

Pengenalan Gambar Braille Menggunakan *Depthwise Separable Convolutional Neural Network*

Braille Image Recognition Based on Depthwise Separable Convolutional Neural Network

Kevin Maulana Afriyanto¹, Abas Setiawan^{*2}

¹Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

²Teknik Informatika, Universitas Negeri, Semarang, Indonesia

E-mail: ¹111201811523@mhs.dinus.ac.id, ^{2*}abas.setiawan@dsn.dinus.ac.id

**Corresponding author*

Received 23 February 2025; Revised 15 April 2025; Accepted 30 April 2025

Abstrak - Huruf Braille adalah huruf yang memiliki sistem tulisan yang menggunakan indra peraba untuk menggunakannya dan umum digunakan oleh penyandang tunanetra. Pada orang normal yang sering berinteraksi dengan para penyandang tunanetra terkadang mengalami kesulitan dalam menggunakan huruf braille, sehingga diperlukan suatu pendekatan teknologi sistem cerdas untuk dapat membantu mengenali huruf braille. Penelitian ini bertujuan untuk menggunakan metode *depthwise separable convolutional neural network* untuk melakukan pengenalan terhadap gambar braille. Dataset yang digunakan memiliki ukuran 28x28 dengan total dataset sebanyak 1560 citra. Arsitektur yang digunakan adalah input layer, yaitu sebesar 28x28x3 dan layer *depthwise separable convolution* pertama, kedua, dan ketiga menggunakan masing masing 3x3x128, 3x3x256, dan 2x2x512 dengan fungsi aktivasi *rectified linear unit* (ReLU) dan setiap konvolusi diikuti dengan lapisan-lapisan *maxpooling* dan *batch normalization*. Setelah itu, dilanjutkan dengan dua *fully connected layer* dengan jumlah neuron 256 dan 128 serta diakhiri dengan *output layer* sejumlah 26 neuron. Arsitektur tersebut menghasilkan model terbaik dengan akurasi sebesar 98.40%. Model tersebut diterapkan pada aplikasi berbasis website. Hasil temuan penelitian ini diharapkan dapat membantu masyarakat dalam mengenali huruf braille dengan mudah.

Kata kunci: tunanetra, braille, depthwise separable convolution, model, akurasi

Abstract - Braille is a letter with a writing system that uses the sense of touch. Visually impaired people commonly use it. People who often interact with blind people sometimes have difficulty using Braille, so an intelligent system technology approach is needed to help recognize Braille. This study aims to use the depthwise separable convolutional neural network method to recognize Braille images. The dataset used is 28x28, with a total dataset of 1560 images. The architecture used is the input layer, which is 28x28x3, and the first, second, and third depthwise separable convolution layers use 3x3x128, 3x3x256, and 2x2x512 respectively, with the rectified linear unit (ReLU) activation function and each convolution is followed by maxpooling layers and batch normalization. After that, it continued with two fully connected layers with the number of neurons 256 and 128 and ended with an output layer of 26 neurons. The architecture produces the best model with an accuracy of 98.40%. The model is applied to a website-based application. The findings of this study are expected to help people recognize braille easily.

Keywords: visually impaired people, braille, depthwise separable convolution, model, accuracy

1. PENDAHULUAN

Membaca dan menulis merupakan kegiatan seseorang untuk memperoleh ilmu dan pengetahuan yang kelak akan digunakan untuk kesejahteraan negara ataupun dunia dan merupakan kegiatan yang menciptakan peradaban modern seperti saat ini. Namun, kegiatan tersebut akan sangat sulit bagi orang yang memiliki kebutuhan khusus seperti tunanetra atau memiliki gangguan pada penglihatan. Orang yang memiliki gangguan pada penglihatan juga masih satu kesatuan dengan masyarakat biasa yang tidak bisa diabaikan [1].

Secara global, World Of Organization (WHO) memperkirakan bahwa ada 2,2 miliar orang yang mengalami gangguan penglihatan jarak jauh atau dekat dan setidaknya ada 1 miliar orang yang mengalami gangguan penglihatan dari sedang hingga berat atau kebutaan karena kelainan refraksi (88,4 juta), katarak (94 juta), glaukoma (7,7 juta), corneal opacities (4,2 juta), diabetic retinopathy (3,9 juta), trakoma (2 juta), serta gangguan penglihatan jarak dekat yang disebabkan oleh presbiopia yang tidak tertangani (826 juta) [2]. Meskipun orang-orang tersebut diberi keterbatasan pada penglihatan namun orang-orang tersebut tetap berusaha keras untuk berkontribusi semaksimal mungkin kepada masyarakat. Meskipun demikian, masalah yang dihadapi ialah bagaimana penyandang tunanetra menyampaikan ide dan aspirasinya ke masyarakat dikarenakan penyandang tunanetra menggunakan penulisan yang berbeda yaitu menggunakan huruf braille yang tidak banyak dipahami oleh masyarakat umum. Huruf braille ini sendiri ditemukan oleh guru dari Prancis, Louis Braille, pada tahun 1824. Huruf braille merupakan sistem representasi kode binary yang digunakan orang yang memiliki gangguan penglihatan yang tulisan ini tidak menggunakan metode menulis dan mencetak yang biasa, huruf braille memiliki efek timbul di 6 titik yang dapat dirasakan oleh ujung jari pada huruf braille [3].

Kebanyakan dari penyandang tunanetra menggunakan sistem braille untuk membaca, namun orang yang memiliki penglihatan normal tidak paham bagaimana membaca huruf braille. Orang yang memiliki penglihatan normal juga perlu paham bagaimana membaca huruf braille agar transfer informasi antara penyandang tunanetra dan orang yang memiliki penglihatan normal tetap terjalin. Oleh karena itu diciptakan sistem *assistive technology* yang diharapkan dapat menyelesaikan masalah yang dihadapi ini. Sistem *assistive technology* dapat digunakan untuk menerjemahkan huruf braille menjadi huruf alphabet ataupun juga suara yang digunakan untuk menunjang siswa yang memiliki keterbatasan pada penglihatan di sekolah ataupun juga di universitas. Namun, masalah ini tidak sepenuhnya berada pada penyandang tunanetra, orang yang memiliki penglihatan normal juga menghadapi permasalahan yang serupa, orang yang memiliki penglihatan normal memiliki kesulitan untuk memahami buku yang dicetak menggunakan huruf braille yang memungkinkan buku tersebut memiliki informasi yang sangat penting di bidang matematika ataupun kimia sebagai contoh [4].

Optical Character Recognition (OCR) merupakan metode menangkap gambar tulisan tangan ataupun cetakan menjadi natural language characters. Namun sampai saat ini belum ada OCR yang secara efektif mendeteksi huruf braille. Lalu teknologi Optical Braille Recognition (OBR) hadir untuk menyelesaikan masalah tersebut, perangkat lunak penerjemah yang ditawarkan oleh Neovision diciptakan untuk menterjemahkan huruf braille menjadi sebuah text yang dapat dibaca, namun alat tersebut belum dapat digunakan oleh semua kalangan dikarenakan harganya yang mahal, ketinggalan zaman, dan kurang efisien[5].

Penelitian mengenai identifikasi huruf pada citra braille telah banyak dilakukan dengan menggunakan metode yang berbeda – beda. AlSalman *et al* menggunakan metode DCNN (Deep Convolutional Neural Network) untuk merekognisi huruf pada citra braille. Dengan menggunakan metode DCNN diperoleh akurasi sebesar 99.28% [4]. Lalu pada penelitian selanjutnya yang dilakukan oleh Bi Min Hsu yang melakukan penelitian rekognisi huruf pada citra braille menggunakan metode CNN (Convolutional Neural Network) dan juga metode RSCA (Ratio Character Segmentation Algorithm) yang digunakan untuk mendapatkan sudut yang terbaik untuk menangkap citra pada huruf braille. Pada penelitian ini diperoleh akurasi sebesar 98.73% setelah dilakukan training menggunakan model CNN [5].

Berdasarkan penelitian sebelumnya, rumusan masalah yang dapat diajukan adalah bagaimana membangun sistem machine learning sebagai teknologi bantu yang mampu mengenali huruf Braille agar dapat dibaca oleh orang normal secara akurat. Oleh karena itu, penelitian ini bertujuan untuk melakukan pengenalan huruf pada citra braille menggunakan algoritma CNN dan mengganti convolutional layer tradisional dengan convolutional layer depthwise separable untuk menghasilkan hasil akurasi yang lebih baik.

2. METODE PENELITIAN

2.1 Prosedur Pengambilan Data

Dalam penelitian ini, data yang digunakan akan dibagi menjadi dua yaitu training dataset dan testing dataset yang digunakan untuk proses pengujian. Data-data ini diperoleh dari situs website kaggle yang berjudul Braille Character Dataset yang dibuat oleh Shankar Kumar (<https://www.kaggle.com/codebreaker619/braille-image-classifier-using-neural-networks/data>). Gambar 1 merupakan salah satu contoh sampel citra braille dari dataset yang digunakan.



Gambar 1 Contoh dari citra braille

Pada dataset ini data sudah dilakukan augmentasi yaitu dengan cara menggeser tinggi lebar, melakukan rotasi, dan mengubah kecerahan pada citra dengan 20 nilai augmentasi yang berbeda yang dilakukan pada 26 citra karakter braille sehingga didapatkan citra total 1560 citra. Pemberian porsi dataset yang digunakan pada project ini adalah sebanyak 80% digunakan untuk training set yang berjumlah 1248 dan juga testing set sebanyak 20% yang berjumlah 312.

2.2 Metode dan Proses Training CNN

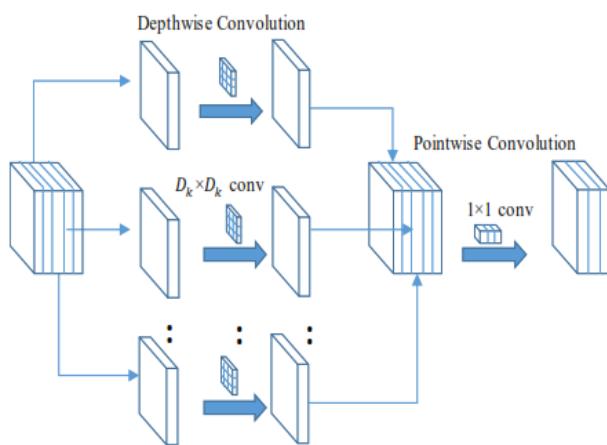
Metode yang digunakan pada penelitian ini mengusulkan metode Convolutional Neural Network (CNN) yang digunakan untuk menklasifikasi huruf braille seperti yang terlihat pada Gambar 2. Input yang digunakan pada penelitian ini menggunakan citra yang memiliki format.jpg berwarna yang memiliki channel RGB atau *red, green, blue*. Arsitektur CNN terdiri atas tiga kombinasi dari tiga tipe lapisan yaitu *convolutional, pooling, dan fully connected*. Output dari lapisan convolutional dan lapisan maxpooling di kelompokkan menjadi bidang 2D yang dinamakan pemetaan fitur. Output lapisan terakhir dari CNN digunakan sebagai input pada bagian *fully connected network* yang dimana dipanggil lapisan classification. Pada lapisan classification, jumlah fitur yang diinginkan digunakan sebagai input dengan sehubungan kepada dimensi bobot matiks saraf akhir. Namun pada saat ini, ada beberapa teknik baru selain *fully connected network* contohnya *average pooling* dan *global average pooling*. Skor dari masing-masing kelas dihitung di lapisan klasifikasi teratas menggunakan lapisan soft-max. Berdasarkan skor tertinggi, classifier memberikan output untuk kelas yang sesuai [6].



Gambar 2 Arsitektur dari CNN yang diusulkan.

Pada penelitian ini layer konvolusi yang digunakan adalah *depthwise separable convolution* untuk melakukan proses konvolusi. Depthwise separable convolution, pada

umumnya dipanggil dengan sebutan “separable convolution/konvolusi terpisah” pada framework deep learning seperti contohnya Tensorflow dan Keras, terdiri atas depthwise convolution, yaitu konvolusi yang dilakukan secara independen pada masing masing channel input, diikuti dengan pointwise convolution, yaitu 1×1 konvolusi, memproyeksikan channel output dengan depthwise convolution menuju ruang channel baru [7]. Depthwise separable convolution, pada umumnya dipanggil dengan sebutan “*separable convolution* atau konvolusi terpisah” pada framework deep learning seperti contohnya Tensorflow dan Keras, terdiri atas depthwise convolution, yaitu konvolusi yang dilakukan secara independen pada masing masing channel input, diikuti dengan pointwise convolution, yaitu 1×1 konvolusi, memproyeksikan channel output dengan depthwise convolution menuju ruang channel baru. Hal ini membuat berbeda dengan spatial separable convolution, yang dimana pada umumnya dipanggil dengan “*separable convolution*” oleh komunitas pengolahan citra [7,8]. Gambar 3 merupakan ilustrasi dari proses *depthwise separable convolution*.



Gambar 3 Proses *depthwise convolution* [8]

Pada setiap layer konvolusi akan digunakan aktivasi ReLU pada parameter lapisan konvolusi dan peneliti menggunakan aktivasi ReLU sebagai fungsi aktivasi pada fully connected layer. Aktivasi ReLU ini mengubah nilai dari matriks yang memiliki negatif menjadi 0 dan menjaga nilai matriks apabila nilai tersebut positif dengan rumus yang digunakan adalah $y = \max(x, 0)$ [9].

Setelah melakukan proses konvolusi maka selanjutnya akan dilakukan proses *pooling*. Pada proses pooling terdapat dua macam proses *pooling*, yaitu *maxpooling* dan *average pooling*. *Maxpooling* merupakan metode yang digunakan untuk mencari nilai yang terbesar pada fitur matriks lalu menyimpan nilai tersebut sebagai transformasi dari fitur matriks tersebut. Sedangkan *average-pooling* merupakan metode yang digunakan untuk mencari nilai rata-rata dari ukuran pooling yang digunakan, lalu menyimpan nilai tersebut sebagai transformasi dari fitur matriks tersebut. Pada penelitian ini peneliti akan menggunakan metode max-pooling [10].

Selanjutnya setelah melalui lapisan-lapisan konvolusi selanjutnya akan dilanjutkan pada tahap *fully connected layer*, umumnya setelah melakukan konvolusi akan menggunakan metode flatten untuk mengubah ukuran matriks menjadi ukuran satu dimensi, namun pada penelitian ini peneliti akan menggunakan metode Global Max Pooling. Alasan dari peneliti menggunakan metode tersebut dikarenakan flatten layer akan mengambil tensor dalam bentuk apa saja dan mengubahnya menjadi menjadi tensor satu dimensi dan tetap menyimpan semua nilai pada tensor. Hal ini memiliki resiko terjadinya *overfitting* pada dataset training. Kelebihan yang dimiliki oleh Global Max Pooling daripada Flatten adalah lebih asli kepada struktur konvolusi dengan memaksa korespondensi antara feature map dan kategori. Dengan demikian feature map dapat ditafsirkan sebagai kategori confidence maps. Dan kelebihan lainnya adalah tidak

diperlukannya parameter untuk mengoptimasi pada global max pooling sehingga overfitting dapat dihindari pada layer ini [11].

Layer selanjutnya adalah menggunakan layer dense, layer ini merupakan layer yang sering ditemui pada neural network, tujuan dari dense layer ini adalah mengumpulkan semua output dari lapisan sebelumnya dan digunakan untuk menunjang proses klasifikasi, pada penelitian menggunakan dua layer dense yang masing masing memiliki 256 dan 128 neuron. Lalu pada output layer digunakan 26 neuron sesuai kelas yang ingin diklasifikasi yaitu huruf alfabet. Dikarenakan penelitian ini merupakan multi-class classification fungsi aktivasi yang digunakan adalah softmax, tujuan digunakan fungsi tersebut dikarenakan aktivasi softmax hanya memberikan nilai 1 pada neuron yang memiliki nilai terbesar.

Selanjutnya dilakukan proses Backpropagation yang dimana melakukan perbandingan antara hasil output dengan hasil prediksi dengan menggunakan Loss Function dan memperbarui nilai bobot dan bias yang didapatkan setelah proses forward pass. Loss Function sendiri berfungsi untuk mengukur apakah hasil dari Training CNN bagus atau tidak. Dan dikarenakan penelitian yang dilakukan merupakan klasifikasi maka loss function yang digunakan adalah Categorical Crossentropy. Dan digunakan juga optimizer pada proses backpropagation untuk mengubah atribut seperti bobot dan learning rate untuk memperkecil nilai loss, pada penelitian peneliti menggunakan optimizer Adam.

Pada algoritma backpropagation terdapat tiga fungsi yaitu pertama DeltaCalculation yang berfungsi menghitung error antara output layer dan hidden layer. Fungsi digunakan pada proses memperbarui nilai bobot. Selanjutnya fungsi UpdateInputWeights yang berfungsi memperbarui bobot yang terhubung dengan input layer dan hidden layer. Terakhir yaitu fungsi UpdateHiddenWeights yang berfungsi memperbarui bobot yang terhubung dengan hidden layer dan output layer [12].

3. HASIL DAN PEMBAHASAN

3.1 Skenario Percobaan

Dalam membangun arsitektur CNN dilakukan berbagai skenario dengan mengubah layer konvolusi yang berbeda dan mengubah nilai dari berbagai layer yang digunakan seperti learning rate, dan dropout. Hal ini dilakukan dengan tujuan mendapatkan nilai akurasi yang terbaik agar model dapat melakukan klasifikasi secara akurat. Tabel 1 merupakan skenario percobaan yang dilakukan oleh peneliti untuk mendapatkan nilai akurasi yang terbaik dengan mengubah jenis dan nilai pada parameter layer. Pada skenario percobaan jumlah epoch yang digunakan adalah 100.

Lapisan yang digunakan pada skenario 1 adalah menggunakan tiga layer konvolusi tradisional yang memiliki input sebesar 28x28x3 dengan lapisan pertama menggunakan filter sebanyak 128 dengan ukuran kernel 3x3, fungsi aktivasi yang digunakan pada skenario ini adalah ReLu. Setiap lapisan konvolusi yang digunakan akan dihubungkan oleh lapisan maxpooling dan batch normalization. Pada skenario layer konvolusi menggunakan dua jenis versi konvolusi yaitu layer konvolusi biasa dan layer separable konvolusi sebagai perbandingan. Layer konvolusi kedua menggunakan filter sebanyak 256 dengan ukuran kernel 3x3 yang dihubungkan layer maxpooling dan batch normalization. Selanjutnya pada layer konvolusi ketiga menggunakan filter sebanyak 512 dengan ukuran kernel 2x2 yang dihubungkan oleh layer maxpooling dan batch normalization.

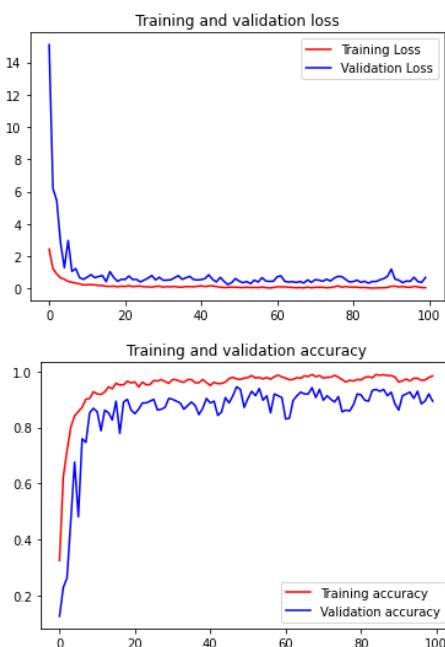
Berikutnya, lapisan sebelumnya dihubungkan dengan fully connected layer melalui lapisan flatten. Dua fully connected layer tersebut menggunakan layer dense sebanyak 256 dan 128. Yang akhirnya pada layer output menggunakan 26 neuron dengan aktivasi softmax. Konfigurasi yang digunakan pada penelitian ini adalah menggunakan optimizer Adam dengan nilai learning rate 0.001. Untuk fungsi loss yang digunakan adalah categorical crossentropy dikarenakan kasus ini merupakan multiclass classification. Proses pelatihan dilakukan sebanyak

100 epoch dengan pengaturan callback, model checkpoint, dan reduce learning rate. Untuk skenario percobaan lain diperlihatkan pada Tabel 1.

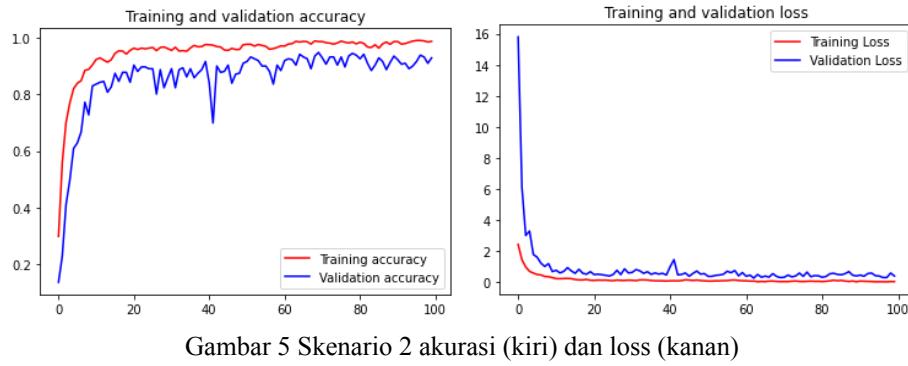
Tabel 1 Skenario Percobaan

Skenario Model	Perbedaan
2	Dropout(0.2)
3	Dropout(0.3)
4	Dropout(0.4)
5	Menggunakan depthwise separable convolution pada layer konvolusi
6	Menggunakan Global Max Pooling pada fully connected layer
7	Menggunakan adam dengan nilai 0.0001 pada optimizer
8	Menggunakan adam dengan nilai 0.001 pada optimizer
9	Menggunakan adam dengan nilai 0.01 pada optimizer
10	Menggunakan adam dengan fungsi reduce learning rate on plateau pada optimizer
11	Menambahkan kernel regularizer pada layer dense

Sebanyak 11 skenario percobaan dilakukan demi menemukan model yang berhasil melakukan klasifikasi dengan baik. Pada percobaan 1 dengan menggunakan proses konvolusi biasa didapat nilai akurasi masih dibawah yang diharapkan (98%) akan tetapi sudah cukup stabil. Meski demikian, perbedaan antara nilai akurasi dan validasi akurasi masih cukup besar sehingga model masih overfitting yang ditunjukkan di Gambar 4. Pada skenario 2 dengan menggunakan dropout dengan nilai 0.2 didapat nilai akhir dari validasi akurasi sudah mencapai 92% lebih baik dari sebelumnya namun pada epoch ke 42 nilai validasi akurasi drop dari 83% ke 69% diperlihatkan pada Gambar 5.

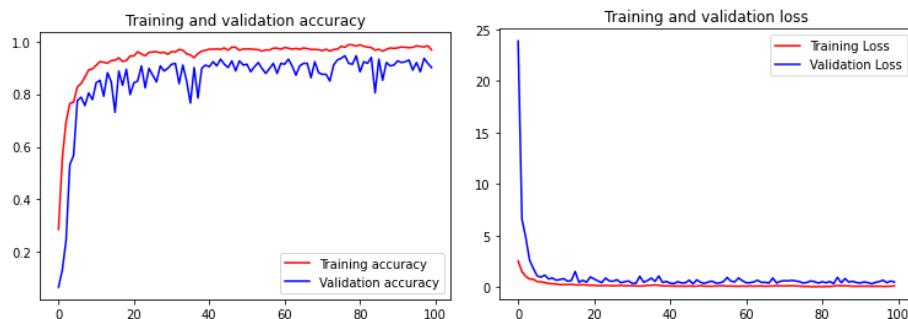


Gambar 4 Skenario 1 akurasi (kiri) dan loss (kanan)

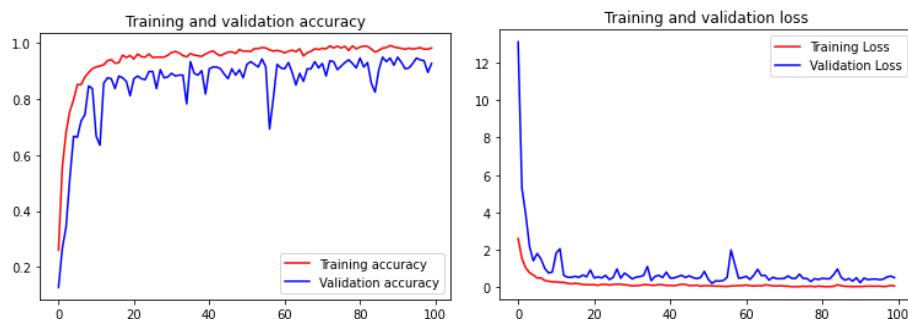


Gambar 5 Skenario 2 akurasi (kiri) dan loss (kanan)

Pada skenario 3 dengan menggunakan dropout dengan nilai 0.3 didapat nilai validasi akurasi 90.06% lebih sedikit dengan sebelumnya dan lebih tidak stabil dan sering ditemui kenaikan dan penurunan yang signifikan namun nilai loss sudah cukup stabil (Gambar 6). Pada gambar 7 memperlihatkan skenario percobaan 4 dengan menggunakan dropout dengan nilai 0.4 didapat nilai akurasi lebih baik dari sebelumnya yaitu dengan nilai 92.63% namun ditemukan pada epoch 57 kemudian dilanjutkan dengan penurunan yang sangat signifikan.

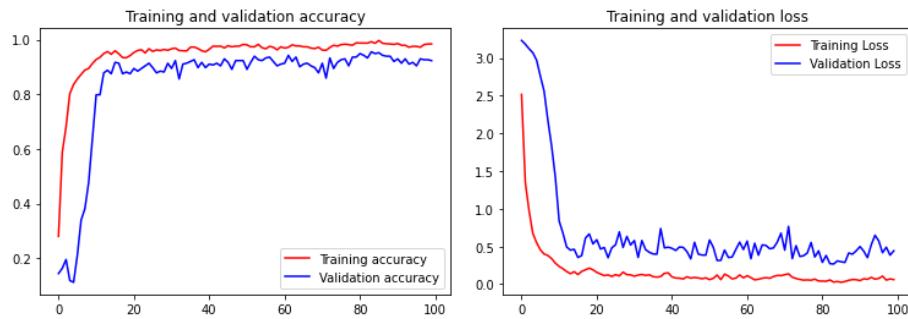


Gambar 6 Skenario 3 akurasi (kiri) dan loss (kanan)

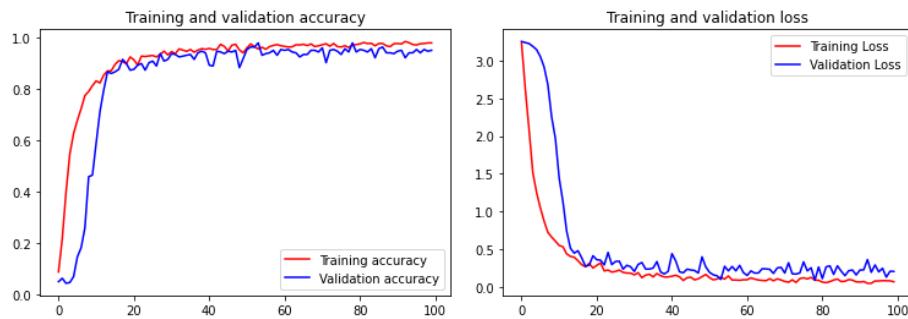


Gambar 7 Skenario 4 akurasi (kiri) dan loss (kanan)

Gambar 8 merupakan hasil percobaan skenario 5 dengan menggunakan *depthwise separable convolution* pada lapisan konvolusi didapat perbedaan nilai akurasi yang lebih stabil dari sebelumnya namun nilai loss dan loss validasi terjadi perbedaan yang cukup signifikan. Skenario 6 dengan menggunakan global max pooling pada fully connected layer dengan hasil perbandingan antara nilai akurasi dan validasi yang sudah lebih stabil dan selisih dari keduanya sudah lebih kecil namun nilai akurasi validasi masih kurang dari 98% yang ditunjukkan di Gambar 9.

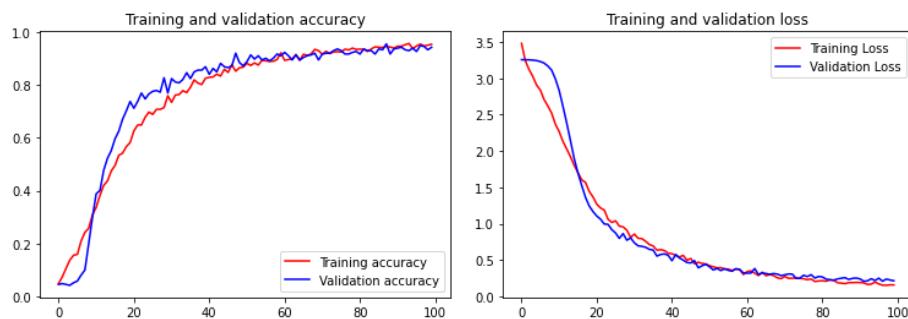


Gambar 8 Skenario 5 akurasi (kiri) dan loss (kanan)

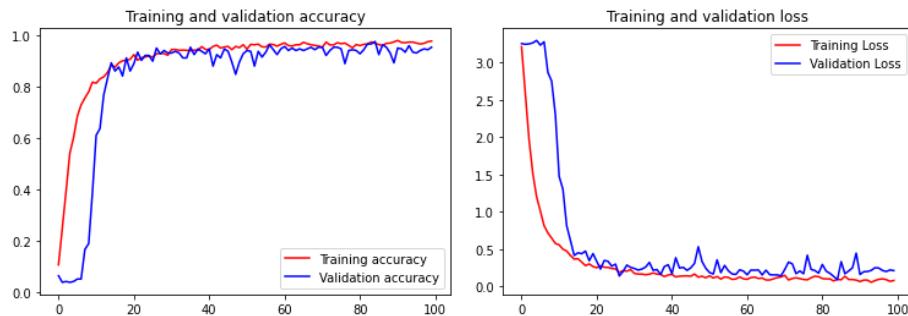


Gambar 9 Skenario 6 akurasi (kiri) dan loss (kanan)

Pada skenario 7 dengan menggunakan optimizer Adam dengan nilai 0.0001 didapat nilai akurasi sudah lebih stabil begitu juga dengan nilai loss namun nilai validasi akurasi yang didapatkan masih dibawah 98% (Gambar 10). Gambar 11 memperlihatkan perjalanan skenario 8 dengan menggunakan adam dengan nilai 0.001 pada optimizer didapatkan kenaikan grafik sedikit lebih tidak stabil dari sebelumnya namun nilai akhir yang dihasilkan jauh lebih tinggi yaitu 95.51% dibandingkan sebelumnya begitu juga dengan loss yang dihasilkan.

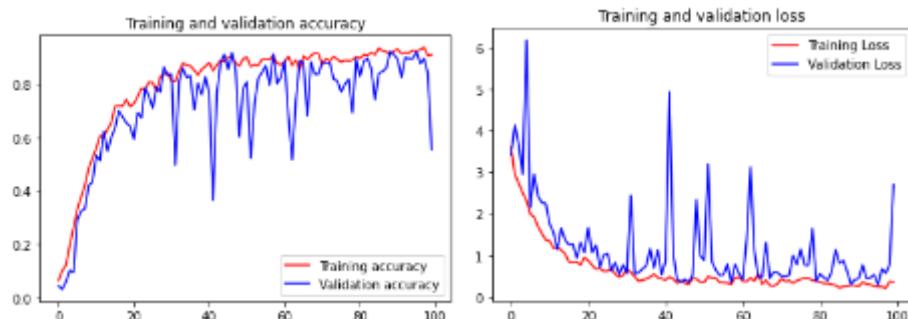


Gambar 10 Skenario 7 akurasi (kiri) dan loss (kanan)

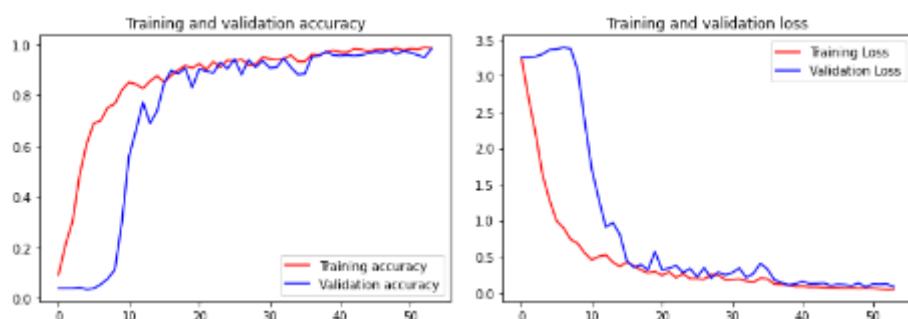


Gambar 11 Skenario 8 akurasi (kiri) dan loss (kanan)

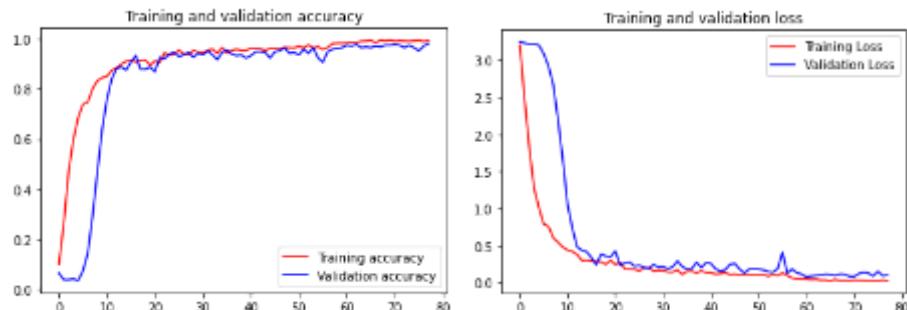
Pada skenario 9 yang ditunjukkan pada Gambar 12, menggunakan *learning rate* pada Adam dengan nilai 0.01 didapatkan kenaikan kurva sangat tidak stabil dan nilai akhir akurasi sangat berbanding jauh dengan nilai validasinya sehingga menyebabkan model tersebut fluktuatif. Pada skenario percobaan 10 dengan menggunakan fungsi *reduce learning rate on plateau* didapatkan nilai dari validasi akurasi sudah cukup tinggi yaitu 98.40% namun model baru bisa stabil pada epoch 17 (Gambar 13). Terakhir, Gambar 14 memperlihatkan hasil skenario 11 menggunakan kernel regularizer dengan nilai 0.00001 pada layer dense didapatkan model yang terbentuk sudah stabil dan validasi akurasi yang didapatkan sudah memenuhi harapan yaitu diatas 98.08%.



Gambar 12 Skenario 9 akurasi (kiri) dan loss (kanan)



Gambar 13 Skenario 10 akurasi (kiri) dan loss (kanan)



Gambar 14 Skenario 11 akurasi (kiri) dan loss (kanan)

Setelah melakukan sebeleas kali percobaan, didapatkan skenario percobaan pertama yang menggunakan konvolusi biasa dan flatten menghasilkan akurasi sebesar 89.42%. Skenario percobaan kedua menggunakan dropout dengan nilai 0.2 menghasilkan akurasi 92.95%. Skenario percobaan ketiga mengubah nilai dropout dengan nilai 0.3 menghasilkan akurasi 90.06%. Skenario percobaan keempat mengubah nilai dropout dengan nilai 0.4 menghasilkan akurasi 92.63%. Skenario percobaan kelima mengubah konvolusi biasa dengan menggunakan *depthwise separable konvolusi* menghasilkan akurasi 92.31%. Skenario keenam mengubah flatten layer dengan global max pooling menghasilkan akurasi 94.23%. Skenario ketujuh menggunakan optimizer adam dengan nilai learning rate sebesar 0.0001 menghasilkan akurasi 94.23%. Skenario kedelapan mengubah nilai learning rate sebesar 0.001 menghasilkan akurasi 88.46%. Skenario kesembilan mengubah nilai learning rate menjadi 0.01 menghasilkan akurasi 55.45%. Skenario kesepuluh menambahkan fungsi dengan *reduce learning rate* menghasilkan akurasi 98.40%. Skenario kesebelas dengan menambahkan parameter kernel_regularizer dengan nilai 0.00001 akurasi yang didapatkan adalah 98.08%. Keseluruhan hasil percobaan dijabarkan pada Tabel 2. Dari keseluruhan skenario yang diuji oleh peneliti dapat disimpulkan bahwa skenario percobaan kesepuluh memperoleh akurasi tertinggi, sehingga skenario 10 digunakan untuk model yang ditanam pada aplikasi berbasis web untuk klasifikasi huruf braille.

Tabel 2 Hasil Percobaan

Skenario Percobaan	Validasi Akurasi	Akurasi
1	89.42%	98.40%
2	92.95%	98.80%
3	90.06%	96.79%
4	92.63%	98.16%
5	92.31%	98.40%
6	94.87%	97.76%
7	94.23%	95.35%
8	95.51%	96.47%
9	55.45%	90.87%
10	98.40%	98.56%
11	98.08%	99.28%

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian klasifikasi huruf braille disimpulkan bahwa penggunaan Convolution Neural Network dengan menggunakan Depthwise Separable Convolutional Network mampu mengklasifikasikan gambar huruf barille dengan memberikan akurasi sebesar 98.40% dari skenario percobaan terbaik. Penelitian ini dapat dikembangkan agar proses klasifikasi yang diterapkan adalah melakukan klasifikasi menggunakan satu kalimat ataupun paragraf dari dokumen braille sehingga proses untuk melakukan proses penerjemahan dari huruf braille ke huruf alfabet lebih mudah. Pengembangan model algoritma yang lebih baik akan mampu untuk mewujudkan system yang dapat langsung digunakan oleh masyarakat luas untuk nantinya digunakan untuk menghilangkan jurang pemisah antara orang normal ataupun orang

tunanetra.

DAFTAR PUSTAKA

- [1] S. Padmavathi, M. K.S.S, S. Reddy .S, and M. D, "Conversion of Braille to Text in English, Hindi and Tamil Languages," *Int. J. Comput. Sci. Eng. Appl.*, vol. 3, no. 3, pp. 19–32, 2013, doi: 10.5121/ijcsa.2013.3303.
- [2] R. R. A. Bourne *et al.*, "Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: The Right to Sight: An analysis for the Global Burden of Disease Study," *Lancet Glob. Heal.*, vol. 9, no. 2, pp. e144–e160, 2021, doi: 10.1016/S2214-109X(20)30489-7.
- [3] K. Smelyakov, A. Chupryna, D. Yeremenko, A. Sakhon, and V. Polezhai, "Braille Character Recognition Based on Neural Networks," *Proc. 2018 IEEE 2nd Int. Conf. Data Stream Min. Process. DSMP 2018*, pp. 509–513, 2018, doi: 10.1109/DSMP.2018.8478615.
- [4] A. AlSalman, A. Gumaei, A. AlSalman, and S. Al-Hadhrami, "A Deep Learning-Based Recognition Approach for the Conversion of Multilingual Braille Images," *Comput. Mater. Contin.*, vol. 67, no. 3, pp. 3847–3864, 2021, doi: 10.32604/cmc.2021.015614.
- [5] B. M. Hsu, "Braille recognition for reducing asymmetric communication between the blind and non-blind," *Symmetry (Basel)*, vol. 12, no. 7, 2020, doi: 10.3390/SYM12071069.
- [6] M. Z. Alom *et al.*, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," 2018, [Online]. Available: <http://arxiv.org/abs/1803.01164>.
- [7] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *SAE Int. J. Mater. Manuf.*, vol. 7, no. 3, pp. 560–566, 2014, doi: 10.4271/2014-01-0975.
- [8] D. Hossain, M. H. Imtiaz, T. Ghosh, V. Bhaskar, and E. Sazonov, "Real-Time Food Intake Monitoring Using Wearable Egocentric Camera," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, no. March 2021, pp. 4191–4195, 2020, doi: 10.1109/EMBC44109.2020.9175497.
- [9] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU) Deep Learning using Rectified Linear Units (ReLU)," no. March 2018, 2020.
- [10] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks, a Review," 2020, [Online]. Available: <http://arxiv.org/abs/2009.07485>.
- [11] M. Lin, Q. Chen, and S. Yan, "Network in network," in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014, pp. 1–10.
- [12] M. A. Amin, M. K. Hanif, M. U. Sarwar, A. Rehman, F. Waheed, and H. Rehman, "Parallel backpropagation neural network training techniques using Graphics Processing Unit," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 2, pp. 563–566, 2019, doi: 10.14569/ijacsa.2019.0100270.