokumen braille tulisan tangan yang digunakan pada ujian dan evaluasi menjadi lebih mudah dan efektif. Sistem yang diusulkan tidak memerlukan perangkat keras dengan daya pemrosesan tinggi, peralatan rumit, hemat biaya, dapat bekerja secara *offline* dan dapat dibagikan sebagai file yang dapat dieksekusi.

Perera, T. D. S. H., et al., (2018) Hasil dari penelitian yang berjudul *Optical Braille Recognition Based on Histogram of Oriented Gradient Features and Support-Vector Machine*, [7] menunjukkan Sinhala Braille dapat diterjemahkan ke bahasa Sinhala atau Braille Inggris tingkat 1 ke bahasa Inggris secara akurat. Sistem juga berhasil mengenali angka dalam Braille Sinhala dan Braille Bahasa Inggris tingkat 1. Secara keseluruhan sistem berhasil diterapkan untuk memfasilitasi komunikasi antara penyandang tunanetra dan orang awas.

K. Smelyakov, et al., (2018) Penelitiannya yang berjudul *Braille Character Recognition Based on Neural Networks*, [4] menghasilkan sistem identifikasi Braille dengan sejumlah gambar yang diperlukan untuk pelatihan dan pengujian dapat mencapai akurasi yang baik menggunakan *Neural Network*, sehingga mampu membantu siapa saja

yang tidak tahu Bahasa Braille, tetapi berurusan dengan orang tunanetra.

3. Metodologi Penelitian

Untuk dapat mengenali karakter Braille diperlukan beberapa tahapan mulai dari *preprocessing*, *segmentation*, dan *classification*. Dataset dalam penelitian ini berupa gambar hasil *scanner* dokumen braille yang terdiri alfabet dari A – Z seperti pada Tabel 1.

Tabel 1. Daftar Kelas Karakter Braille.

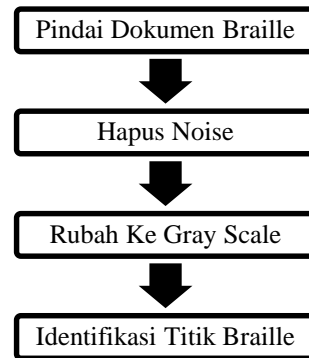
Braille	Alfabet	Braille	Alfabet
•	A	••	N
••	B	••	O
••	C	••	P
••	D	••	Q
••	E	••	R
••	F	••	S
••	G	••	T
••	H	••	U
••	I	••	V
••	J	••	W
••	K	••	X
••	L	••	Y
••	M	••	Z

3.1 Preprocessing

Preprocessing diperlukan untuk mendapatkan gambar input yang siap untuk diolah ketahap *segmentasi*. Adapun tahapan dalam *preprocessing* ditunjukkan pada Gambar 3.

Pertama dokumen Braille dipindai menggunakan alat pemindai resolusi tinggi dengan resolusi sebesar 1200 dpi dan berekstensi **.tif**. Setelah gambar didapatkan, selanjutnya

menghapus *noise* pada gambar dengan menggunakan *median filtering* [8]. Gambar yang bersih dari *noise* memiliki ketajaman yang berpengaruh dalam memperoleh informasi dengan baik [9]–[13].



Gambar 3. Alur Proses *Preprocessing*.

Gambar yang sudah bersih dari *noise* dirubah menjadi *gray scale*, gambar *gray scale* lebih mudah untuk diidentifikasi setiap titik braille yang ada.



Gambar 4. Hasil *Preprocessing*

3.2 Segmentation

Hasil dari proses *preprocessing* akan menjadi input untuk proses *segmentation*. Proses *segmentation* menggunakan *thresholding* [14] untuk merubah gambar dari *gray scale* menjadi *black white*. Dalam proses ini nilai *pixel* gambar yang lebih atau sama dengan nilai *threshold* akan diubah menjadi putih (1) dan nilai *pixel* yang kurang dari nilai *threshold* akan dirubah menjadi hitam (0).

Untuk mendapatkan gambar braille dalam satu baris perlu memotong setiap tepi dokumen sehingga gambar tidak memiliki ruang kosong pada setiap tepinya. Selanjutnya gambar akan dipotong

sama rata dengan ukuran yang sesuai dengan panjang baris dan lebar 3 titik braille menjadi 26 baris seperti pada gambar 5.



Gambar 5. Segmentation Baris Braille.

Gambar *segmentation* baris braille akan dipotong menjadi 34 bagian sama rata dengan ukuran maksimal 2 titik braille sehingga setiap gambar merupakan kesatuan sel braille yang akan diklasifikasikan menjadi alfabet. Tahap tersebut akan diulangi sampai semua *segmentation* baris braille menjadi gambar *segmentation* sel Braille. Ukuran gambar *segmentation* sel Braille akan diresize menjadi 96x64 px untuk memudahkan gambar untuk diproses dalam tahap *classification*.



Gambar 6. Segmentation Sel Braille.

3.3 Classification

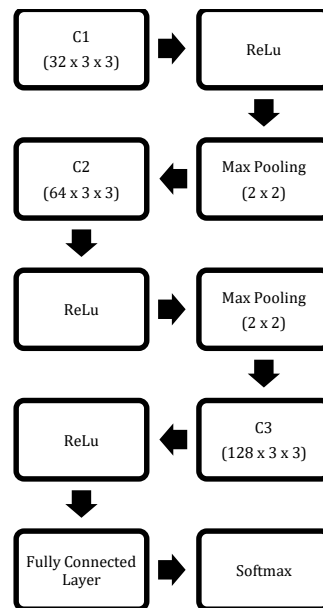
Dalam penelitian ini proses *classification* menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur yang mengadopsi dari penelitian Huang et al. [15]. Gambar input merupakan gambar hasil dari *segmentation* sel dengan ukuran 96x64x1 dalam skala hitam putih. Layer C1 memiliki kernel ukuran 3x3 sebanyak 32 dengan *activation ReLu* dan *max pooling* dengan ukuran 2x2. Selanjutnya Layer C2 memiliki ukuran kernel 3x3 sebanyak 64 dengan *activation ReLu* dan *max pooling* dengan ukuran 2x2. Layer C3 memiliki ukuran kernel 3x3 sebanyak 128 dengan *activation ReLu*. Terakhir ditambahkan *fully connected layer*

dengan 26 *output* untuk mengelompokkan dalam 26 kelas alfabet dengan *activation Softmax*. Gambar 7 menunjukkan arsitektur CNN yang digunakan.

Aktivasi ReLu diperlukan untuk mempercepat proses komputasi dengan mengaplikasikan fungsi pada Rumus 1. *Aktivasi ReLu* menjadikan nilai *thresholding* yang kurang dari nol menjadi nol [16]. Selain itu *Aktivasi ReLu* juga digunakan untuk mengurangi *overfitting* pada proses *classification*.

$$f(x) = \max(0, x) \quad (1)$$

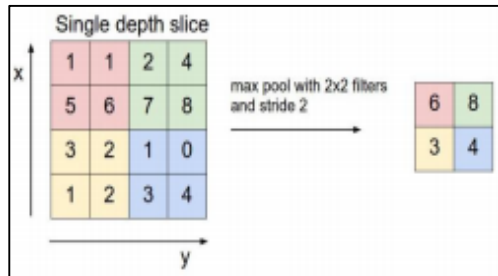
Pooling Layer memiliki filter berukuran 2x2 dengan proses bergeser keseluruhan bagian dari *feature map*. Penelitian ini menggunakan *pooling layer max pooling* dimana *max pooling* mengambil nilai tertinggi dari setiap area yang dilalui. *Max pooling* bertujuan untuk mengurangi ukuran gambar hingga 75% untuk mempercepat proses komputasi [17].



Gambar 7. Struktur CNN

Fully Connected Layer memiliki fungsi untuk melakukan transformasi

terhadap dimensi data agar dapat diklasifikasikan secara linear [18]. *Neuron convolution layer* akan dirubah ke data satu dimensi sebelum dimasukkan ke *fully connected layer*. *Fully connected layer* digunakan pada akhir jaringan karena data akan kehilangan informasi spasial yang tidak *reversibel*.



Gambar 8. Proses *Max Pooling*

Softmax merupakan suatu *activation* yang digunakan untuk mengklasifikasi lebih dari dua kelas target. *Softmax* menginterpretasikan probabilitas dengan lebih baik dan dapat menghitung probabilitas dari semua kelas yang ada [19]. *Softmax* akan merubah nilai vektor menjadi bernilai nol sampai satu dan apabila nilai pada kelas tersebut dijumlahkan bernilai satu.

4. Hasil dan Pembahasan

Komputer dengan *processor core i5*, *memory* 8 GB, dan GPU 4 GB digunakan dalam proses training dan testing dalam penelitian ini. *Library trainNetwork* dari aplikasi Matlab digunakan untuk proses training dan *library classify* untuk proses testing.

Dataset berupa gambar hasil segmentasi sel braille dengan total 4160 gambar. Keseluruhan gambar terbagi menjadi 2 bagian 3900 data training dan 260 data testing untuk mengetahui tingkat akurasi.

Data dalam proses training penelitian ini dikelompokkan menjadi 3 kelompok dan setiap kelompok akan

dibagi menjadi dua bagian yaitu 80% data training dan 20% data validasi. Kelompok pertama data yang digunakan sebanyak 60 data untuk setiap kelas yang berjumlah 1560 data dengan 1248 sebagai data training dan 312 sebagai data validasi. Kelompok kedua data yang digunakan berjumlah 100 data setiap kelas yang berjumlah 2600 data dengan 2080 sebagai data training dan 520 sebagai data validasi. Kelompok terakhir menggunakan 150 pada setiap kelas yang berjumlah 3900 data dengan 3120 sebagai data training dan 780 sebagai data validasi. Proses training dilakukan sebanyak 4 kali pada setiap kelompok, dengan masing-masing menggunakan 5 *epoch*, 10 *epoch*, 25 *epoch*, dan 50 *epoch*.

Proses training dilakukan terhadap kelompok pertama dengan 4 *epoch* berbeda. Akurasi training yang didapatkan secara berurutan dari *epoch* terkecil bernilai 0.9154, 0.9423, 0.9462, dan 0.9577 dengan nilai loss 0.3835, 0.2746, 0.2209, dan 0.2365. Nilai tersebut sudah baik namun dalam implementasi dengan data testing menghasilkan nilai 0.7885, 0.7038, 0.65, dan 0.6846 nilai tersebut dirasa masih kurang baik karena masih memiliki perbedaan yang cukup besar antara nilai akurasi training dan nilai akurasi testing.

Kelompok data kedua juga ditraining dengan 4 *epoch* dan arsitektur yang digunakan pada data pertama. Hasil akurasi training yang didapat secara berurutan bernilai 0.8212, 0.8423, 0.8269, dan 0.8173 dengan nilai loss 0.5038, 0.4806, 0.4699, dan 0.3615 dibandingkan dengan kelompok pertama, nilai akurasi dan loss pada kelompok kedua menurun. Namun nilai testing yang didapat meningkat menjadi 0.8577, 0.8577, 0.8269, dan 0.8962.

Terakhir kelompok data ketiga ditraining dengan cara yang sama seperti kedua kelompok data sebelumnya. Hasil yang didapat mengalami peningkatan yang sangat baik dengan nilai akurasi secara berurutan bernilai 0.9987, 0.9987, 0.9962, dan 0.9987 dengan nilai loss yang juga mengalami peningkatan menjadi bernilai 0.0046, 0.0232, 0.0102, dan 0.0209. dalam proses testing juga mengalami peningkatan dengan hasil akurasi bernilai 0.9962, 0.9962, 0.9923, dan 0.9923.

Hasil dari proses training mengalami penurunan dan peningkatan pada setiap kelompok data yang digunakan. Dimana nilai akurasi tertinggi didapatkan dengan menggunakan 150 data dan nilai akurasi terendah dengan menggunakan 100 data. Hal yang serupa juga terjadi pada hasil *loss* yang didapatkan pada setiap kelompok data yang digunakan.

Dalam proses tesing nilai akurasi pada setiap kelompok data mengalami peningkatan. Nilai akurasi terbaik pada kelompok pertama sebesar 0.7885 pada training dengan 5 epoch, dan kesalahan pengenalan 55 gambar dari 260 gambar testing.

Tabel 2. Hasil Pengenalan Terbaik Pada Setiap Kelompok Data

Kelas	60 Data		100 Data		150 Data	
	5		50		10	
	Acc	Err	Acc	Err	Acc	Err
A	5	5	10	0	10	0
B	5	5	7	3	10	0
C	5	5	10	0	10	0
D	8	2	9	1	10	0
E	5	5	9	1	10	0
F	8	2	6	4	10	0
G	10	0	8	2	10	0
H	4	6	9	1	10	0
I	7	3	9	1	10	0
J	4	6	8	2	10	0
K	8	2	8	2	10	0
L	9	1	10	0	10	0
M	9	1	10	0	10	0

N	9	1	6	4	10	0
O	10	0	10	0	10	0
P	9	1	8	2	10	0
Q	7	3	10	0	9	1
R	8	2	10	0	10	0
S	10	0	10	0	10	0
T	10	0	10	0	10	0
U	9	1	10	0	10	0
V	10	0	10	0	10	0
W	7	3	7	3	10	0
X	10	0	10	0	10	0
Y	9	1	9	1	10	0
Z	10	0	10	0	10	0

Pada kelompok data kedua nilai akurasi terbaik bernilai 0.8962 didapatkan dengan menggunakan 50 epoch, dan kesalahan dalam pengenalan sebanyak 27 gambar dari 260 gambar testing. Kelompok data ketiga mendapatkan nilai akurasi terbaik yaitu 0.9962 dengan 10 epoch, dan kesalahan pengenalan sebanyak 1 gambar dari 260 gambar testing. Detail kesalahan pengenalan dapat dilihat pada Tabel 2.

5. Penutup

Berdasarkan penelitian yang telah dilakukan dengan beberapa kelompok data, arsitektur yang telah diadopsi dari penelitian Huang et al.[15] dapat disesuaikan dengan kebutuhan pemrosesan gambar. Tingkat keberhasilan dapat dibuktikan dengan pencapaian akurasi testing mencapai 0.9662 atau 99.62% dengan nilai loss 0.0232. Dari percobaan dalam penelitian ini model terbaik didapatkan dengan menggunakan 150 data dan 10 epoch, kesalahan pengenalan hanya terjadi pada kelas huruf Q.

6. Daftar Pustaka

- [1] J. Subur, T. A. Sardjono, and R. Mardiyanto, "Braille character recognition using find contour method," *Proceedings - 5th International Conference on Electrical Engineering and Informatics: Bridging the*

- Knowledge between Academic, Industry, and Community, ICEEI 2015*, pp. 699–703, 2015.
- [2] J. Mennens, L. van Tichelen, G. François, and J. J. Engelen, “Optical Recognition of Braille Writing Using Standard Equipment,” *IEEE Transactions on Rehabilitation Engineering*, vol. 2, no. 4, pp. 207–212, 1994.
- [3] M. Hanumanthappa and V. V. Murthy, “Optical Braille recognition and its correspondence in the conversion of Braille script to text - A literature review,” *2016 International Conference on Computation System and Information Technology for Sustainable Solutions, CSITSS 2016*, pp. 297–301, 2016.
- [4] K. Smelyakov, A. Chupryna, D. Yeremenko, A. Sakhon, and V. Polezhai, “Braille Character Recognition Based on Neural Networks,” *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018*, pp. 509–513, 2018.
- [5] R. Indah Purwatiningsih, “Program studi teknik informatika fakultas teknik universitas bengkulu 2014,” pp. 1–14, 2014.
- [6] N. D. S. M. K. De Silva and S. Vasanthapriyan, “Optical braille recognition platform for Sinhala,” *18th International Conference on Advances in ICT for Emerging Regions, ICTer 2018 - Proceedings*, pp. 7–12, 2019.
- [7] W. K. I. L. I. Perera, T. D. S. H., & Wanniarachchi, “Optical Braille Recognition Based on Histogram of Oriented Gradient Features and Support-Vector Machine,” *International Journal of Engineering Science*, vol. 8, no. 10, pp. 19192–19195, 2018.
- [8] E. Arias-Castro and D. L. Donoho, “Does median filtering truly preserve edges better than linear filtering?,” *Annals of Statistics*, vol. 37, no. 3, pp. 1172–1206, 2009.
- [9] F. A. Hermawati and Tigor, “Perbaikan Efek Ketidakteraturan Pada Citra Dengan Histogram Statistik,” *Konvergensi*, vol. 11, no. 2, pp. 1–11, 2015.
- [10] Dwi Harini Sulistyawati and H. S. Utomo, “Perbaikan Citra Dengan Noise Missing Block Menggunakan Implementasi Algoritma Projection Onto Convex Sets (POCS),” *Konvergensi*, vol. 12, no. 2, pp. 11–19, 2016.
- [11] B. A. Pratama and F. A. Hermawati, “Sistem Pengenalan Manusia Melalui Keunikan Fisiologis Selaput Pelangi Mata Dengan Menggunakan Filter Log Gabor,” *KONVERGENSI*, vol. 13, no. 1, pp. 11–17, 2017.
- [12] N. S. Fathonah, A. Y. Pratama, and F. A. Hermawati, “Kalkulator Saintifik Berbasis Kamera,” *Konvergensi*, vol. 15, no. 2, pp. 107–112, 2019.
- [13] A. K. W. Hapantenda, A. Januantoro, and I. Listiowarni, “Studi Independen Komparasi Segmentasi Sel darah Putih Menggunakan Ruang Warna HSV Dengan CIE-L*a*b,” *Konvergensi*, vol. 15, no. 2, pp. 97–104, 2019.
- [14] Y. Zhang and L. Wu, “Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach,” *Entropy*, vol. 13, no. 4, pp. 841–859, 2011.
- [15] R. Huang, B. Liu, W. Su, and H. Lin, “Research on braille music

- recognition based on convolutional neural network,” *ICNC-FSKD 2018 - 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 837–843, 2018.
- [16] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” no. 1, pp. 2–8, 2018.
- [17] A. Giusti, D. C. Cireşan, J. Masci, L. M. Gambardella, and J. Schmidhuber, “Fast image scanning with deep max-pooling convolutional neural networks,” *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, pp. 4034–4038, 2013.
- [18] W. Ma and J. Lu, “An Equivalence of Fully Connected Layer and Convolutional Layer,” no. 3, pp. 1–9, 2017.
- [19] A. S. Rawat, J. Chen, F. Yu, A. T. Suresh, and S. Kumar, “Sampled softmax with random fourier features,” *Advances in Neural Information Processing Systems*, vol. 32, no. 2, 2019.