



Pengenalan Tulisan Tangan Angka menggunakan CNN dengan Arsitektur DenseNet-201 pada Dataset MNIST

Handwritten Digit Recognition using CNN with DenseNet-201 Architecture on the MNIST Dataset

Muhammad Fajril Fadillah Lubis & Susilawati

Teknik Informatika, Fakultas Teknik, Universitas Medan Area, Indonesia

Diterima: 16 April 2025; Direview: 20 April 2025; Disetujui: 24 April 2025

*Corresponding Email: donisinagaa25@gmail.com

Abstrak

Pengenalan tulisan tangan angka dengan dataset MNIST merupakan salah satu contoh penerapan dalam pengolahan citra digital. Pemilihan hyperparameter pada arsitektur CNN untuk pengenalan tulisan tangan menjadi tantangan dalam menghasilkan akurasi pengenalan yang lebih baik. Penelitian ini berfokus pada penerapan arsitektur Densenet-201 dalam mengenali tulisan tangan angka pada dataset MNIST. Tahapan penelitian ini dimulai dari preprocessing dataset, model training, model testing dan evaluasi model. Dataset MNIST memiliki 60.000 data training dan 10.000 data testing. Preprocessing dataset dilakukan dengan mengubah ukuran gambar menjadi lebih besar pada dataset MNIST. Model training menerapkan arsitektur densenet-201 dengan pemilihan hyperparameter seperti fungsi aktivasi Softmax dan Relu, optimizer yaitu Adam, RMSprop, dan SGD dan learning rate yaitu 0.1, 0.01, dan 0.001. Pada model testing menerapkan satu dari sembilan model training terbaik yang dilakukan. Evaluasi model menggunakan confusion matriks untuk mengevaluasi tingkat akurasi dan pengenalan dataset MNIST. Hasil penelitian menunjukkan bahwa arsitektur densenet-201 dengan optimizer RMSprop dan Learning rate 0.001 mendapat akurasi pengenalan tulisan tangan angka mencapai 99,49%. Penelitian ini memberikan wawasan tentang arsitektur CNN serta pemilihan hyperparameter yang optimal untuk pengolahan citra digital.

Kata Kunci: Deep Learning; Densenet-201; CNN; Pengolahan Citra Digital

Abstract

Handwritten digit recognition using the MNIST dataset is one of the applications in digital image processing. The selection of hyperparameters in the CNN architecture for handwriting recognition presents a challenge in achieving better recognition accuracy. This research focuses on the implementation of the DenseNet-201 architecture for recognizing handwritten digits in the MNIST dataset. The research stages include dataset preprocessing, model training, model testing, and model evaluation. The MNIST dataset consists of 60,000 training data and 10,000 testing data. Dataset preprocessing involves resizing the images to a larger size. The model training applies the DenseNet-201 architecture with selected hyperparameters such as activation functions (Softmax and ReLU), optimizers (Adam, RMSprop, and SGD), and learning rates (0.1, 0.01, and 0.001). The model testing uses one of the nine best-performing trained models. Model evaluation uses a confusion matrix to assess the accuracy and recognition performance on the MNIST dataset. The results show that the DenseNet-201 architecture with the RMSprop optimizer and a learning rate of 0.001 achieved a handwritten digit recognition accuracy of 99.49%. This study provides insights into CNN architectures and optimal hyperparameter selection for digital image processing

Keywords: Deep Learning; Densenet-201; CNN; Digital Image Processing



PENDAHULUAN

Pengenalan gambar adalah masalah dasar dalam *Computer Vision*, penerapan pengenalan pada gambar banyak di aplikasikan di berbagai sektor yaitu, pendidikan dalam keamanan, pengenalan sebuah gambar, dan diagnosis di dunia medis [1]. Identifikasi angka adalah salah satu tugas utama dalam bidang pengenalan citra digital, karena karakteristiknya dapat membedakan dapat diekstraksi. Pada pengenalan pola terdapat kesulitannya sendiri karena terdapat beragam gaya tulisan manusia [2].

Penelitian ini menggunakan arsitekur *Convolution Neural Network* (CNN) untuk mengenali tulisan tangan angka pada dataset MNIST. CNN dapat dikatakan mirip dengan *Artificial Neural Network* (ANN), karena terdiri dari *neuron-neuron* yang dapat mengoptimalkan dirinya sendiri dengan proses pembelajaran. CNN memiliki empat lapisan utama. Lapisan konvolusi bertujuan untuk mengekstraksi fitur dari sebuah gambar dengan cara menggeser matriks kecil (*kernel* atau *filter*) di seluruh gambar dan menghasilkan *feature maps*. Lapisan *pooling* digunakan untuk mempertahankan aspek terpenting dengan mereduksi ukuran *feature maps*, sehingga mengurangi jumlah parameter dan meningkatkan efisiensi komputasi. Lapisan terhubung penuh (*fully connected layer*) menghubungkan setiap *neuron* di lapisan tersebut dengan *neuron* dari lapisan sebelumnya dan seterusnya. Selanjutnya Lapisan *output* bertugas dalam melakukan prediksi. Karena arsitektur seperti ini, CNN membutuhkan lebih sedikit parameter untuk dipelajari dan mengurangi jumlah data yang diperlukan untuk melatih model [3]. CNN memiliki kemampuan untuk mendeteksi fitur-fitur penting dari suatu citra digital secara otomatis tanpa harus adanya intervensi dari manusia, itulah yang membuat CNN ini lebih efektif dari pendahulunya [4]. Banyak variasi arsitektur yang dapat diterapkan pada arsitektur CNN, salah satunya yaitu arsitektur Densenet-201. Arsitektur Densenet-201 merupakan bagian dari *Dense Convolutional Network* (DenseNet) yang menggunakan pola konektivitas sederhana untuk menghubungkan semua lapisan secara langsung dalam bentuk *feed-forward*, yaitu setiap lapisan menerima tambahan masukan dari semua lapisan sebelumnya dan meneruskan peta fitur (*feature maps*) ke semua lapisan berikutnya. DenseNet memiliki beberapa keunggulan termasuk mengatasi



masalah *vanishing gradient*, memperkuat propagasi fitur, mendorong penggunaan kembali fitur, dan secara signifikan mengurangi jumlah parameter [5].

Penelitian terkait penerapan arsitektur CNN dalam berbagai bidang telah banyak dilakukan dengan menghasilkan akurasi pengenalan yang sangat tinggi. Penelitian yang dilakukan oleh Yifan Wan tentang pengenalan tulisan tangan menggunakan dataset MNIST dengan menerapkan arsitektur CNN yaitu LeNet-5 memperoleh hasil akurasi pengenalan sebesar 99.32% [6]. Penelitian tentang pengenalan penyakit tanaman padi yang dilakukan oleh Primatua Sitompul dengan menerapkan arsitektur CNN yaitu DenseNet-201 memperoleh hasil akurasi pengenalan sebesar 82.99% [7]. Penelitian tentang pendeteksi pengemudi mengantuk yang dilakukan oleh Ali Hussein Hasan dengan menerapkan arsitektur CNN yaitu DenseNet-201 mendapat hasil akurasi pengenalan sebesar 99% [8]. Penelitian terdahulu tentang pengenalan tumor otak menggunakan foto MRI yang dilakukan oleh Agus Eko dengan menerapkan arsitektur DenseNet-201 memperoleh hasil akurasi pengenalan 99.65% [9].

Penelitian ini berfokus pada penerapan arsitektur Densenet-201 menggunakan hyperparameter tiga *optimizer*, yaitu Adam, RMSprop, SGD dengan tiga *learning rate* yang berbeda yaitu 0.1, 0.01, 0.001. Penelitian ini bertujuan untuk memberikan pengetahuan dalam pemilihan *hyperparameter* terhadap penerapan Arsitektur CNN dalam pengenalan tulisan tangan angka pada dataset MNIST

METODE PENELITIAN

Dataset

Dataset MNIST (*Modified National Institute of Standards and Technology*) merupakan salah satu dataset standar yang digunakan untuk menguji model pengenalan angka, yang berisi 60.000 gambar latih dan 10.000 gambar uji dengan resolusi 28×28 piksel dengan *grayscale*. Dataset ini terdiri dari 10 kelas yang berbeda, yaitu angka 0-9. Gambar latih digunakan untuk melatih model dan gambar uji akan digunakan untuk menguji model yang sudah dilatih [10]. Berikut adalah distribusi data pada dataset MNIST dan beberapa sample gambar yang ada di dataset MNIST :

Tabel 1. Distribusi Dataset MNIST

Angka	0	1	2	3	4	5	6	7	8	9	Total
Data Latih	5926	6742	5942	6155	5849	5421	5913	6265	5851	5936	60000
Data Uji	980	1135	1032	1010	982	892	958	1028	974	1009	10000





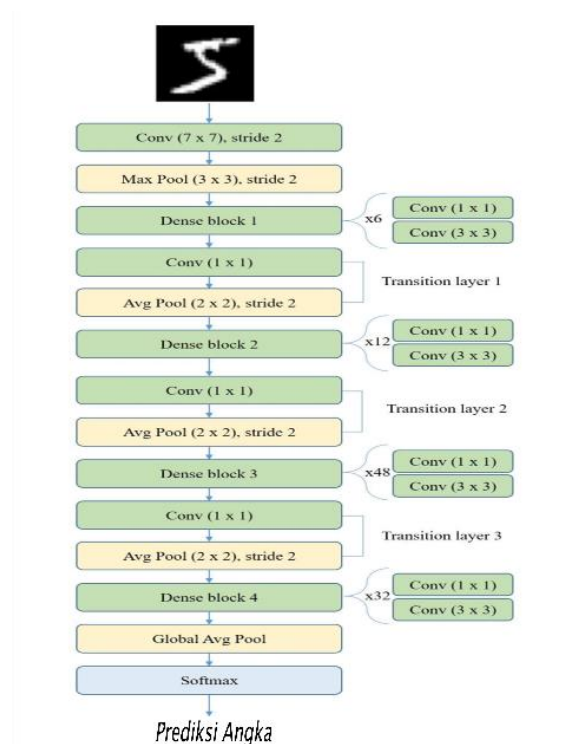
Gambar 1 Sample Gambar Pada Dataset MNIST

Preprocessing Data

Preprocessing data dimulai dari merubah resolusi pada dataset yang sebelumnya 28x28 piksel menjadi 64x64 piksel dan tetap dengan *grayscale* agar Arsitektur Densenet-201 dapat melakukan pelatihan.

Arsitektur Model

DenseNet-201 adalah arsitektur CNN yang dipopulerkan oleh Huang et al. DenseNet-201 adalah salah satu arsitektur yang dikenal mutakhir dan memiliki kinerja yang sangat baik dalam tugas-tugas *benchmark* seperti Imagenet dan CIFAR-100 [11]. Berikut arsitektur model pengenalan tulisan tangan angka menggunakan arsitektur DenseNet-201 pada dataset MNIST.



Gambar 2 Arsitektur Densenet-201 [12]

DenseNet-201 dapat meningkatkan variasi dalam masukan ke lapisan-lapisan berikutnya untuk mendapatkan efisiensi yang baik, lapisan-lapisan yang berdekatan disebut sebagai *transition layer* dan DenseNet-201 juga terdiri dari 4 *dense block* dan 3 *transition layer* agar dapat meningkatkan kemampuan dalam identifikasi [13].

Proses arsitektur Densenet-201 dimulai dari gambar yang masuk kedalam lapisan konvolusi awal berukuran 7x7, lalu ditambahkan *max pooling* agar mengurangi ukuran gambar. Setelah itu data akan melewati empat *dense block*. Setiap *dense block* terdiri dari beberapa lapisan konvolusi yang saling terhubung, ini memungkinkan fitur lapisan yang diperoleh sebelumnya dapat digunakan dilapisan berikutnya. Diantara blok-blok tersebut terdapat *Transition Layer* yang mengontrol ukuran data agar tetap efisien. Setiap blok akan memproses input berupa matriks piksel gambar. Di dalamnya terdapat fungsi aktivasi ReLU digunakan untuk menghilangkan nilai negatif dan mempertahankan nilai positif sehingga mempercepat proses pembelajaran. Lalu Filter konvolusi 3x3 mengekstraksi fitur dari gambar, dan hasilnya kembali diproses dengan ReLU. Setelah seluruh blok dilewati, data diringkas menggunakan *Global Average Pooling*, yang merangkum informasi penting dari seluruh fitur yang telah dipelajari. Hasilnya dikirim ke lapisan *dense* dengan 128 unit dan ReLU, yang berfungsi untuk mengenali pola yang lebih kompleks. Terakhir, lapisan output menggunakan fungsi aktivasi SoftMax untuk menghasilkan probabilitas dari setiap kelas, sehingga model dapat memberikan prediksi angka secara akurat [14].

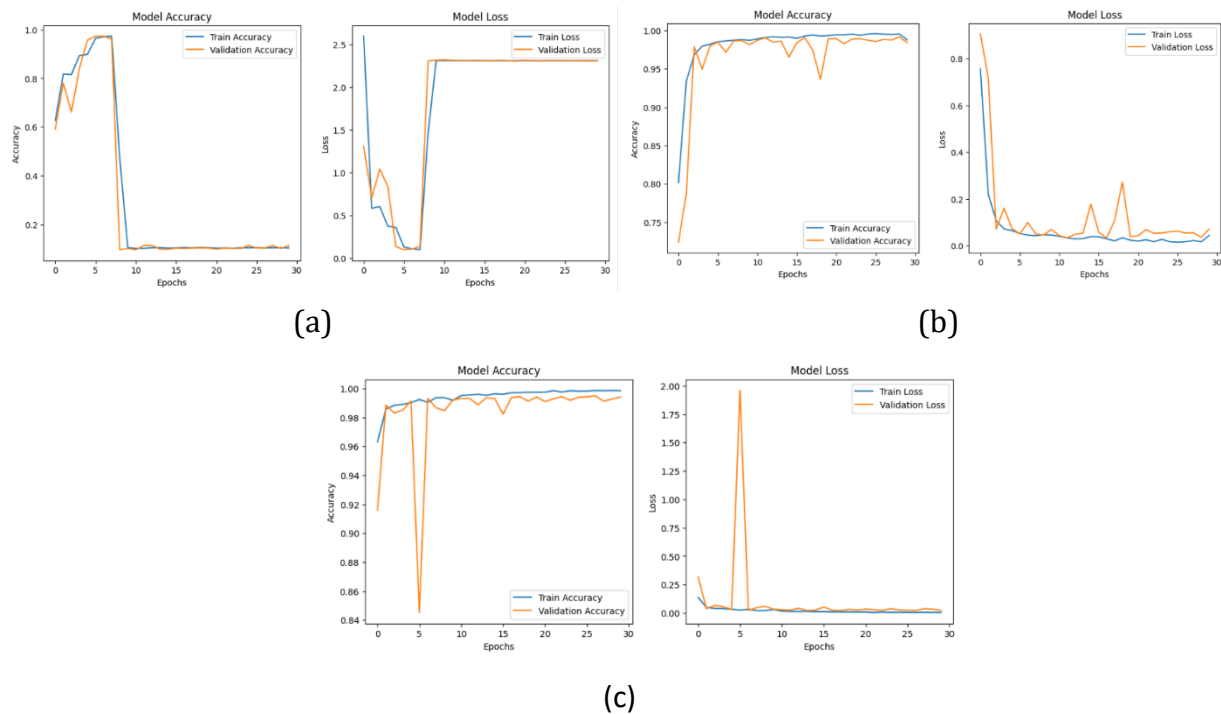
Arsitektur DenseNet-201 diterapkan untuk mengklasifikasikan tulisan tangan angka pada dataset MNIST. Model dilatih menggunakan berbagai *optimizer* yaitu Adam, RMSprop, dan SGD dan juga berbagai *learning rate* yang berbeda-beda pula yaitu 0.1, 0.01, dan 0.001 yang akan dievaluasi berdasarkan *precision*, *Recall*, *F1 score*, serta akurasi. Dengan memanfaatkan keunggulan DenseNet-201, penelitian ini bertujuan untuk memperoleh model yang lebih akurat dan efisien dibandingkan model CNN konvensional dalam mengenali tulisan tangan angka.

HASIL DAN PEMBAHASAN

Pelatihan Model

Model dilatih dengan kombinasi Hyperparameter yang disesuaikan dengan menggunakan 3 *optimizer* dan 3 *learning rate* yang berbeda dengan menggunakan *batch*

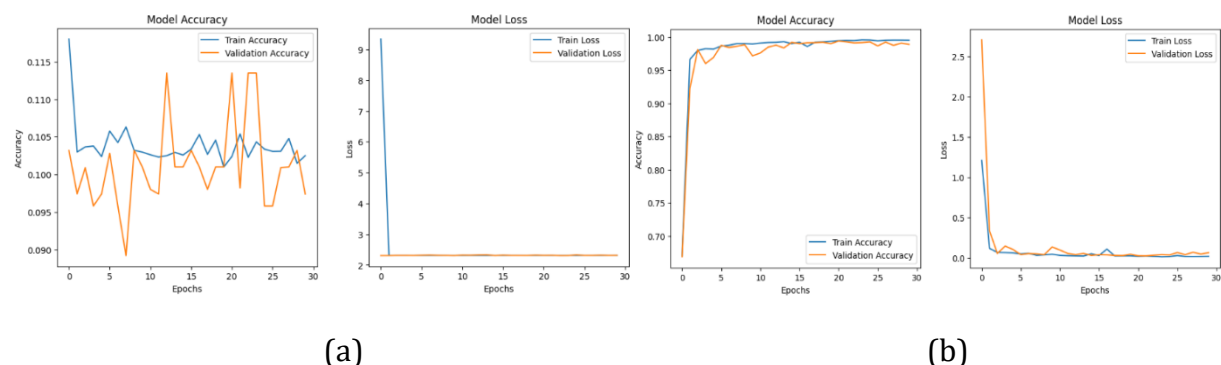
size 64 dan *epoch* sebanyak 30. Akan ada 9 model yang dilatih berdasarkan kombinasi *optimizer* dan *learning rate*.

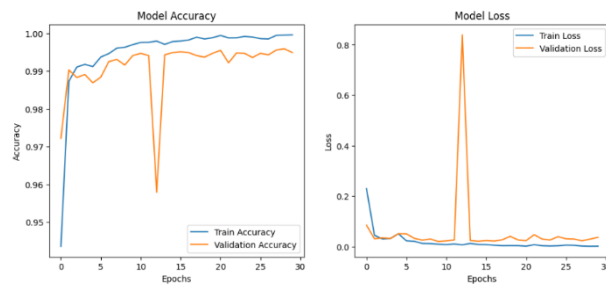


Gambar 3 Grafik Akurasi dan Model Loss Pada *Optimizer* Adam Dengan *Learning Rate* (a) 0.1, (b) 0.01, dan (c) 0.001

Gambar 3 adalah hasil dari pelatihan dengan menggunakan *optimizer* Adam dengan *learning rate* 0.1, 0.01, 0.001. Dapat dilihat pada grafik pertama kombinasi *optimizer* Adam dengan *learning rate* 0.1 mendapat akurasi yang tinggi namun sejak *epoch* ke-7 akurasi mengalami penurunan hingga seterusnya.

Selanjutnya pada grafik kedua dan ketiga kombinasi *optimizer* Adam dengan *learning rate* 0.01 dan 0.001 tidak memiliki masalah berarti dalam melakukan pelatihan.



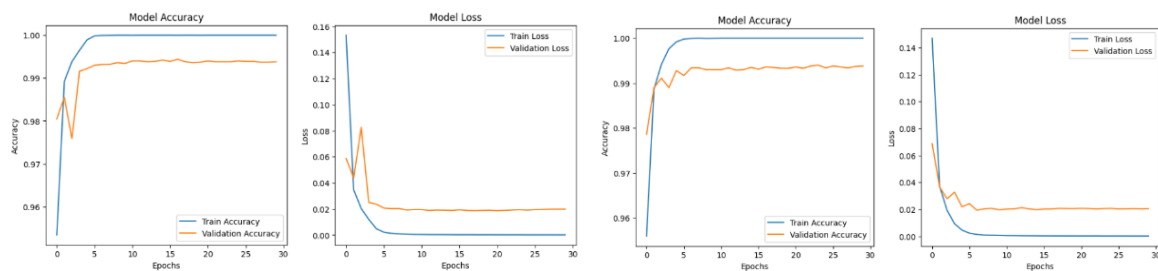


(c)

Gambar 4 Grafik Akurasi dan Model Loss Pada Optimizer RMSprop dengan Learning Rate (a) 0.1, (b) 0.01, dan (c) 0.001

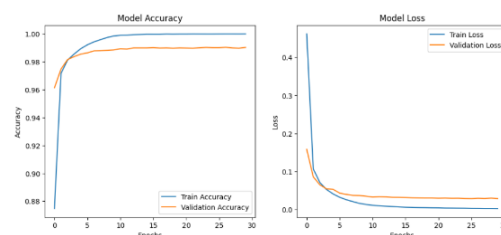
Gambar 4 adalah grafik akurasi dan *model loss* dalam pelatihan model menggunakan *optimizer* RMSprop dengan *learnig rate* 0.1, 0.01, dan 0.001. Pada grafik pertama kombinasi *optimizer* RMSprop dengan *learning rate* 0.1 juga mengalami masalah yang serupa seperti kombinasi *optimizer* Adam dengan *learning rate* 0.1 dan mendapat hasil akurasi dibawah 10%.

Pada grafik kedua dan ketiga kombinasi *opimizer* RMSprop dengan *learning rate* 0.01 dan 0.001 tidak memiliki masalah dalam melakukan pelatihan, hanya pada kombinasi *learning rate* 0.001 saja pada *epoch* ke-12 mengalami penurunan akurasi secara drastis namun untuk *epoch* selanjutnya hasil akurasi kembali naik.



(a)

(b)



(c)

Gambar 5 Grafik Akurasi dan Model Loss Pada Optimizer SGD dengan Learning Rate (a) 0.1, (b) 0.01, dan (c) 0.001

Gambar 5 adalah grafik akurasi dan *model loss* pada pelatihan model menggunakan *optimizer* SGD dengan *learning rate* 0.1, 0.01, dan 0.001. Pada 3 kombinasi ini menunjukkan hasil yang baik tanpa adanya masalah, tidak seperti pada *optimizer* Adam dan RMSprop

pada *learning rate* 0.1 pada *optimizer* SGD ini tidak mengalami penurunan akurasi seperti yang terjadi dengan kombinasi *learning rate* 0.1 yang lain.

Pengujian Model

Setelah dilakukan proses Pelatihan Model selanjutnya dilakukan pengujian menggunakan data tes yang berjumlah 10.000 data. 9 model tersebut diuji untuk menghasilkan nilai akurasi, *precision*, *recall*, dan *F1-score*. Berikut tabel hasil dari pengujian model yang telah dilakukan:

Tabel 2 Hasil Pengujian Dataset

Optimizer	Learning Rate	Akurasi	Precision	Recall	F1-Score
Adam	0.1	0.1135	0.0114	0.1000	0.0204
	0.01	0.9841	0.9839	0.9842	0.9839
	0.001	0.0041	0.9940	0.9942	0.9941
RMSprop	0.1	0.0974	0.0097	0.9942	0.9941
	0.01	0.9890	0.9889	0.9889	0.9889
	0.001	0.9949	0.9949	0.9949	0.9949
SGD	0.1	0.9938	0.9938	0.9937	0.9938
	0.01	0.9938	0.9938	0.9938	0.9938
	0.001	0.9903	0.9903	0.9902	0.9903

Dapat dilihat pada tabel 2 menunjukkan *optimizer* RMSprop dengan *learning rate* 0.001 mendapatkan akurasi tertinggi yaitu 99.49%. Terdapat 2 model yang mendapatkan akurasi yang sangat rendah yaitu kombinasi *optimizer* Adam dan RMSprop dengan *learning rate* 0.1 yang hanya mendapat akurasi dibawah 12%. Berikut adalah tabel hasil dari pengujian Dataset pada setiap label:

Tabel 3 Hasil Pengujian Dataset Per Label

Optimiz er	Learning Rate	Accurac y	Label	Jumlah Aktual	Jumlah Prediksi	Precisio n	Recall	F-1 Score
Adam	0.1	0.1135	0	980	0	0.0000	0.0000	0.0000
			1	1135	1135	0.1135	1.0000	0.2039
			2	1032	0	0.0000	0.0000	0.0000
			3	1010	0	0.0000	0.0000	0.0000
			4	982	0	0.0000	0.0000	0.0000
			5	892	0	0.0000	0.0000	0.0000
			6	958	0	0.0000	0.0000	0.0000
			7	1028	0	0.0000	0.0000	0.0000
			8	974	0	0.0000	0.0000	0.0000
			9	1009	0	0.0000	0.0000	0.0000
	0.01	0.9841	0	980	966	0.9928	0.9857	0.9892
			1	1135	1125	0.9831	0.9912	0.9871
			2	1032	1019	0.9812	0.9874	0.9843
			3	1010	986	0.9840	0.9762	0.9801



			4	982	976	0.9849	0.9939	0.9894
			5	892	889	0.9478	0.9966	0.9716
			6	958	946	0.9723	0.9875	0.9798
			7	1028	1003	0.9931	0.9757	0.9843
			8	974	944	0.9558	0.9692	0.9625
			9	1009	987	0.9870	0.9782	0.9826
0.001	0.9941		0	980	977	0.9959	0.9969	0.9964
			1	1135	1127	0.9973	0.9930	0.9951
			2	1032	1023	0.9961	0.9913	0.9937
			3	1010	1005	0.9931	0.9918	0.9924
			4	982	977	0.9919	0.9948	0.9933
			5	892	890	0.9810	0.9878	0.9844
			6	958	952	0.9969	0.9937	0.9953
			7	1028	1020	0.9942	0.9922	0.9932
			8	974	973	0.9938	0.9998	0.9968
			9	1009	997	0.0000	0.0000	0.0000
RMSpro	0.1	0.0974	0	980	0	0.0000	0.0000	0.0000
P			1	1135	0	0.0000	0.0000	0.0000
			2	1032	0	0.0000	0.0000	0.0000
			3	1010	0	0.0000	0.0000	0.0000
			4	982	0	0.0000	0.0000	0.0000
			5	892	0	0.0000	0.0000	0.0000
			6	958	0	0.0000	0.0000	0.0000
			7	1028	0	0.0000	0.0000	0.0000
			8	974	974	0.0974	1.0000	0.0000
			9	1009	0	0.0000	0.0000	0.0000
	0.01	0.9890	0	980	972	0.9939	0.9918	0.9928
			1	1135	1132	0.9974	0.9966	0.9970
			2	1032	1026	0.9837	0.9942	0.9889
			3	1010	1005	0.9795	0.9802	0.9798
			4	982	963	0.9918	0.9807	0.9862
			5	892	887	0.9758	0.9444	0.9598
			6	958	945	0.9937	0.9864	0.9900
			7	1028	1022	0.9922	0.9942	0.9932
			8	974	953	0.9784	0.9784	0.9784
			9	1009	985	0.9909	0.9762	0.9835
0.001	0.9949		0	980	980	0.9949	1.0000	0.9974
			1	1135	1133	0.9965	0.9892	0.9928
			2	1032	1026	0.9920	0.9942	0.9931
			3	1010	1005	0.9950	0.9960	0.9955
			4	982	980	0.9879	0.9980	0.9929
			5	892	888	0.9922	0.9955	0.9938

			6	958	949	0.9979	0.9906	0.9942
			7	1028	1022	0.9930	0.9942	0.9936
			8	974	971	0.9969	0.9969	0.9969
			9	1009	995	0.9950	0.9861	0.9905
SGD	0.1	0.9938	0	980	0	0.9929	0.9990	0.9959
			1	1135	1035	0.9953	0.9942	0.9947
			2	1032	0	0.9913	0.9942	0.9927
			3	1010	0	0.9951	0.9702	0.9825
			4	982	0	0.9929	0.9929	0.9929
			5	892	0	0.9944	0.9944	0.9944
			6	958	0	0.9979	0.9916	0.9947
			7	1028	0	0.9922	0.9893	0.9907
			8	974	0	0.9982	0.9898	0.9940
			9	1009	0	0.9911	0.9891	0.9901
	0.01	0.9938	0	980	979	0.9929	0.9990	0.9959
			1	1135	1133	0.9953	0.9942	0.9947
			2	1032	1026	0.9913	0.9942	0.9927
			3	1010	1007	0.9951	0.9702	0.9825
			4	982	975	0.9929	0.9929	0.9929
			5	892	887	0.9944	0.9944	0.9944
			6	958	950	0.9979	0.9916	0.9947
			7	1028	1017	0.9922	0.9893	0.9907
			8	974	966	0.9982	0.9898	0.9940
			9	1009	998	0.9911	0.9891	0.9901
	0.001	0.9903	0	980	977	0.9869	0.9969	0.9919
			1	1135	1131	0.9873	0.9965	0.9919
			2	1032	1024	0.9894	0.9922	0.9908
			3	1010	1004	0.9911	0.9041	0.9456
			4	982	975	0.9898	0.9929	0.9913
			5	892	884	0.9880	0.9910	0.9895
			6	958	938	0.9958	0.9791	0.9874
			7	1028	1013	0.9873	0.9854	0.9863
			8	974	967	0.9928	0.9928	0.9928
			9	1009	990	0.9910	0.9812	0.9861

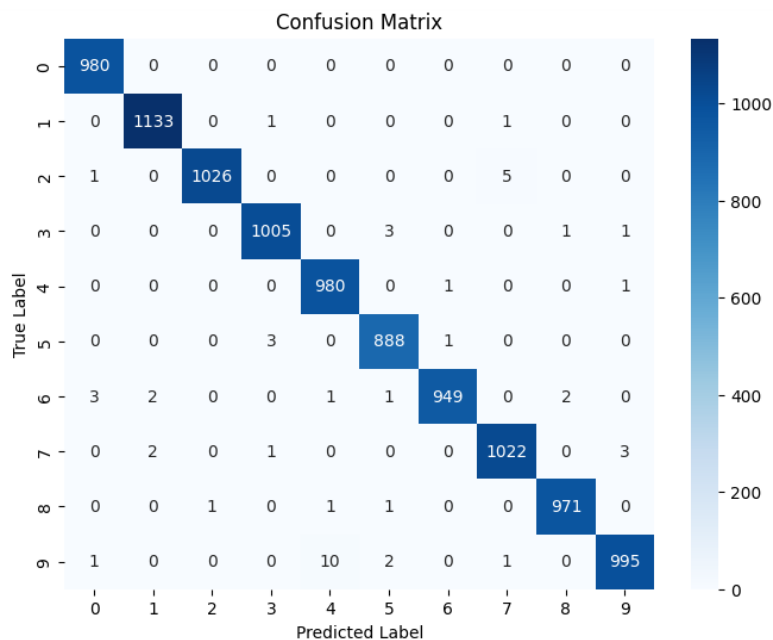
Tabel 3 adalah hasil pelatihan secara keseluruhan pada setiap label dengan 9 model yang mengkombinasikan *optimizer* Adam, RMSprop, dan SGD dengan *learning rate* 0.1, 0.01, dan 0.001

Visualisi Hasil

Visualisasi hasil pada penelitian ini menggunakan *Confusion Matrix*. Sebagaimana dalam pengujian model dilakukan didapat kombinasi terbaik yaitu *optimizer* RMSprop



dengan *learning rate* 0.001 yang memiliki tingkat akurasi 99.49%, maka berikut adalah *Confusion Matrix* pada kombinasi tersebut.



Gambar 6 Confusion Matrix pada Model Optimizer RMSprop dengan Learning Rate 0.001

Gambar 6 menunjukkan bahwa terdapat beberapa kesalahan dalam pengenalan tulisan tangan angka. Kesalahan terbanyak adalah dalam model mengenali angka 9 yang dikenali sebagai angka 4 sebanyak 10 kali dan dalam mengenali angka 2 yang dikenali sebagai angka 7 sebanyak 5 kali, terlepas dari 2 angka tersebut model dapat mengenali tulisan tangan angka pada dataset MNIST ini dengan baik dengan rata-rata hanya melakukan kesalahan pengenalan sebanya 1-3 pada setiap angka.

SIMPULAN

Arsitektur DenseNet-201 dapat melakukan pengenalan tulisan tangan angka pada dataset MNIST dengan baik. DenseNet-201 dengan kombinasi *optimizer* RMSprop dengan *learning rate* 0.001 mendapat akurasi sebesar 99.49%. Evaluasi menggunakan *confusion matrix* juga memperlihatkan hanya ada 2 angka yang memiliki kesalahan pengenalan yang terbesar, yaitu angka 9 yang dikenali sebagai angka 4 sebanyak 10 kali dan angka 2 yang dikenali sebagai angka 7 sebanyak 5 kali. Ini membuktikan bahwa DenseNet-201 dengan *optimizer* RMSprop dengan *learning rate* 0.001 adalah kombinasi terbaik dalam pengenalan tulisan tangan angka pada dataset MNIST. Penelitian menggunakan dataset yang sama yaitu MNIST dilakukan oleh Yifan Wan menghasilkan akurasi sebesar 99.32%

dengan menerapkan arsitektur LeNet-5 *optimizer self tuning* dengan learning rate 0.0001. Diharapkan penelitian ini dapat menjadi referensi bagi penelitian mendatang untuk pengenalan citra digital lain yang lebih kompleks dan dapat membuat aplikasi pendukung agar dapat dilakukan pengenalan secara *real time*.

DAFTAR PUSTAKA

- [1] B. G. e. all, "Handwritten Digits Identification Using Mnist Database Via," *IOP Conference Series: Materials Science and Engineering*, pp. 1-12, 2021.
- [2] K. e. a. S, "CNN Model for Image Classification on MNIST," *Journal of Scientific Research*, pp. 374-384, 2020.
- [3] F. S. e. all, "DenseNet-201 and Xception Pre-Trained Deep Learning Models," *Electronics*, pp. 1-23, 2023.
- [4] Y. W. e. all, "Improvement of MNIST Image Recognition Based," *IOP Conf. Series: Earth and Environmental Science*, pp. 1-8, 2020.
- [5] H. H. e. all, "Driver Drowsiness Detection Based On the DenseNet 201 Model," *Turkish Journal of Computer and Mathematics Education*, pp. 3682-3692, 2021.
- [6] C. A. T. Jaby, "Identification of Corn Leaf Diseases Comprising of Blight, Grey Spot and Rust Using DenseNet-201," *Borneo Journal of Resource Science and Technology*, pp. 125-134, 2022.
- [7] M. B. e. all, "DenseNet Based Model for Plant Diseases Diagnosis," *European Journal of Electrical Engineering and Computer Science*, pp. 1-9, 2022.
- [8] K. Adam, "A Selective Mitigation Technique of Soft Errors for DNN Models Used in Healthcare Application :DenseNet201 Case Study," *Digital Object Identifier*, pp. 65803-65832, 2021.
- [9] P. S. dkk, "Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan DenseNet 201," *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, pp. 143-150, 2022.
- [10] R. e. all, "Classification of Alzheimer disease using DenseNet-201 based on deep transfer learning technique," *PLOS ONE*, pp. 1-23, 2024.
- [11] N. R. e. all, "Enhancing Image Recognition on MNIST Dataset Through VGG16 in CNN," *Journal of Propulsion Technology*, pp. 1895-1907, 2024.
- [12] Agus Eko Minarno et all, "Classification of Brain Tumors on MRI Images Using DenseNet and Support Vector Machine," *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION*, vol. 6, no. 2, pp. 404-410, 2022.
- [13] S. A. et.all, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," *Sensors*, pp. 1-18, 2020.
- [14] M. S. e. all, "MNIST handwritten digit recognition with different CNN architectures," *Journal of Applied Technology and Innovation*, pp. 7-10, 2021.

