

PENGENALAN ANGKA PADA CITRA TULISAN TANGAN MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)

Calista Salsabila Citra Putri Winanto, Annisa Intan Nuraini, Riza Ibnu Adam

Informatika, Universitas Singaperbangsa Karawang

Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat

2110631170058@student.unsika.ac.id

ABSTRAK

Pada era digital, penelitian tentang pengenalan angka tulisan tangan dalam *computer vision* dan pengenalan objek semakin berkembang. Digitalisasi data berbasis tulisan tangan menghadapi tantangan dalam akurasi pengenalan karena variasi pola angka yang beragam. Penelitian ini bertujuan melakukan pengenalan angka pada tulisan tangan menggunakan dataset *Handwriting Letter and Number* dan algoritma CNN. Model dibangun menggunakan arsitektur CNN dengan augmentasi citra sebagai *preprocessing* dan pelatihan selama 20 *epoch* pada ukuran citra 50 x 50 *pixels* dalam mode *grayscale*. Penelitian ini menerapkan dua skema pengujian, yakni menggunakan data uji yang telah disediakan dan mengunggah citra secara *real-time* dengan citra manual. Hasil penelitian menunjukkan bahwa model mencapai akurasi sebesar 97,09% untuk *training set* dan 97,85% untuk *validation set* pada epoch terakhir, dengan akurasi validasi secara keseluruhan adalah 97,23%. Selain itu, pengujian menunjukkan bahwa model mampu memprediksi angka dari dataset uji citra secara akurat, meskipun masih mengalami kesulitan dalam mengenali angka '4'.

Kata kunci : *Convolutional Neural Network (CNN), Object Recognition, Handwritten Digit Recognition*

1. PENDAHULUAN

Seiring meningkatnya digitalisasi, sistem pengenalan objek semakin banyak digunakan oleh masyarakat umum. Teknologi ini penting dalam berbagai aplikasi, seperti digitalisasi dokumen, sistem pembayaran otomatis, dan analisis data numerik.

Pengenalan tulisan tangan merupakan hal yang cukup banyak diteliti sebelumnya, sebab tulisan tangan manusia memiliki pola yang cukup unik. Fitur pengenalan tulisan tangan, terutama pada angka, digunakan untuk mendigitalisasi data tradisional menjadi bentuk digital secara cepat dan dapat diproses secara komputasional untuk hal-hal tertentu, seperti mencari koordinat tempat melalui internet, melakukan perhitungan dengan angka, dan sebagainya. Variasi tulisan tangan menjadi tantangan dalam klasifikasi otomatis [1].

Convolutional Neural Network (CNN) merupakan metode yang telah digunakan secara luas untuk pengenalan objek pada citra dan klasifikasi [2]. CNN merupakan pengembangan dari *Multilayer Perceptron* sehingga termasuk dalam salah satu algoritma *Deep Learning* yang dirancang untuk mengolah data dua dimensi, seperti suara dan citra [3].

CNN memiliki kemampuan untuk mengambil citra sebagai input, kemudian mementingkan berbagai aspek yang dibutuhkan, serta mampu membedakannya satu sama lain. Beberapa penelitian sebelumnya telah menggunakan CNN untuk deteksi wajah dan klasifikasi objek, seperti mobil dan sebagainya. CNN juga pernah digunakan dalam berbagai penelitian untuk mengenali tulisan tangan, khususnya angka.

Sebelumnya, telah dilakukan penelitian-penelitian untuk mengenali tulisan tangan, baik untuk aksara daerah Indonesia maupun huruf abjad

menggunakan CNN dan memberikan hasil yang cukup baik [3], [4], [5].

Dengan penelitian-penelitian sebelumnya sebagai bahan pertimbangan, penelitian ini akan melakukan pengenalan angka pada citra tulisan tangan menggunakan algoritma *Convolutional Neural Network (CNN)* dan dataset *Handwriting Letter and Number*.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Berdasarkan penelitian sebelumnya oleh Alfael Maradu Andar Turnip dan rekan-rekan yang berjudul "Pengenalan Tulisan Tangan Karakter Aksara Batak Toba dengan Metode *Convolutional Neural Network (CNN)*" [4], dilakukan implementasi metode *Convolutional Neural Network* untuk mengenali tulisan tangan karakter aksara Batak Toba. Hasil yang diperoleh model CNN dengan ukuran kernel 3 x 3, 1 hidden layer, ukuran citra masukan sebesar 150 x 150 *pixel*, dengan batas epoch sebesar 300 epoch, merupakan akurasi dari lima pengujian. Pengujian I memberikan akurasi sebesar 89,47%, akurasi pengujian II sebesar 73,68%, akurasi pengujian III sebesar 57,89%, akurasi pada pengujian IV sebesar 84,21%, dan akurasi pada pengujian V sebesar 84,21%.

Pada penelitian Nanda Putri Sazqiah dan rekan-rekan yang berjudul "Pengenalan Aksara Lampung Menggunakan Metode CNN (*Convolutional Neural Network*)" [3], CNN diimplementasikan untuk mengenali aksara Lampung dengan bahasa pemrograman Python dan Google Colaboratory. Berdasarkan hasil perhitungan menggunakan *confusion matrix*, didapatkan hasil akurasi sebesar 100%. Penelitian ini juga menjelaskan bahwa hal ini

dapat dicapai karena aksara Lampung memiliki pola yang sederhana dan pembeda utamanya berada di rotasi karakternya.

Terakhir, pada penelitian Savita Ahlawat dan rekan-rekan yang berjudul “*Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)*” [5], dilakukan implementasi CNN untuk pengenalan angka pada tulisan tangan yang telah ditingkatkan performanya. Penelitian ini mengevaluasi varian dari *convolutional neural network* untuk menghindari *preprocessing* yang kompleks, ekstraksi fitur yang mahal komputasi, serta pendekatan kombinasi yang kompleks. Penelitian ini menyarankan peran dari beragam hiperparameter dan *fine-tuning* diperlukan untuk meningkatkan performa arsitektur CNN, sehingga mendapatkan nilai akurasi 99,89% menggunakan Adam *optimizer* untuk database MNIST.

2.2. Citra

Citra merupakan suatu gambar dua dimensi yang ditampilkan pada layar komputer berbasis piksel berdasarkan nilai digital dengan pendekatan kuantisasi dan *sampling*. Kuantisasi menyatakan nilai keabuan yang besarnya berdasarkan nilai tingkat kecerahan, sedangkan *sampling* merupakan bentuk dari suatu citra piksel yang berukuran kecil atau besar [6]. Oleh karena itu, citra digital merupakan pemrosesan setiap data dua dimensi.

2.3. Pengenalan Objek

Pengenalan objek merupakan salah satu dasar dari *computer vision* yang mengimplementasikan tahap lanjutan dari deteksi objek. Deteksi objek merupakan kombinasi dari klasifikasi citra dan lokalisasi objek [7].

Tujuan dari pengenalan objek pada *computer vision* adalah pemeriksaan, registrasi, dan manipulasi. Meskipun sangat efektif pada lingkungan rekayasa tertentu, seperti lingkungan dengan pose dan pencahayaan yang terkontrol dengan baik, komputasi masih dapat kesulitan ketika terjadi rotasi objek, perubahan skala, iluminasi, dan pose 3D yang bervariasi.

2.4. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan bagian dari jaringan saraf tiruan pada kecerdasan buatan yang banyak digunakan untuk memproses citra [8].

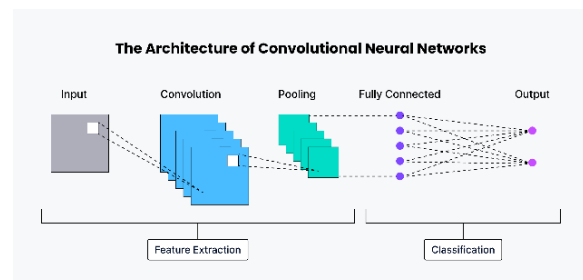
Salah satu bentuk pemrosesan citra yang menggunakan CNN adalah pengenalan objek pada citra dengan pendekatan *deep learning* [2].

Sebagai kelanjutan dari Multi-Layer Perceptron (MLP), CNN memiliki ketahanan yang cukup baik terhadap variasi citra menggunakan struktur jaringan multilayer yang runtut.

CNN menggunakan pendekatan *convolutions*, yakni mekanisme yang dirancang untuk menangkap karakteristik atau fitur tertentu dari citra input. CNN

terdiri atas lima tingkatan: i) *Input Layer*: lapisan untuk memasukkan data training, ii) *Convolutional Layer*: lapisan utama yang memproses data secara konvolusi dari layer sebelumnya, di mana fitur kunci utama diekstrak dari data, iii) *Pooling Layer*: lapisan yang mengumpulkan operasi dari setiap data citra dari lapisan sebelumnya, iv) *Fully Connected Layer*: lapisan yang terkoneksi secara utama pada setiap node dari lapisan sebelumnya, serta v) *Output Layer*: lapisan yang menghasilkan output final dari keseluruhan jaringan saraf [9].

Ilustrasi arsitektur CNN dapat dilihat pada gambar 1.



Gambar 1. Arsitektur *convolutional neural network*

2.5. Google Colaboratory

Google Colaboratory atau Google Colab merupakan tools yang disediakan oleh Google untuk membantu para peneliti mengolah data, khususnya pada bidang kecerdasan buatan. Tools ini mirip dengan Jupyter Notebook dan perbedaannya terletak pada tempatnya dijalankan, sebab Google Colaboratory dijalankan pada lingkungan Cloud.

2.6. Python

Python merupakan salah satu bahasa pemrograman tinggi yang interaktif, *object-oriented*, dan dapat dijalankan pada beragam Sistem Operasi. Penulisan bahasa Python cukup sederhana dan dilengkapi modul-modul struktur data tingkat tinggi sehingga mudah dipelajari dan diimplementasikan dalam pengembangan sistem.

2.7. TensorFlow

TensorFlow merupakan perpustakaan perangkat lunak yang dikembangkan dalam penelitian Google untuk *Machine Learning* dan *Artificial Neural Network* untuk melakukan tugas seperti regresi, klasifikasi, pengenalan pola, dan sebagainya [10].

Perpustakaan perangkat lunak ini berisi aljabar komputasi teknik pengoptimalan kompilasi sehingga mempermudah perhitungan secara matematis. Peran *TensorFlow* sebagai alat dan sumber daya sangat membantu pembangunan model deteksi tulisan tangan angka dengan lebih efisien.

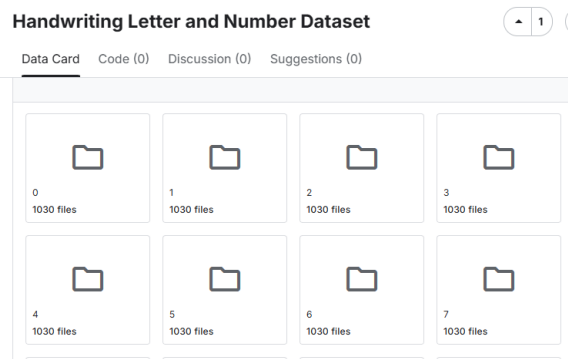
3. METODE PENELITIAN

Sistem pengenalan tulisan tangan, terutama angka, perlu melalui beberapa tahap untuk dapat memberikan hasil yang akurat sesuai dengan basis

data. Citra tulisan tangan yang didapatkan akan mengalami pengambilan fitur berupa nilai numerik untuk komputasi. Metodologi penelitian yang digunakan terdiri atas beberapa tahapan, yakni:

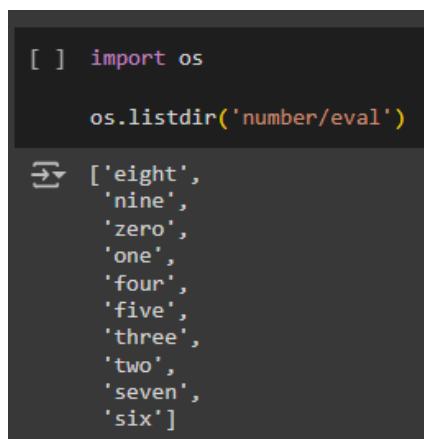
3.1. Pengumpulan data

Pada penelitian ini, dataset yang digunakan didapatkan dari *open source* kaggle.com *Handwriting Letter and Number Dataset*. Di dalamnya terdapat masing-masing 1030 data untuk setiap angka dan huruf alfabet, seperti contoh yang ditampilkan pada Gambar 2. Karena penelitian ini dilakukan untuk mendeteksi angka saja, maka dataset huruf dihilangkan, sehingga menyisakan dataset angka.



Gambar 2. *Handwriting letter and number dataset*

Dataset kemudian dimuat ke dalam Google Drive agar dapat dibaca oleh Google Colaboratory pada akun yang sama. Setelah dataset di *import* ke dalam Google Drive, folder-folder yang telah di *unzip* disatukan ke dalam folder baru. Isi folder number ditampilkan sesuai dengan Gambar 3.

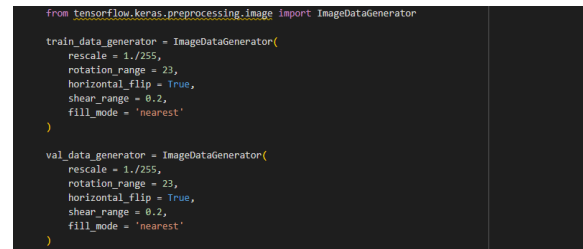


Gambar 3. Folder *number* yang telah di *unzip*

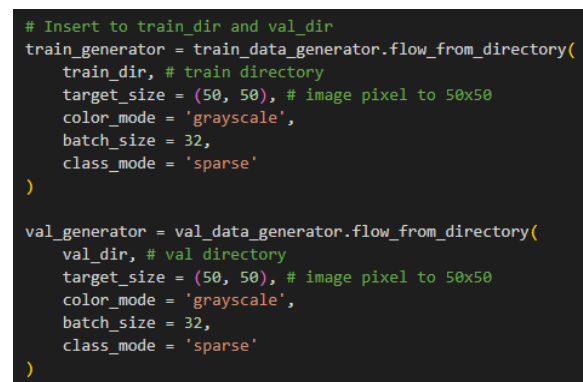
Dengan melalui bantuan pustaka *split-folders*, folder dataset dibagi menjadi dua bagian, yakni data *training* dan data *validation* dengan rasio 60% *training* dan 40% *validation*. Data *training* yang telah dipisahkan kemudian dibagi kembali menjadi *test set* dan *training set* menggunakan fungsi *train_test_split* pada modul *sklearn* dengan rasio 30% *test set* dan 70% *training set*.

3.2. Data Pre-processing

Pada penelitian ini, preprocessing citra tulisan tangan terdiri atas augmentasi citra menggunakan fungsi *ImageDataGenerator* pada modul *tensorflow* serta memberikan label pada data. Dataset citra diubah dari sisi skala/ukuran, rotasi, flip, dan lainnya sesuai dengan yang ditampilkan pada Gambar 4.



Gambar 4. *ImageDataGenerator* untuk augmentasi citra



Gambar 5. Dataset dikembalikan dalam *directory training* dan *validation*

Gambar 5 menunjukkan bahwa citra yang telah di augmentasi akan dimasukkan kembali ke dalam direktori masing-masing dengan ukuran target 50 x 50 *pixels*, mode warna *grayscale*, ukuran *batch* 32 dan mode kelas *'sparse'*. Langkah ini memuat hasil sebanyak 6180 citra pada data *training set* dengan jumlah klasifikasi sebanyak 10 kelas dan 4120 citra pada data *validation set* dengan jumlah klasifikasi yang sama.

3.3. Perancangan Sistem

Sistem dirancang menggunakan arsitektur *Convolutional Neural Network* (CNN) untuk penelitian ini. Arsitektur ini terdiri atas lapisan-lapisan yang disusun secara runtut, seperti 4 lapisan konvolusi yang menggunakan aktivasi ReLU serta *Max Pooling* pada lapisan *pooling*. Selain itu, terdapat lapisan *flatten* dan *dropout* untuk memaksimalkan komputasi menggunakan fitur-fitur yang penting. Pada urutan akhir, terdapat lapisan *dense* dengan aktivasi ReLU dan *softmax* karena sistem akan memprediksi hasil terhadap beberapa kelas.

3.4. Implementasi Sistem

Pada implementasi sistem ini, digunakan fungsi *loss sparse_categorical_crossentropy* dan *optimizer*

Adam. Fungsi *loss sparse_categorical_crossentropy* merupakan fungsi yang umum digunakan untuk klasifikasi multi-kelas dimana label targetnya adalah integer, sedangkan *optimizer Adam* merupakan algoritma optimasi umum yang biasa digunakan dalam pelatihan model *Machine Learning*.

Data yang sebelumnya telah di-generate oleh *train* dan *validation* generator dilatih sebanyak 20 *epoch* dengan *step* sebanyak ukuran *epoch* pada tiap generator. Penelitian dimulai dengan menggunakan beberapa data *testing* yang telah disiapkan sebelumnya. Data tersebut sebagian diambil dari tulisan tangan pribadi, coretan pada aplikasi *Paint*, dan dataset pada penelitian ini. Input citra juga dapat digunakan dalam implementasi sistem ini, dimana citra di-input sebagai *variable* uploaded pada *path 'fn'*. Kemudian, *variable img* akan membaca data pada *path 'fn'* lalu melakukan proses *rescale* dan mengubah mode warna menjadi *grayscale*. Citra tersebut disimpan dalam bentuk *array* dan ditampilkan pada proses output dan prediksi.

3.5. Evaluasi

Dataset terbagi atas dua bagian, yakni data *training* dan data validasi. Setelah komputasi, data hasil prediksi akan dibandingkan dengan data validasi untuk memastikan keakuratan hasil prediksi. Metrik evaluasi yang digunakan adalah *accuracy*.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Training Model

Waktu training yang dibutuhkan kurang lebih 4 menit untuk 20 *epoch* pada 32 *batch* training set. Hasil akurasi pada *epoch* terakhir menunjukkan angka 97,09% untuk *training set* dan 97,85% untuk *validation set* yang ditunjukkan pada gambar 6.

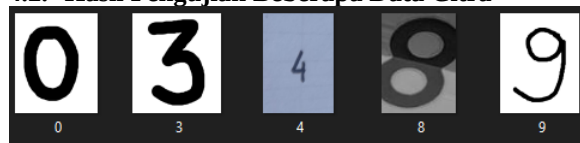
```

Epoch 17/20 ----- 14s 38ms/step - loss: 2.8660 - accuracy: 0.3174 - val_loss: 1.2519 - val_accuracy: 0.6699
Epoch 18/20 ----- 12s 30ms/step - loss: 0.9114 - accuracy: 0.6857 - val_loss: 0.6352 - val_accuracy: 0.7832
Epoch 19/20 ----- 12s 37ms/step - loss: 0.5349 - accuracy: 0.8194 - val_loss: 0.4916 - val_accuracy: 0.8564
Epoch 20/20 ----- 12s 37ms/step - loss: 0.4587 - accuracy: 0.8418 - val_loss: 0.4587 - val_accuracy: 0.8418
Epoch 17/20 ----- 18s 10ms/step - loss: 0.4384 - accuracy: 0.8882 - val_loss: 0.4872 - val_accuracy: 0.8887
Epoch 18/20 ----- 11s 352ms/step - loss: 0.3258 - accuracy: 0.8994 - val_loss: 0.2964 - val_accuracy: 0.9141
Epoch 19/20 ----- 12s 36ms/step - loss: 0.2889 - accuracy: 0.9872 - val_loss: 0.2429 - val_accuracy: 0.9297
Epoch 20/20 ----- 13s 41ms/step - loss: 0.2585 - accuracy: 0.9289 - val_loss: 0.2118 - val_accuracy: 0.9395
Epoch 17/20 ----- 12s 384ms/step - loss: 0.1585 - accuracy: 0.9578 - val_loss: 0.1758 - val_accuracy: 0.9561
Epoch 18/20 ----- 18s 314ms/step - loss: 0.1782 - accuracy: 0.9414 - val_loss: 0.1883 - val_accuracy: 0.9182
Epoch 19/20 ----- 12s 487ms/step - loss: 0.1378 - accuracy: 0.9488 - val_loss: 0.1638 - val_accuracy: 0.9588
Epoch 20/20 ----- 9s 292ms/step - loss: 0.1284 - accuracy: 0.9541 - val_loss: 0.1479 - val_accuracy: 0.9688
Epoch 17/20 ----- 12s 37ms/step - loss: 0.1147 - accuracy: 0.9658 - val_loss: 0.1794 - val_accuracy: 0.9531
Epoch 18/20 ----- 13s 40ms/step - loss: 0.1085 - accuracy: 0.9521 - val_loss: 0.1353 - val_accuracy: 0.9648
Epoch 19/20 ----- 11s 466ms/step - loss: 0.1879 - accuracy: 0.9787 - val_loss: 0.1211 - val_accuracy: 0.9668
Epoch 20/20 ----- 18s 11ms/step - loss: 0.1211 - accuracy: 0.9658 - val_loss: 0.8928 - val_accuracy: 0.9795
Epoch 17/20 ----- 11s 35ms/step - loss: 0.1340 - accuracy: 0.9580 - val_loss: 0.1284 - val_accuracy: 0.9588
Epoch 18/20 ----- 12s 36ms/step - loss: 0.1278 - accuracy: 0.9551 - val_loss: 0.1256 - val_accuracy: 0.9787
Epoch 19/20 ----- 18s 38ms/step - loss: 0.1146 - accuracy: 0.9668 - val_loss: 0.1895 - val_accuracy: 0.9746
Epoch 20/20 ----- 12s 30ms/step - loss: 0.1189 - accuracy: 0.9789 - val_loss: 0.8885 - val_accuracy: 0.9785
kloras.src.callbacks.History at 0x78538f280eb

```

Gambar 6. Hasil training model dengan 20 epoch

4.2. Hasil Pengujian Beberapa Data Citra



Gambar 7. Data citra yang diambil dari dataset

Pada pengujian ini digunakan citra asli dari data *test* sebagaimana ditunjukkan pada gambar 7. Dipilih sebanyak 5 angka dengan beberapa ketebalan dan ukuran yang berbeda, yakni angka nol, tiga, empat, delapan, dan sembilan.

Hasil Pengujian:

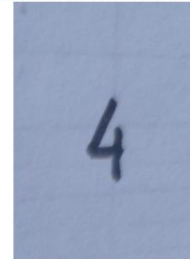
Gambar	Label Asli	Prediksi	Kesesuaian	Plot Path
0	9.png	9	Sesuai	/content/prediction_results/prediction_1_9.png...
1	0.png	0	Sesuai	/content/prediction_results/prediction_2_0.png...
2	8.jpg	8	Sesuai	/content/prediction_results/prediction_3_8.jpg...
3	4.jpg	4	Tidak Sesuai	/content/prediction_results/prediction_4_4.jpg...
4	3.png	3	Sesuai	/content/prediction_results/prediction_5_3.png...

Gambar 8. Hasil pengujian data *test*

Hasil pengujian ditampilkan pada tabel dalam gambar 8. Tabel tersebut menunjukkan hasil prediksi dari masing-masing citra, kesesuaian label, dan *path* citra hasil yang di analisis. Berdasarkan tabel tersebut, sebanyak 4 dari 5 angka dapat dikenali dengan benar. Angka yang tidak dapat dikenali secara sesuai adalah angka '4', yang diprediksi sebagai angka '1'.

Hal ini dapat disebabkan oleh ukuran dan ketebalan angka '4' yang lebih kecil dan tipis dibandingkan angka yang lain dan karakteristik tersebut tidak cukup terwakili dalam data pelatihan. Gambar 9 menunjukkan hasil prediksi pada tulisan angka 4.

Gambar: 4.jpg | Label Asli: 4 | Prediksi: 1 | Kesesuaian: Tidak Sesuai



Gambar 9. Hasil prediksi tulisan tangan angka 4

4.3. Hasil Pengujian Citra Input



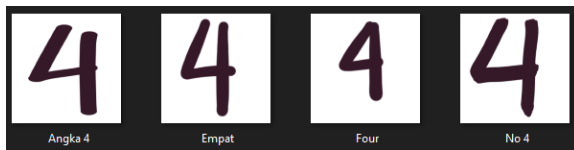
Gambar 10. Citra input manual

Pada pengujian dengan input manual dimana citra di *upload* secara *real-time* ke dalam sistem, diberikan citra berupa angka satu, dua, lima, enam, dan tujuh yang ditunjukkan pada gambar 10. Hasil pengujian ditampilkan pada tabel dalam gambar 11. Tabel tersebut berisi gambar yang diunggah, label asli, hasil prediksi dari masing-masing citra, kesesuaian label, serta *path* citra hasil yang di analisis. Hasil ini menunjukkan bahwa program berhasil memprediksi citra tulisan tangan kelima angka tersebut dengan benar.

Hasil Pengujian:					
	Gambar	Label Asli	Prediksi	Kesesuaian	Plot Path
0	1.png	1	1	Sesuai	/content/prediction_results/prediction_1_1.png...
1	2.png	2	2	Sesuai	/content/prediction_results/prediction_2_2.png...
2	5.png	5	5	Sesuai	/content/prediction_results/prediction_3_5.png...
3	6.png	6	6	Sesuai	/content/prediction_results/prediction_4_6.png...
4	7.png	7	7	Sesuai	/content/prediction_results/prediction_5_7.png...

Gambar 11. Hasil pengujian citra input

Selain itu, dilakukan pengujian dengan meng-upload beberapa citra angka '4' dengan beberapa ketebalan dan gaya, akan tetapi model masih tidak dapat memprediksi angka tersebut secara sesuai. Hal ini dapat dilihat pada gambar 12 dan 13.



Gambar 12. Citra angka '4'

Hasil Pengujian:					
	Gambar	Label Asli	Prediksi	Kesesuaian	Plot Path
0	Angka 4.jpg	4	1	Tidak Sesuai	/content/prediction_results/prediction_1_Angka...
1	Empat.jpg	4	7	Tidak Sesuai	/content/prediction_results/prediction_2_Empat...
2	Four.jpg	4	1	Tidak Sesuai	/content/prediction_results/prediction_3_Four...
3	No 4.jpg	4	7	Tidak Sesuai	/content/prediction_results/prediction_4_No 4...

Gambar 13. Pengujian manual pada angka '4'

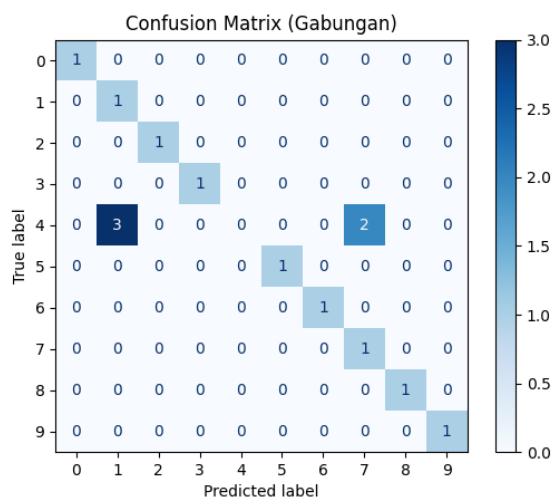
Oleh karena itu, didapatkan bahwa model memang memiliki kesulitan dalam mengenali angka '4'.

4.4. Hasil Evaluasi Akurasi Model

129/129 [=====] - 10s 75ms/step - loss: 0.8999 - accuracy: 0.9723
Validation accuracy: 97.23%

Gambar 14. Hasil validation accuracy sistem

Untuk mendapatkan hasil evaluasi pada kelompok data *validation*, digunakan fungsi `model.evaluate`. Gambar 14 menunjukkan bahwa hasil akurasi terakhir yang didapatkan adalah 97,23%. Hasil tersebut menandakan bahwa program berjalan dengan baik dan dapat mengenali tulisan tangan angka secara variatif dan akurat.



Gambar 15. Hasil confusion matrix pengujian

Hasil pengujian terhadap beberapa data ditunjukkan secara langsung melalui tabel pada gambar 8, gambar 11, dan gambar 13. Kemudian, *confusion matrix* pada gambar 15 menunjukkan pemetaan hasil keseluruhan pengujian yang telah dilakukan, sehingga didapatkan bahwa model dapat mengenali tulisan tangan angka dari '0' sampai '9' dengan baik dan cukup konsisten kecuali angka '4', meskipun telah diuji beberapa kali dengan ketebalan, ukuran, dan gaya yang berbeda.

5. KESIMPULAN DAN SARAN

Image recognition adalah salah satu implementasi dari teknik pengolahan citra digital. Berdasarkan hasil yang telah didapatkan, pengenalan tulisan tangan angka menggunakan arsitektur CNN dengan *optimizer* Adam dan fungsi loss *sparse_categorical_crossentropy* berhasil diimplementasikan. Dataset yang digunakan merupakan dataset *Handwriting Letter and Number Dataset*, khususnya data angka. Berdasarkan hasil penelitian dengan 20 epoch dan 32 batch size, didapatkan akurasi sebesar 97,09% untuk *training set* dan 97,85% untuk *validation set* pada epoch terakhir. Hasil ini menunjukkan kemampuan untuk mengenali tulisan tangan angka dan mengklasifikasinya dengan baik. Akan tetapi, pada beberapa kali pengujian dengan angka yang berbeda, mesin tidak dapat mengenali angka '4' dengan benar meskipun telah diuji dengan ukuran, ketebalan, dan gaya citra yang berbeda. Hal ini terjadi karena ketebalan tulisan pada dataset yang dipakai untuk *training* model berbeda dengan tulisan tangan pada data citra yang dipakai untuk *testing*. Penelitian selanjutnya diharapkan dapat meningkatkan kualitas sistem yang telah dikembangkan dengan menambahkan variasi model tulisan tangan, ukuran, serta ketebalan tulisan sehingga model bisa memberikan prediksi yang akurat dan adaptif.

DAFTAR PUSTAKA

- [1] A. Biswas and M. S. Islam, "An Efficient CNN Model for Automated Digital Handwritten Digit Classification," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 7, no. 1, p. 42, 2021, doi: 10.20473/jisebi.7.1.42-55.
- [2] L. Ming Seng, B. Bang Chen Chiang, Z. Arabee Abdul Salam, G. Yih Tan, and H. Tong Chai, "MNIST handwritten digit recognition with different CNN architectures," *J. Appl. Technol. Innov.*, vol. 5, no. 1, pp. 2600–7304, 2021.
- [3] N. Sazqiah *et al.*, "Pengenalan Aksara Lampung Menggunakan Metode CNN (Convolutional Neural Network)," *Semin. Nas. Ins. Prof.*, vol. 2, no. 1, 2022, doi: 10.23960/snip.v2i1.165.
- [4] A. M. A. Turnip, N. Fadillah, and M. Munawir, "Pengenalan Tulisan Tangan Karakter Aksara Batak Toba dengan Metode Convolutional Neural Network (CNN)," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 2, p. 242, 2023, doi:

- 10.26418/jp.v9i2.64242.
- [5] R. Jana, S. Bhattacharyya, and S. Das, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," *Deep Learn. Res. Appl.*, pp. 51–68, 2020, doi: 10.1515/9783110670905-003.
- [6] V. A. A. Pellokila, K. Usman, and N. K. C. Pratiwi, "Pemanfaatan Convolutional Neural Network (Cnn) Untuk Klasifikasi Jenis Beras Berbasis Citra," vol. 10, no. 5, pp. 4246–4252, 2023.
- [7] A. Salari, A. Djavadifar, X. Liu, and H. Najjaran, "Object recognition datasets and challenges: A review," *Neurocomputing*, vol. 495, pp. 129–152, 2022, doi: <https://doi.org/10.1016/j.neucom.2022.01.022>.
- [8] R. Sumathy, S. N. Swami, T. P. Kumar, V. L. Narasimha, and B. Premalatha, "Handwriting Text Recognition using CNN and RNN," *Proc. 2nd Int. Conf. Appl. Artif. Intell. Comput. ICAAIC 2023*, no. August, pp. 766–771, 2023, doi: 10.1109/ICAAIC56838.2023.10140449.
- [9] Y. Gong and P. Zhang, "Research on Mnist Handwritten Numbers Recognition based on CNN," *J. Phys. Conf. Ser.*, vol. 2138, no. 1, 2021, doi: 10.1088/1742-6596/2138/1/012002.
- [10] I. Maulana, N. Khairunisa, and R. Mufidah, "Deteksi Bentuk Wajah Menggunakan Convolutional Neural Network (Cnn)," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 6, pp. 3348–3355, 2024, doi: 10.36040/jati.v7i6.8171.