



Pendekatan Arsitektur Efficientnet Pada CNN Untuk Meningkatkan Pengenalan Tulisan Tangan Angka: Studi Kasus Dataset MNIST

Efficientnet's Architectural Approach to CNN to Improve Handwriting Recognition of Numbers: A Case Study of MNIST Dataset

Rahmadani Syahriful Amin Matondang & Susilawati*

Teknik Informatika, Fakultas Teknik, Universitas Medan Area, Indonesia

*Corresponding Email: susilawati@staff.uma.ac.id

Abstrak

Penelitian ini bertujuan untuk mengevaluasi kinerja arsitektur EfficientNetB1 dalam mengenali angka tulisan tangan menggunakan dataset MNIST. Dataset ini terdiri dari 60.000 data latih dan 10.000 data uji dalam format grayscale 28x28 piksel. Metode yang digunakan mencakup preprocessing data seperti resize citra, konversi grayscale ke RGB, normalisasi piksel, dan augmentasi data. Model EfficientNetB1 digunakan sebagai feature extractor, diikuti dengan lapisan dense dan output softmax. Model dilatih menggunakan tiga jenis optimizer—Adam, SGD, dan RMSprop—dengan variasi learning rate (0.001, 0.01, dan 0.1). Hasil menunjukkan bahwa kombinasi RMSprop dengan learning rate 0.001 memberikan akurasi terbaik, yaitu 97.9% pada data validasi. Kesalahan klasifikasi terutama terjadi pada angka-angka yang memiliki bentuk mirip seperti 2 dan 5. Penelitian ini memberikan kontribusi dalam pemilihan arsitektur dan hyperparameter yang optimal dalam klasifikasi citra tulisan tangan.

Kata Kunci: EfficientNetB1; MNIST; CNN; Optimizer; Pengenalan Tulisan Tangan.

Abstract

This study aims to evaluate the performance of the EfficientNetB1 architecture in recognizing handwritten digits using the MNIST dataset, which consists of 60,000 training images and 10,000 testing images in 28x28 grayscale format. The methodology includes preprocessing steps such as image resizing, grayscale to RGB conversion, pixel normalization, and data augmentation. EfficientNetB1 is used as a feature extractor, followed by dense layers and a softmax output layer for classification. The model is trained using three optimizers—Adam, SGD, and RMSprop—with varying learning rates (0.001, 0.01, and 0.1). Experimental results indicate that the combination of RMSprop and a 0.001 learning rate yields the highest validation accuracy of 97.9%. Classification errors mostly occur on digits with similar visual structures, such as 2 and 5. This research contributes valuable insights into the effective use of EfficientNetB1 and hyperparameter optimization for handwritten digit classification tasks.

Keywords: EfficientNetB1, MNIST, CNN, Optimizer, Handwritten Recognition.

PENDAHULUAN

Pengenalan tulisan tangan merupakan salah satu bidang penelitian yang berkembang pesat dalam kecerdasan buatan dan pengolahan citra. Tulisan tangan adalah representasi tulisan individu yang unik dan sering kali digunakan dalam berbagai aplikasi, seperti verifikasi tanda tangan dan pengenalan dokumen otomatis. Dataset MNIST (*Modified National Institute of Standards and Technology*) merupakan salah satu kumpulan data standar yang sering digunakan dalam penelitian pengenalan angka tulisan tangan [1], [2]. Dataset ini terdiri dari 60.000 data latih dan 10.000 data uji dalam format *grayscale* berukuran 28×28 piksel [3], [4]. Dataset ini telah menjadi standar dalam penelitian *deep learning* karena sifatnya yang sederhana namun menantang dalam klasifikasi angka [5]

Seiring berkembangnya teknologi *Deep Learning* khususnya pada penerapan algoritma *Convolutional Neural Network* (CNN), berbagai arsitektur CNN telah diterapkan dalam tugas klasifikasi citra untuk meningkatkan akurasi pengenalan tulisan tangan menggunakan dataset MNIST. Penelitian yang telah dilakukan seperti, Klasifikasi Citra Digital Tulisan Tangan Angka Menggunakan Metode CNN berhasil mendeteksi tulisan tangan dengan mencapai nilai akurasi 0,9800 dan nilai *loss* 0,0665 [6]. Penelitian yang lain, Mengoptimalkan kinerja model CNN dalam mengklasifikasikan angka tulisan tangan pada dataset MNIST. Hasil menunjukkan bahwa model dengan 4 lapisan konvolusional mencapai akurasi tertinggi sebesar 92,41%, yang menunjukkan peningkatan signifikan dalam kemampuan model untuk mengekstraksi fitur penting dari gambar dibandingkan dengan model dengan lebih sedikit lapisan [7], [8], [9].

Penelitian lain, Model CNN yang efisien untuk klasifikasi digit tulis tangan digital otomatis dengan mengusulkan lima arsitektur CNN yang berbeda untuk klasifikasi digit tulisan tangan yang mencapai akurasi 99,53% dan 98,93% dengan *optimizer* SGDm [10]. Banyaknya variasi arsitektur CNN yang dapat diimplementasikan untuk menguji kinerja CNN, membuka peluang untuk meningkatkan kinerja CNN [11]. Variasi arsitektur CNN lain yang dapat diterapkan yaitu arsitektur EfficientNet. Arsitektur EfficientNet melakukan penskalaan lebih efisien dengan menyeimbangkan kedalaman, lebar, dan resolusi yang dapat membantu dalam meningkatkan kinerja agar lebih baik [12]. Arsitektur EfficientNet memiliki beberapa variasi seperti EfficientNetB1. Arsitektur EfficientNetB1 memiliki keseimbangan antara jumlah parameter dan akurasi yang dihasilkan, sehingga diharapkan mampu memberikan performa yang baik dengan efisiensi yang tinggi [13].

Selain pemilihan arsitektur model, pemilihan *optimizer* sangat penting pada tingkat akurasi dari model [14]. Beberapa algoritma optimasi populer seperti Adam, SGD, dan RMSprop telah banyak digunakan untuk mempercepat konvergensi model dan meningkatkan stabilitas pelatihan. Penelitian yang dilakukan seperti, adam dan RMSprop yang diterapkan untuk meningkatkan konvergensi dan mencegah overfitting dan hasil menunjukkan bahwa kombinasi teknik optimasi yang diterapkan secara signifikan meningkatkan performa model CNN [15], [16]. Namun, pengaruh dari *learning rate* terhadap performa model masih menjadi topik penelitian yang menarik. Learning rate adalah salah satu parameter *training* yang ditetapkan untuk menghitung nilai koreksi bobot pada waktu proses *training* [9], [17]. Pemilihan *learning rate* yang optimal berpengaruh signifikan terhadap akurasi model *deep learning* [10].

Penelitian ini bertujuan untuk mengevaluasi kinerja model EfficientNetB1 dalam mengenali angka tulisan tangan dengan mengukur akurasi klasifikasi, *confusion matrix*, serta stabilitas model dalam proses pelatihan. Selain itu, penelitian ini akan membandingkan berbagai *optimizer* seperti Adam, SGD, dan RMSprop dengan variasi *learning rate* untuk menganalisis dampaknya terhadap akurasi klasifikasi MNIST. Hasil yang diperoleh diharapkan dapat memberikan wawasan terhadap efektivitas arsitektur EfficientNetB1 untuk mengklasifikasi tulisan tangan serta memberikan panduan dalam pemilihan *optimizer* dan pengaturan *learning rate* yang optimal.

METODE PENELITIAN

Dataset

Dataset MNIST merupakan salah satu kumpulan data pencitraan klasik terkemuka dalam bidang pembelajaran mesin, dan secara luas digunakan untuk perbandingan dalam klasifikasi gambar. Penelitian ini menggunakan dataset MNIST, yang terdiri dari 60.000 data latih dan 10.000 data uji dalam format *grayscale* dengan ukuran 28×28 piksel. Dataset ini digunakan sebagai standar dalam pengujian model klasifikasi angka tulisan tangan.

Tabel 1. Sebaran 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

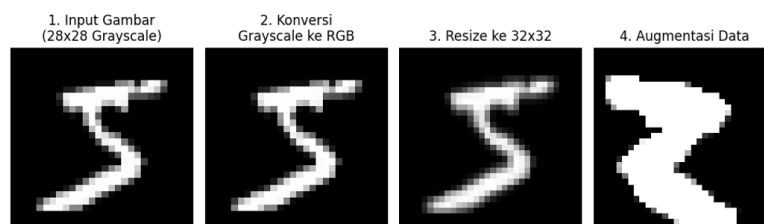
Berikut sampel gambar dataset MNIST.



Gambar 1 Sample gambar dataset MNIST

Preprocessing Data

Agar dataset MNIST dapat digunakan sebagai input pada model EfficientNetB1, dilakukan beberapa tahap *preprocessing* seperti, *resize* citra, konversi grayscale ke RGB, normalisasi piksel, dan augmentasi data. Dalam *resize* citra, gambar asli berukuran 28x28 piksel diperbesar menjadi 32x32 piksel menggunakan interpolasi bilinear agar sesuai dengan kebutuhan input EfficientNetB1. Dalam proses konversi *grayscale* ke RGB, EfficientNetB1 dirancang untuk menerima input citra dengan 3 kanal warna (RGB), sehingga gambar grayscale dikonversi menjadi 3-channel dengan menyalin nilai piksel grayscale ke setiap kanal RGB. Dalam proses normalisasi piksel, semua nilai piksel dinormalisasi ke rentang 0-1 dengan cara membagi nilai piksel dengan 255. Pada bagian augmentasi data dilakukan proses untuk meningkatkan kemampuan generalisasi model, augmentasi data menggunakan *ImageDataGenerator* yang mencakup rotasi hingga 10 derajat, pergeseran horizontal dan vertikal sebesar 10%, dan flip horizontal untuk meningkatkan variasi data. Pada augmentasi data menggunakan *ImageDataGenerator* yang digunakan tidak menambah data secara permanen, namun hanya mengubah gambar saat *training*.

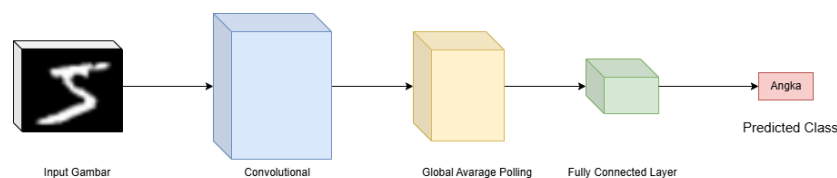


Gambar 2 Tahap *preprocessing* data

Arsitektur Model

Model yang digunakan dalam penelitian ini adalah EfficientNetB1, yang merupakan varian dari EfficientNet dengan efisiensi tinggi dalam klasifikasi citra. Model ini digunakan sebagai *feature extractor*, dengan struktur sebagai berikut:

1. EfficientNetB1 sebagai *backbone* untuk mengekstrak fitur dari gambar.
2. Lapisan *Global Average Pooling* untuk mereduksi dimensi fitur, mengurangi jumlah parameter model.
3. Lapisan *Dense* dengan 128 unit dan aktivasi ReLU digunakan untuk meningkatkan generalisasi model.
4. Lapisan *Output Dense* dengan 10 *neuron* dan aktivasi Softmax digunakan untuk melakukan klasifikasi angka 0-9.



Gambar 3 Arsitektur Model

HASIL DAN PEMBAHASAN

Training Model dengan Variasi Optimizer dan Learning Rate

Model dilatih menggunakan tiga jenis *optimizer* (Adam, SGD, RMSprop) dengan tiga *learning rate* yang berbeda (0.001, 0.01, 0.1). Berikut adalah tabel yang menunjukkan akurasi pelatihan dan validasi untuk setiap kombinasi *optimizer* dan *learning rate*:

Tabel 1. Model pelatihan dengan variasi *optimizer* dan *learning rate*

Optimizer	Learning Rate	Train Accuracy	Validation Accuracy	Train Loss	Validation Loss
Adam	0,001	91.3%	94.1%	0.2822	0.1826
Adam	0,01	11%	10.9%	2.3034	2.3019
Adam	0,1	10.4%	10.2%	2.3220	2.3082
SGD	0,001	52.6%	63.2%	1.3358	1.0863
SGD	0,01	88.7%	93.3%	0.3402	0.2087
SGD	0,1	93.9%	93.9%	0.2003	0.2373
RMSprop	0,001	95.8%	97.9%	0.1656	0.0746
RMSprop	0,01	11%	11.1%	2.3039	2.3018
RMSprop	0,1	11%	11.1%	2.3041	2.3018

Hasil pelatihan model EfficientNetB1 menunjukkan bahwa kombinasi *optimizer* RMSprop dengan *learning rate* 0.001 memberikan hasil terbaik, dengan *train accuracy* sebesar 95.8% dan *validation accuracy* mencapai 97.9%. Berdasarkan Tabel 1, dapat dilihat bahwa kombinasi *optimizer* dan *learning rate* yang tepat sangat berpengaruh

terhadap keberhasilan model dalam melakukan klasifikasi angka MNIST. Sebaliknya, nilai *learning rate* yang terlalu besar seperti 0.1 menyebabkan model gagal belajar, yang terlihat dari akurasi yang sangat rendah yaitu sekitar 10% pada semua *optimizer* yang digunakan.

Optimizer Adam dan SGD juga menunjukkan performa yang cukup baik pada *learning rate* 0.001 dan 0.01, tetapi performanya masih lebih rendah dibandingkan dengan RMSprop. *Optimizer* SGD dengan *learning rate* 0.1 menghasilkan akurasi yang cukup tinggi, tetapi nilai *validation loss*-nya lebih tinggi dibandingkan *training loss*, yang mengindikasikan adanya *overfitting*.

Hasil Pengujian Model

Berikut adalah tabel yang menunjukkan hasil pengujian model pengenalan tulisan tangan angka MNIST yang berhasil diprediksi dengan benar.

Tabel 2. hasil pengenalan pengenalan tulisan tangan angka MNIST

Angka	Optimizer	Learning Rate	Jumlah Aktual	Jumlah Angka Berhasil Prediksi	Jumlah Angka yang Diprediksi dengan Benar	Precision	Recall	F1-Score	Akurasi Prediksi
0	Adam	0,001	980	990	974	0.9838	0.9939	0.9888	99.39
	Adam	0,01	980	0	0	0	0	0	0
	Adam	0,1	980	0	0	0	0	0	0
	SGD	0,001	980	994	934	0.9396	0.9531	0.9463	95.31
	SGD	0,01	980	987	977	0.9899	0.9969	0.9934	99.69
	SGD	0,1	980	988	977	0.9889	0.9969	0.9929	99.69
	RMSprop	0,001	980	988	977	0.9889	0.9969	0.9929	99.69
	RMSprop	0,01	980	0	0	0	0	0	0.00
	RMSprop	0,1	980	0	0	0	0	0	0.00
1	Adam	0,001	1135	1171	1133	0.9675	0.9982	0.9827	99.82
	Adam	0,01	1135	0	0	0	0	0	0.00
	Adam	0,1	1135	0	0	0	0	0	0.00
	SGD	0,001	1135	1137	1117	0.9824	0.9841	0.9833	98.41
	SGD	0,01	1135	1130	1127	0.9973	0.9930	0.9951	99.30
	SGD	0,1	1135	1133	1130	0.9974	0.9956	0.9965	99.56
	RMSprop	0,001	1135	1127	1125	0.9982	0.9912	0.9947	99.12
	RMSprop	0,01	1135	10000	1135	0.1135	1.0000	0.2039	100.00
	RMSprop	0,1	1135	10000	1135	0.1135	1.0000	0.2039	100.00

	RMSprop	0,1	1135	10000	1135	0.1135	1.0000	0.2039	100.00
2	Adam	0,001	1132	1042	1007	0.9664	0.8896	0.9264	88.96
	Adam	0,01	1132	0	0	0	0	0	0.00
	Adam	0,1	1132	0	0	0	0	0	0.00
	SGD	0,001	1132	1023	924	0.9032	0.8163	0.8575	81.63
	SGD	0,01	1132	1041	994	0.9549	0.8781	0.9149	87.81
	SGD	0,1	1132	1048	1009	0.9628	0.8913	0.9257	89.13
	RMSprop	0,001	1132	1005	985	0.9801	0.8701	0.9219	87.01
	RMSprop	0,01	1132	0	0	0	0	0	0
	RMSprop	0,1	1132	0	0	0	0	0	0
3	Adam	0,001	1010	1009	998	0.9891	0.9881	0.9886	98.81
	Adam	0,01	1010	0	0	0	0	0	0
	Adam	0,1	1010	0	0	0	0	0	0
	SGD	0,001	1010	952	774	0.8130	0.7663	0.7890	76.63
	SGD	0,01	1010	1014	994	0.9803	0.9842	0.9822	98.42
	SGD	0,1	1010	998	990	0.9920	0.9802	0.9861	98.02
	RMSprop	0,001	1010	1015	1008	0.9931	0.9980	0.9956	99.80
	RMSprop	0,01	1010	0	0	0	0	0	0
	RMSprop	0,1	1010	0	0	0	0	0	0
4	Adam	0,001	982	981	969	0.9878	0.9868	0.9873	98.68
	Adam	0,01	982	0	0	0	0	0	0
	Adam	0,1	982	0	0	0	0	0	0
	SGD	0,001	982	1012	957	0.9457	0.9745	0.9599	97.45
	SGD	0,01	982	983	969	0.9858	0.9868	0.9863	98.68
	SGD	0,1	982	994	975	0.9809	0.9929	0.9868	99.29
	RMSprop	0,001	982	932	932	1.0000	0.9491	0.9739	94.91
	RMSprop	0,01	982	0	0	0	0	0	0
	RMSprop	0,1	982	0	0	0	0	0	0
5	Adam	0,001	892	904	865	0.9569	0.9697	0.9633	96.97
	Adam	0,01	892	0	0	0	0	0	0
	Adam	0,1	892	0	0	0	0	0	0
	SGD	0,001	892	823	666	0.8092	0.7466	0.7767	74.66
	SGD	0,01	892	883	852	0.9649	0.9552	0.9600	95.52
	SGD	0,1	892	911	872	0.9572	0.9776	0.9673	97.76
	RMSprop	0,001	892	925	976	1.0551	1.0942	1.0743	109.42

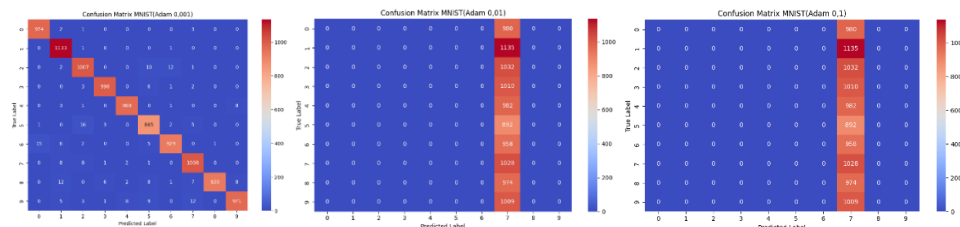
	RMSprop	0,01	892	0	0	0	0	0	0
	RMSprop	0,1	892	0	0	0	0	0	0
6	Adam	0,001	958	947	929	0.9810	0.9697	0.9753	96.97
	Adam	0,01	958	0	0	0	0	0	0
	Adam	0,1	958	0	0	0	0	0	0
	SGD	0,001	958	969	876	0.9040	0.9144	0.9092	91.44
	SGD	0,01	958	946	931	0.9841	0.9718	0.9779	97.18
	SGD	0,1	958	950	942	0.9916	0.9833	0.9874	98.33
	RMSprop	0,001	958	948	933	0.9842	0.9739	0.9790	97.39
	RMSprop	0,01	958	0	0	0	0	0	0
	RMSprop	0,1	958	0	0	0	0	0	0
7	Adam	0,001	1028	1038	1008	0.9711	0.9805	0.9758	98.05
	Adam	0,01	1028	10000	1028	0.1028	1.0000	0.1864	100.00
	Adam	0,1	1028	10000	1028	0.1028	1.0000	0.1864	100.00
	SGD	0,001	1028	1033	972	0.9409	0.9455	0.9432	94.55
	SGD	0,01	1028	1025	1009	0.9844	0.9815	0.9830	98.15
	SGD	0,1	1028	1019	1009	0.9902	0.9815	0.9858	98.15
	RMSprop	0,001	1028	1020	1020	1.0000	0.9922	0.9961	99.22
	RMSprop	0,01	1028	0	0	0	0	0	0
	RMSprop	0,1	1028	0	0	0	0	0	0
8	Adam	0,001	974	931	930	0.9989	0.9548	0.9764	95.48
	Adam	0,01	974	0	0	0	0	0	0
	Adam	0,1	974	0	0	0	0	0	0
	SGD	0,001	974	1041	907	0.8713	0.9312	0.9002	93.12
	SGD	0,01	974	983	966	0.9827	0.9918	0.9872	99.18
	SGD	0,1	974	974	964	0.9897	0.9897	0.9897	98.97
	RMSprop	0,001	974	980	970	0.9898	0.9959	0.9928	99.59
	RMSprop	0,01	974	0	0	0	0	0	0
	RMSprop	0,1	974	0	0	0	0	0	0
9	Adam	0,001	1009	987	971	0.9838	0.9623	0.9729	96.23
	Adam	0,01	1009	0	0	0	0	0	0
	Adam	0,1	1009	0	0	0	0	0	0
	SGD	0,001	1009	1016	921	0.9065	0.9128	0.9096	91.28
	SGD	0,01	1009	1008	988	0.9802	0.9792	0.9797	97.92
	SGD	0,1	1009	985	978	0.9929	0.9693	0.9809	96.93

RMSpro	0,001	1009	1060	1005	0.9481	0.9960	0.9715	99.60
p								
RMSpro	0,01	1009	0	0	0	0	0	0
p								
RMSpro	0,1	1009	0	0	0	0	0	0
p								

Berdasarkan hasil di atas, model menunjukkan performa yang sangat baik dengan rata-rata akurasi prediksi lebih dari 96.54% untuk setiap angka dalam dataset MNIST.

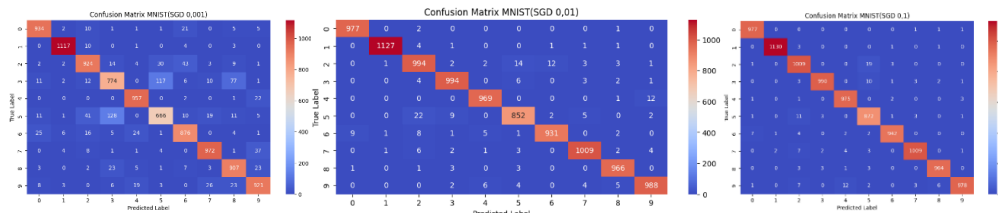
Evaluasi pengujian

Berikut adalah evaluasi menggunakan *confusion matrix* berdasarkan hasil model dengan tiga jenis *optimizer* yaitu Adam, SGD, dan RMSprop dengan tiga *learning rate* yang berbeda yaitu 0.001, 0.01, dan 0.1.



Gambar 4. Evaluasi kesalahan prediksi *optimizer* adam dengan *learning rate* 0.001, 0.01, dan 0.1

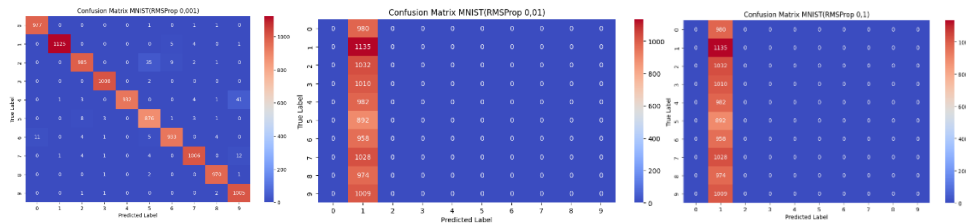
Pada gambar 4, *confusion matrix* ini menunjukkan bagaimana model dengan *optimizer* Adam memprediksi angka MNIST. Pada *learning rate* 0.001, model dapat mengklasifikasikan angka dengan cukup baik, meskipun masih terdapat kesalahan pada angka yang memiliki bentuk serupa seperti 2 dan 5. Namun, pada *learning rate* 0.01 dan 0.1, model gagal belajar dengan baik, terlihat dari banyaknya angka yang salah diklasifikasikan dan distribusi prediksi yang tidak seimbang.



Gambar 5 Evaluasi kesalahan prediksi *Optimizer* SGD dengan *learning rate* 0.001, 0.01, dan 0.1

Pada gambar 5, *confusion matrix* ini menunjukkan bagaimana model dengan *optimizer* SGD memprediksi angka MNIST. Pada *learning rate* 0.001, model memiliki

tingkat kesalahan lebih tinggi dibandingkan dengan RMSprop dan Adam, tetapi performanya meningkat pada *learning rate* 0.01 dan 0.1.

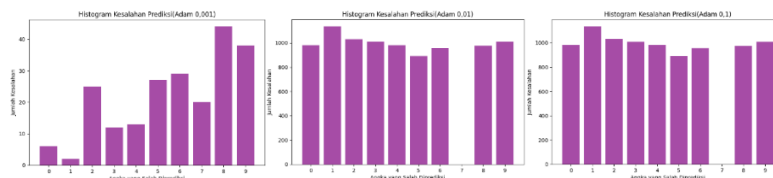


Gambar 6. Evaluasi kesalahan prediksi *optimizer* RMSprop dengan *learning rate* 0.001, 0.01, dan 0.1

Pada gambar 6, Confusion matrix ini menunjukkan hasil prediksi model menggunakan *optimizer* RMSprop. Pada *learning rate* 0.001, model memiliki performa terbaik dengan kesalahan prediksi yang minimal. Pada *learning rate* 0.01 dan 0.1 mengalami kegagalan dalam proses pembelajaran, terlihat dari distribusi prediksi yang tidak sesuai.

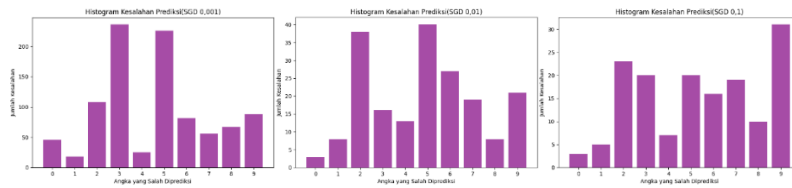
Hasil prediksi evaluasi model menggunakan *confusion matrix* menunjukkan bahwa beberapa angka memiliki tingkat kesalahan prediksi lebih tinggi. Kesalahan prediksi paling sering terjadi pada angka-angka dengan bentuk cenderung serupa, seperti angka 2 dan 5 serta angka 3 dan 8. Hal ini dikarenakan beberapa digit memiliki pola piksel yang mirip, sehingga model mengalami kesulitan dalam membedakan antara angka tersebut.

Untuk menunjukkan distribusi angka-angka yang salah diprediksi dapat digambar melalui gambar grafik histogram berikut.



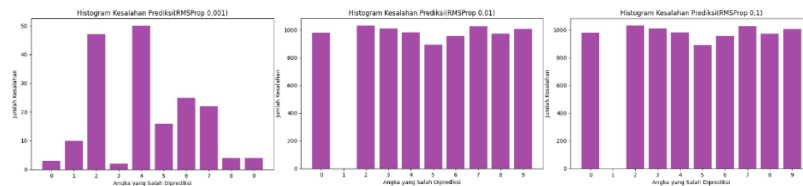
Gambar 7 Histogram kesalahan prediksi *optimizer* Adam dengan *learning rate* 0.001, 0.01, dan 0.1

Histogram ini menunjukkan distribusi jumlah kesalahan prediksi pada masing-masing angka. Pada gambar diatas, *learning rate* 0.001 mengalami kesalahan prediksi yang tersebar merata, sedangkan pada *learning rate* 0.01 dan 0.1, kesalahan meningkat drastis, menunjukkan bahwa model tidak dapat mengenali angka dengan baik.



Gambar 8. Histogram kesalahan prediksi *optimizer* SGD dengan *learning rate* 0.001, 0.01, dan 0.1

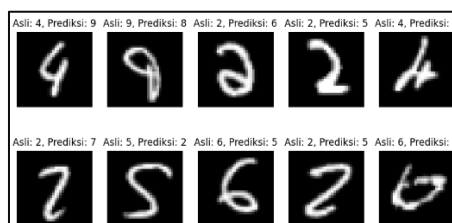
Histogram ini menunjukkan bahwa *optimizer* SGD dengan *learning rate* 0.001 memiliki lebih banyak kesalahan dibandingkan dengan *learning rate* 0.01 dan 0.1. *Learning rate* 0.1 menghasilkan kesalahan prediksi yang lebih rendah tetapi berpotensi mengalami overfitting.



Gambar 9 Histogram kesalahan Prediksi *optimizer* RMSprop dengan *learning rate* 0.001, 0.01, dan 0.1

Histogram ini menunjukkan bahwa model dengan *optimizer* RMSprop dan *learning rate* 0.001 memiliki jumlah kesalahan prediksi yang paling sedikit. Sementara itu, pada *learning rate* 0.01 dan 0.1, kesalahan prediksi sangat tinggi karena model gagal belajar dengan baik.

Berikut adalah sample gambar angka yang mengalami kesalahan prediksi.



Gambar 10 gambar angka yang mengalami kesalahan prediksi.

Beberapa sampel angka prediksi yang salah dapat diamati dengan melihat angka yang diklasifikasikan secara keliru oleh model. Sebagai contoh, angka 2 terkadang diklasifikasikan sebagai angka 5

SIMPULAN

Penelitian ini telah menguji performa model EfficientNetB1 dalam mengklasifikasikan angka tulisan tangan pada dataset MNIST dengan menerapkan tiga jenis *optimizer* yaitu Adam, SGD, dan RMSprop serta tiga variasi *learning rate* yaitu 0.001,



0.01, 0.1. Dari hasil eksperimen, ditemukan bahwa kombinasi *optimizer* RMSprop dengan *learning rate* 0.001 memberikan akurasi tertinggi, yaitu *train accuracy* sebesar 95.8% dengan *validation accuracy* mencapai 97.9%. Evaluasi menggunakan *confusion matrix* menunjukkan bahwa kesalahan prediksi terutama terjadi pada angka yang memiliki bentuk serupa, seperti angka 2 yang sering diklasifikasikan sebagai angka 5. Meskipun demikian, rata-rata akurasi prediksi model mencapai lebih dari 96.54%, yang membuktikan efektivitasnya dalam mengklasifikasi angka tulisan tangan. Histogram kesalahan prediksi angka mengungkap bahwa sebagian besar kesalahan terjadi pada angka yang memiliki bentuk mirip satu sama lain. Beberapa contoh prediksi yang keliru juga telah dianalisis untuk memahami pola kesalahan yang terjadi. Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa model EfficientNetB1 yang dikombinasikan dengan *optimizer* RMSprop dan *learning rate* 0.001 adalah kombinasi terbaik untuk meningkatkan akurasi pengenalan tulisan tangan angka MNIST. Studi ini memberikan wawasan penting terkait pemilihan arsitektur *deep learning* serta pengaturan *hyperparameter* yang optimal dalam klasifikasi citra. Selain itu, penelitian ini juga dapat dijadikan referensi untuk pengembangan model pengenalan tulisan tangan lainnya dengan dataset yang lebih kompleks dan bervariasi di masa mendatang.

DAFTAR PUSTAKA

- [1] O. Nocentini, J. Kim, M. Bashir, F. C.- Sensors, and undefined 2022, "Image classification using multiple convolutional neural networks on the fashion-MNIST dataset," *mdpi.com*, Accessed: Apr. 16, 2025. [Online]. Available: <https://www.mdpi.com/1424-8220/22/23/9544>
- [2] H. Xiao, K. Rasul, R. V. preprint arXiv:1708.07747, and undefined 2017, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arxiv.org*, 2017, Accessed: Apr. 16, 2025. [Online]. Available: <https://arxiv.org/abs/1708.07747>
- [3] G. Cohen, S. Afshar, ... J. T.-2017 international joint, and undefined 2017, "EMNIST: Extending MNIST to handwritten letters," *ieeexplore.ieee.org*, Accessed: Apr. 16, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7966217/>
- [4] F. Chen, N. Chen, H. Mao, H. H. preprint arXiv:1811.08278, and undefined 2018, "Assessing four neural networks on handwritten digit recognition dataset (MNIST)," *arxiv.org*, Accessed: Apr. 16, 2025. [Online]. Available: <https://arxiv.org/abs/1811.08278>
- [5] Feiyang Chen *et al.*, "Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST)," *Journal of Computer Science Research*, vol. 6, no. 3, pp. 17–22, 2024, doi: 10.30564/jcsr.v6i3.6804.
- [6] A. A. Kurniawan, R. D. Syah, and R. Ariyani, "Klasifikasi Citra Digital Tulisan Tangan Angka Menggunakan Metode Convolutional Neural Network," *JUIT (Jurnal Ilmiah Teknik*, vol. 1, no. 1, pp. 36–41,.
- [7] D. Septiani, M. Fadhli, and S. Soim, "CNN Algorithm Optimization for Classifying Numbers in Handwriting," *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 27, no. 1, pp. 57–63,.
- [8] K. Cheng, R. Tahir, L. K. Eric, and M. Li, "An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset," *Multimed Tools Appl*, vol. 79, no. 19–20, pp. 13725–13752, May 2020, doi: 10.1007/S11042-019-08600-2.

- [9] Feiyang Chen *et al.*, "Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST)," *Journal of Computer Science Research*, vol. 6, no. 3, pp. 17–22, Jul. 2024, doi: 10.30564/jcsr.v6i3.6804.
- [10] A. Biswas and Md. S. Islam, "An Efficient CNN Model for Automated Digital Handwritten Digit Classification," *Journal of Information Systems Engineering and Business Intelligence*, vol. 7, no. 1, p. 42, 2021, doi: 10.20473/jisebi.7.1.42-55.
- [11] S. Kadam, A. Adamuthe, A. P.-J. of scientific research, and undefined 2020, "CNN model for image classification on MNIST and fashion-MNIST dataset," *bhu.ac.in*, doi: 10.37398/JSR.2020.640251.
- [12] W. R. PERDANI, R. MAGDALENA, and N. K. CAECAR PRATIWI, *Deep Learning untuk Klasifikasi Glaukoma dengan menggunakan Arsitektur EfficientNet*, vol. 10, no. 2. ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, 2022. doi: 10.26760/elkomika.v10i2.322.
- [13] S. Supriyanto, S. Sunardi, and I. Riadi, "Pengaruh Nilai Hidden layer dan Learning rate Terhadap Kecepatan Pelatihan Jaringan Saraf Tiruan Backpropagation," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 1, p. 27, 2022, doi: 10.26798/jiko.v6i1.508.
- [14] S. A. Syifa and I. A. Dewi, "Arsitektur Resnet-152 dengan Perbandingan Optimizer Adam dan RMSProp untuk Mendeteksi Penyakit Paru – Paru," *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, vol. 7, no. 2, pp. 139–150, 2022, [Online]. Available: <https://doi.org/10.26760/mindjournal.v7i2.139-150>
- [15] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, "RESOLUSI : Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network," *Media Online*, vol. 4, no. 5, p. 509, 2024, [Online]. Available: <https://djournals.com/resolusi>
- [16] A. Baldominos, Y. Saez, P. I.-A. Sciences, and undefined 2019, "A survey of handwritten character recognition with mnist and emnist," *mdpi.com*, Accessed: Apr. 16, 2025. [Online]. Available: <https://www.mdpi.com/2076-3417/9/15/3169>
- [17] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," Aug. 2017, Accessed: Apr. 16, 2025. [Online]. Available: <http://arxiv.org/abs/1708.07747>